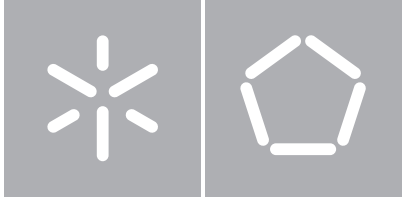
**Universidade do Minho**
Escola de Engenharia



**Universidade do Minho**
Escola de Engenharia

Dissertação de Mestrado

Mestrado em Redes e Serviços de Comunicações

# Uma Abordagem Estratificada à Monitorização de Serviços Cloud

Nuno Miguel Felgueiras Palhares

Dissertação submetida à Universidade do Minho para obtenção do grau de Mestre em Redes e Serviços de Comunicações, elaborada sob a supervisão científica da Prof. Dr$^{a}$ Maria Solange Pires Ferreira Rito Lima.

Universidade do Minho

Escola de Engenharia

Departamento de Informática

Outubro, 2012

# Agradecimentos

Esta dissertação assinala a conclusão de mais um ciclo da minha vida académica. Contudo, não seria possível sem a ajuda e apoio de algumas pessoas. Como autor desta dissertação gostaria de agradecer a todos os que tornaram possível a realização da mesma.

Em primeiro lugar gostaria de agradecer à minha orientadora, Professora Doutora Solange Rito Lima, pelo apoio e orientação científica, assim como pela disponibilidade que sempre apresentou para com o meu trabalho.

Um agradecimento ao Ricardo Vilaça do Departamento de Informática da Universidade do Minho, pela disponibilidade e ajuda prestada na elaboração de alguns testes práticos no mesmo departamento.

Agradeço também aos meus colegas David Cunha e André Araújo, pela ajuda e companheirismo que perdura desde o início do nosso percurso académico até à realização deste trabalho.

Por fim, uma palavra de agradecimento para os meus pais, António e Isabel, pela paciência, apoio e compreensão incondicional durante a elaboração desta dissertação.

# Resumo

A monitorização é uma tarefa essencial na gestão e engenharia das redes de comunicações atuais. Face a paradigmas como *Cloud Computing* e *Cloud Services*, os desafios colocados à monitorização de redes e serviços são ainda mais variados e exigentes. *Cloud Computing* inclui modelos de serviços distintos (IaaS, PaaS, SaaS), compartilhando algumas necessidades comuns na medição de infraestruturas, mas com especificidades de acordo com o tipo de serviço prestado e recursos envolvidos. A complexidade associada à monitorização destes ambientes e a falta de normas até agora, exige uma análise cuidadosa e sistematizada, de modo a obter uma melhor compreensão acerca dos pontos-chave envolvidos na avaliação dos serviços prestados.

Neste contexto, esta dissertação apresenta uma abordagem estratificada à monitorização de Serviços *Cloud*. O objetivo principal prende-se com a identificação das várias dimensões da monitorização de serviços *Cloud*, combinando as perspetivas do fornecedor de infraestruturas e de serviços, e dos clientes. Consequentemente, a monitorização do estado dos recursos, da qualidade de serviço, qualidade de experiência e contratos de serviço são aspetos a cobrir. Este processo envolve a identificação de parâmetros e métricas relevantes para cada dimensão monitorizada. Através de uma visão estratificada do problema, este estudo contribui para alcançar uma abordagem mais clara e eficiente à monitorização de serviços *Cloud*.

# Abstract

Monitoring is an essential task in the management and engineering of today's communication networks. To face paradigms such as cloud computing and cloud services, the challenges of monitoring networks and services are even more varied and demanding. Cloud Computing includes distinct service models (IaaS, PaaS, SaaS), sharing common needs in measuring the infrastructure, but with specificities according to the type of service and resources involved. The complexity of monitoring these environments and the lack of standards so far urge for a careful analysis, systematizing and understanding the key points involved when assessing the services provided.

In this context, this dissertation presents a layered approach to Cloud Services monitoring. The main objective concerns the identification of the multiple dimensions of monitoring cloud services, combining the perspectives of providers and customers. Consequently, the monitoring of resources, quality of service, quality of experience and service contracts are aspects to cover. This process involves the identification of relevant parameters and metrics for each monitoring dimension. Taking a stratified view of the problem, this study contributes to achieve a clearer and more efficient approach to cloud services monitoring.

# Conteúdo

# Lista de Figuras

# Lista de Tabelas

# Lista de Acrónimos

| | |
|---|---|
| API | Application Programming Interface |
| APIC | Advanced Programmable Interrupt Controller |
| ASP | Application Service Provider |
| AWS | Amazon Web Services |
| CaaS | Communications as a Service |
| CPU | Central Processing Unit |
| CRM | Customer Relationship Management |
| CSA | Cloud Security Alliance |
| CSV | Comma-separated Values |
| DDoS | Distributed Denial of Service |
| DNSSEC | Domain Name System Security Extensions |
| DoS | Denial of Service |
| DPI | Deep Packet Inspection |
| EBS | Elastic Block Storage |
| EC2 | Elastic Compute Cloud |
| HTTP | HyperText Transfer Protocol |
| IaaS | Infrastructure as a Service |
| ICMP | Internet Control Message Protocol |
| IDS | Intrusion Detection Systems |

| | |
|---|---|
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| IPER | IP Packet Error Ratio |
| IPLR | IP Packet Loss Ratio |
| IPPM | IP Performance Metrics |
| IPS | Intrusion Prevention Systems |
| IPTD | IP Packet Transfer Delay |
| ISP | Internet Service Provider |
| ITU-T | International Telecommunications Union - Telecommunication Standardization Sector |
| MOS | Mean Opinion Scores |
| NaaS | Network as a Service |
| NFS | Network File System |
| NNTP | Network News Transfer Protocol |
| OS | Operating System |
| OWD | One-way Delay |
| OWLP | One-way Loss Pattern |
| OWPL | One-way Packet Loss |
| PaaS | Platform as a Service |
| POP3 | Post Office Protocol |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| RAM | Random Access Memory |
| RTT | Round-trip Time |
| S3 | Simple Storage Service |
| SaaS | Software as a Service |

| | |
|---|---|
| SLA | Service Level Agreement |
| SMP | Symmetric Multiprocessing |
| SMTP | Simple Mail Transfer Protocol |
| SNMP | Simple Network Management Protocol |
| SPR | Spurious IP Packet Ratio |
| SSH | Secure Shell |
| SSL | Secure Sockets Layer |
| TELCOS | Telecommunications Companies |
| TI | Tecnologias de Informação |
| UEC | Ubuntu Enterprise Cloud |
| UPS | Uninterruptible Power Supply |
| VEE | Virtual Execution Environments |
| VEEH | VEE Host |
| VEEM | VEE Manager |
| VLAN | Virtual Local Area Network |
| VM | Virtual Machine |
| VoIP | Voice over IP |
| VPN | Virtual Private Network |
| XDR | External Data Representation |
| XML | Extensible Markup Language |

# Capítulo 1

# Introdução

Com os avanços significativos nas Tecnologias de Informação (TI) ao longo do último meio século existe cada vez mais a perceção de que a computação será um dia vista como o quinto utilitário (depois da água, eletricidade, gás e telefone). Estes serviços básicos essenciais (utilitários) são acedidos com tanta frequência que necessitam de estar disponíveis a qualquer altura, ou seja, sempre que o consumidor exigir [1]. Os consumidores estão assim em condições de pagar aos fornecedores de serviço, conforme o respetivo uso dos serviços. A computação, tal comos os outros quatro utilitários existentes, fornece o nível básico de serviço que é considerado essencial para atender às necessidades diárias da comunidade em geral. Deste modo, têm sido propostos alguns paradigmas de computação, dos quais o último é conhecido como ”computação em nuvem” (ou *Cloud Computing*, termo adotado ao longo desta dissertação, fruto da sua maior popularidade). Neste modelo, os utilizadores acedem aos serviços conforme as suas necessidades, sem levar em consideração onde é que os serviços estão hospedados ou como são entregues.

*Cloud Computing* surgiu recentemente como um tema de substancial interesse empresarial e académico, embora o seu significado, alcance e propósito em relação a outros paradigmas esteja a ser muito debatido. Para alguns investigadores, *Clouds* são uma evolução natural para a comercialização de sistemas *Grid*, enquanto que para outros podem ser vistas como uma nova versão das tecnologias *pay-per-use* já existentes. A partir de qualquer ponto de vista, a ”*Cloud*” tornou-se o rótulo de escolha para o acesso *pay-per-use* a uma ampla variedade de aplicações e recursos computacionais de terceiros em larga escala [2].

As mais-valias prometidas pela *Cloud Computing* estão na base do enorme impulso que a tecnologia tem presenciado. Como benefícios principais podemos identificar a facilidade de uso em termos de desenvolvimento, administração e manutenção, assim como a alta escalabilidade e

flexibilidade para criar novos serviços. Para além destes benefícios, *Cloud Computing* promete ainda vantagens económicas significativas para os utilizadores TI (principalmente em termos de redução de custos), através da mudança de serviços, computação e dados para instalações centralizadas e transparentes ou para fornecedores. A entrada em cena das principais organizações internacionais (e.g., Amazon, Google, Microsoft, Apple, IBM, Intel, HP, entre outras), deu um grande contributo para o crescimento exponencial da tecnologia.

## 1.1 Enquadramento e Motivação

Dada a complexidade presente num ambiente de *Cloud Computing*, é possível identificar alguns desafios presentes na tecnologia. Este tipo de complexidade aliada à falta de maturidade típica das tecnologias recentes, traduz-se em algumas limitações e questões por resolver. Um dos tópicos que ainda carece de uma maior investigação e investimento é precisamente a monitorização dos serviços *Cloud*, uma vez que a gestão destes serviços e a garantia de qualidade de serviço (QoS), são tidas como algumas das dificuldades do desenvolvimento da tecnologia. O fornecimento de serviços *Cloud Computing* requer processos de monitorização e gestão sofisticados, tendo como intuito assegurar aspetos relacionados com a QoS, desempenho, robustez, segurança, entre outros. A gestão dos serviços *Cloud* é sustentada por ações como visualização, controlabilidade e automação em ambientes virtuais. Este tipo de ações assume um papel importante no apoio à gestão da complexidade associada a *Cloud Computing*. A visualização é uma vantagem para um gestor de serviços, na medida em que ajuda a responder rapidamente a eventos e a tomar melhores decisões. Por sua vez, o controlo ajuda a gerir riscos e os custos de partilha, enquanto a automação ajuda a reduzir custos, trazendo agilidade às operações [3].

Neste momento, no contexto empresarial, os sistemas de monitorização são encarados como uma peça central de produtividade. Equipas especializadas em TI são responsáveis por monitorizar regularmente os sistemas, redes e aplicações, tentando garantir o correto funcionamento de uma organização. Por outro lado, dado o elevado número de intervenientes envolvidos num ambiente de *Cloud Computing*, a monitorização é também uma preocupação ao nível dos utilizadores e parceiros. Neste caso, existe o interesse em comprovar que usufruem do serviço adquirido conforme o acordo estabelecido com o respetivo fornecedor.

Este cenário leva à necessidade da adotar sistemas de monitorização gerais e amplos, onde devem ser incorporados os interesses dos vários intervenientes e os diversos componentes envolvidos. Neste contexto surge então a problemática abordada nesta dissertação. Num cenário

*Cloud Computing* torna-se essencial ter uma visão global de todos os aspetos que possam interferir na qualidade de um serviço prestado, devendo por sua vez ser tidos em conta no processo de monitorização e gestão. Uma visão estratificada da monitorização dos serviços *Cloud* é bastante útil na medida em que ajuda a ter uma melhor perceção sobre o que se passa a cada nível neste tipo de ambientes complexos. Como exemplo, perante um cenário de falha de um serviço, independentemente da sua natureza, podem ser provocados impactos financeiros indiretos, muitas vezes difíceis de calcular. Neste caso, o recurso a um modelo de monitorização estratificado pode revelar-se uma mais-valia na identificação da causa dos problemas.

## 1.2 Objetivos

Este trabalho tem como objetivo principal a definição de um modelo estratificado de monitorização de serviços *Cloud*, identificando as várias dimensões da monitorização e combinando as perspetivas do fornecedor de infraestruturas e serviços, e dos clientes. Portanto, a monitorização do estado dos recursos, da qualidade de serviço, qualidade de experiência (QoE) e contratos de serviço são aspetos a cobrir. Este processo envolve a identificação dos parâmetros e métricas relevantes para cada dimensão monitorizada. Por fim, depois do estudo dos métodos e ferramentas de monitorização disponíveis para este fim, será elaborado um cenário de teste de serviços de uso mais comum.

Definem-se como objetivos específicos:

(i) identificar as várias dimensões da monitorização e controlo de serviços *Cloud*;

da infraestrutura aos serviços;

perspetiva do fornecedor de serviços e do cliente;

monitorização de recursos, QoS/QoE e *Service Level Agreement* (SLAs);

(ii) identificar os parâmetros e as métricas relevantes para cada dimensão;

(iii) identificar as metodologias de medição e ferramentas disponíveis;

(iv) explorar a monitorização de um ambiente *Cloud*, aplicando algumas dessas ferramentas e classificando as métricas obtidas segundo o modelo proposto;

## 1.3 Sumário das Principais Contribuições

A principal contribuição do trabalho desenvolvido é reunir, esclarecer e sistematizar as principais questões envolvidas na monitorização *Cloud* e plataformas relacionadas, a fim de fundamentar e fomentar o desenvolvimento de serviços de monitorização abrangentes e flexíveis.

A monitorização *Cloud* é uma área de pesquisa recente e ativa, onde a falta de padrões relacionados é evidente. Outra das contribuições do trabalho desenvolvido enquadra-se nos esforços dedicados à normalização, propondo uma abordagem estratificada, identificando e sugerindo parâmetros, métricas e boas práticas para uma monitorização eficiente dos serviços e ambientes *Cloud Computing*.

Como fruto deste trabalho, foram elaborados e submetidos dois artigos científicos na "12ª Conferência de Redes de Computadores" [4], já aceite para publicação, e na "IEEE ICC 2013 - Next-Generation Networking Symposium" [5], encontrando-se o mesmo em processo de revisão.

## 1.4 Organização da Dissertação

A estrutura desta dissertação está organizada em 5 capítulos. A descrição de cada capítulo é a seguinte:

**Capítulo 1:** no presente capítulo apresenta-se o enquadramento e contextualização do trabalho, assim como a motivação que levou ao seu desenvolvimento, objetivos pretendidos e uma síntese das principais contribuições resultantes do trabalho final.

**Capítulo 2:** neste capítulo é feita uma abordagem ao estado da arte na área de *Cloud Computing*, onde é apresentado o conceito e uma breve referência sobre a evolução da tecnologia. São apresentadas também as principais características e os benefícios gerais na perspetiva dos fornecedores, parceiros e utilizadores dos serviços *Cloud*. De seguida são apresentados os vários modelos de serviço e os modelos de implementação *Cloud*. Por fim são abordadas algumas questões relacionadas com a monitorização *Cloud*.

**Capítulo 3:** neste capítulo é abordada a monitorização de serviços *Cloud*. Em primeiro lugar é apresentada a abordagem estratificada proposta, onde é definido um modelo que tem como objetivo abranger as várias dimensões envolvidas na monitorização de serviços *Cloud*. De seguida são abordadas em maior detalhe as camadas que constituem o modelo (Infraestrutura, Rede, Serviço/Aplicação e Cliente/Fornecedor), onde são propostos parâmetros e métricas para a

monitorização, assim como boas práticas a levar em consideração. Posteriormente são discutidas algumas questões relacionadas com QoE resultantes da utilização de um serviço *Cloud*. Por fim é estabelecida uma relação entre o modelo proposto e os modelos de serviço.

**Capítulo 4:** neste capítulo são apresentadas algumas plataformas e ferramentas de monitorização de *Cloud Computing* existentes no mercado. Em primeiro lugar são abordadas duas estratégias de monitorização (PCMONS e Lattice). De seguida são apresentadas algumas ferramentas de monitorização, agrupadas segundo o paradigma a que obedecem (monitorização local, remota ou plataformas de gestão *web*).

**Capítulo 5:** neste capítulo é apresentado um cenário prático, onde são utilizadas duas das ferramentas de monitorização abordadas no capítulo anterior. O objetivo passa por explorar a monitorização de um ambiente *Cloud*, classificando as métricas obtidas segundo o modelo proposto. É apresentado o ambiente de testes, a metodologia utilizada e por fim é feita a análise dos resultados.

**Capítulo 6:** neste último capítulo são apresentadas as principais conclusões resultantes do trabalho e pesquisa efetuada, assim como as principais contribuições desta dissertação. Por fim, é feita uma análise ao possível trabalho futuro associado a este tema.

# Capítulo 2

# Cloud Computing: Estado da arte

Neste capítulo é apresentado o estado da arte na área de *Cloud Computing*. Em primeiro lugar é apresentada uma definição da tecnologia e uma breve referência à sua evolução, seguida da identificação das suas principais características. Posteriormente são apresentados os benefícios associados a *Cloud Computing*, segundo as perspetivas dos fornecedores, parceiros e utilizadores dos serviços. São também descritos os modelos de serviço e os modelos de implementação existentes. Por fim são abordadas algumas questões relacionadas com a monitorização *Cloud*.

## 2.1 Introdução

Hoje em dia temos presenciado um crescimento da oferta de ferramentas baseadas em *Cloud Computing*. Este facto deve-se sobretudo à descida dos custos associados à tecnologia, fruto do aumento da concorrência, assim como às vantagens a ela associadas. Estes fatores têm-se traduzido num aumento exponencial da tecnologia.

Em primeiro lugar é importante definir o conceito de *Cloud*. Devido à existência de várias definições, foi tida como base a definição proposta em [6], resultante da extração das características essenciais consensualmente presentes em mais de 20 definições. Posto isto, as *Clouds* são um grande conjunto de recursos virtualizados facilmente utilizáveis e acessíveis (como *hardware*, plataformas de desenvolvimento e/ou serviços). Esses recursos podem ser dinamicamente reconfigurados para se ajustarem a uma carga variável (escalável), permitindo também uma melhor utilização dos recursos. Este conjunto de recursos é tipicamente explorado por um modelo *pay-per-use* (também conhecido por *pay-as-you-go*), em que as garantias são oferecidas pelo

fornecedor da infraestrutura, por meio de SLAs personalizados.

Apesar de toda a atenção em redor de *Cloud Computing*, provocada pelas razões acima referidas em conjunto com o interesse da *mídia* e os esforços de *marketing* das grandes empresas, esta é uma tecnologia que se baseia em conceitos que não são propriamente novos (como a virtualização e a computação distribuída). Esta tecnologia resulta de um processo evolutivo que começou há sensivelmente 20 anos, em parte com a tecnologia chamada "*Grid Computing*". Contudo ainda existe algum caminho a percorrer antes de se ver uma ampla adoção da *Cloud* [6].

Nos últimos anos, *Cloud Computing* tem sido introduzida como uma nova abordagem para a prestação de serviços de *software* através da rede. Durante os últimos cinco anos, o modelo *Software as a Service* (SaaS) foi considerado um novo conceito relativo ao acesso a uma aplicação de *software*. No modelo SaaS, o *software* não está instalado localmente nos servidores do cliente, sob os termos de uma licença específica. Este é acedido *online*, através de uma rede e com recurso a um *web browser*. Um fornecedor de SaaS tem algumas semelhanças com um *Application Service Provider* (ASP), que por sua vez foi introduzido nos inícios de 2000 como uma evolução de um *Internet Service Provider* (ISP). O novo termo "*Cloud*" foi introduzido para designar as características "*online*" e "entrega através da rede" do SaaS. Os desenvolvimentos recentes ao nível do acesso de alto débito e a melhoria da disponibilidade da camada de rede pelos principais ISPs, podem ser vistos como o principal ponto de partida para o mercado *Cloud* emergente. Se considerarmos *Cloud Computing* como uma evolução do ASP e uma generalização dos SaaS, com uma extensão às plataformas e infraestruturas, esta pode ser também considerada como uma nova abordagem para a implementação e entrega de computação em rede [7].

De maneira semelhante às "nuvens" físicas, *Cloud Computing* também pode ter diferentes tamanhos e formas. A Figura 2.1 ilustra o modelo visual de *Cloud Computing*, onde é possível identificar os vários modelos em que pode ser implementada (*Public*, *Private*, *Hybrid* e *Community*), assim como os principais modelos de serviço (IaaS - *Infrastructure as a Service*, PaaS - *Platform as a Service* e SaaS - *Software as a Service*). As características essenciais da tecnologia encontram-se igualmente ilustradas.

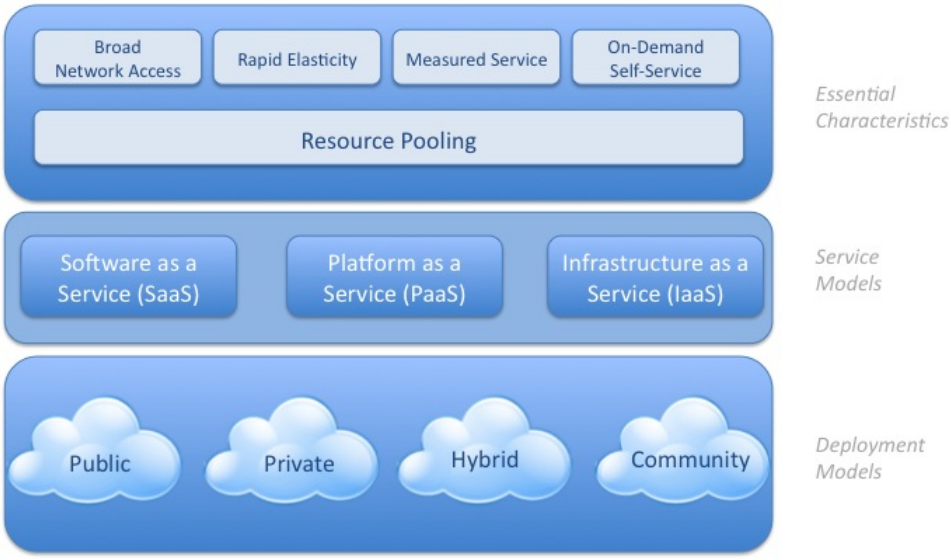



Figura 2.1: Cloud Computing - Modelo Visual [8].

## 2.2 Principais Características

Os pontos fortes da tecnologia prendem-se sobretudo com o fornecimento de armazenamento e processamento de dados, onde os clientes podem correr aplicações que consomem bastante poder de processamento e memória, sem que para tal tenham de possuir tais capacidades no seu computador pessoal. Estes serviços são bastantes versáteis e o cliente apenas paga em função dos recursos que usar ou tempo de utilização. Na base do funcionamento desta tecnologia está um conjunto de características essenciais e que a distingue dos serviços tradicionais. Na Figura 2.1 estão ilustradas as principais características associadas a *Cloud Computing* (amplo acesso à rede, elasticidade rápida, serviços mensuráveis, auto-atendimento a pedido e *pool* de recursos), abordadas em maior detalhe de seguida em conjunto com outras igualmente relevantes [9, 10, 11].

- **Amplo acesso à rede**:

  Um acesso amplo à rede significa que um serviço *Cloud* está disponível através da rede e está acessível através de qualquer plataforma. São utilizados mecanismos normalizados que permitem o acesso através de diversos dispositivos (e.g., estações de trabalho, portáteis, telemóveis, *tablets*, etc.). Com o surgimento de conceitos e de realidades como o

acesso à Internet em qualquer lugar e a qualquer altura, esta torna-se uma característica que contribui para a aceitação e expansão da tecnologia.

- **Elasticidade Rápida**:

  Neste contexto a elasticidade é definida como a habilidade de expansão dos serviços, ou seja, a capacidade de estes escalarem de forma crescente ou decrescente, conforme as necessidades. Esta elasticidade permite que possam ser alocados ao utilizador mais ou menos recursos no momento em que for necessário. Isto faz com que a *Cloud* pareça infinita da perspetiva do utilizador. Esta característica abrange os diversos serviços *Cloud* ao dispor do utilizador, desde infraestruturas a plataformas e *software*.

  A elasticidade é um dos grandes desafios das infraestruturas tradicionais das TI, na medida em que é difícil dimensionar os recursos envolvidos consoante a alteração dos pedidos de uma aplicação. São várias as ocasiões em que existe uma sobrecarga dos recursos, dada a incapacidade de lidar com picos de carga em serviços onde existe geralmente uma baixa utilização de recursos. A situação contrária, onde são adquiridos recursos em excesso para a utilização normal de um serviço, mas suportando picos de carga que possam surgir, acaba por ser pouco viável a nível económico, ambiental, etc.

  Posto isto, esta é uma das principais características que contribuem para que a *Cloud Computing* seja um serviço muito atrativo.

- **Serviços Mensuráveis**:

  Todos os serviços *Cloud* são controlados e monitorizados, de maneira a que tudo fique transparente quer para o fornecedor quer para o utilizador do serviço. Este é um processo crucial para aspetos como faturação, controlo de acessos, otimização de recursos, planeamento, entre outras tarefas.

  À medida que as *Clouds* se vão tornando mais elásticas, a necessidade da existência de um serviço mensurável é cada vez mais evidente.

- **Auto-Atendimento a Pedido**:

  O utilizador possui a capacidade de utilizar os serviços *Cloud* conforme a sua necessidade, ou seja, pode aumentar ou diminuir automaticamente os recursos utilizados. Este é um processo *self-service*, onde não existe a necessidade de interação humana entre o utilizador e o fornecedor do serviço.

O processo por detrás do aumento de capacidade num modelo TI tradicional normalmente envolve diversos aspetos, desde orçamentos, aquisições, planeamentos, recrutamento de pessoal especializado, entre outros, para além de ser um processo moroso (meses ou mesmo anos). Contrastando com este cenário, a capacidade de auto-atendimento nos serviços *Cloud*, permite que o utilizador obtenha os recursos de que necessita com um simples pedido. Este é um processo rápido e é tratado em questões de minutos.

- ***Pool* de Recursos**:

  Os recursos computacionais de um fornecedor de serviços *Cloud* estão reunidos geograficamente e servem os vários consumidores através de um modelo *multi-tenant*. Os recursos físicos e virtuais são atribuídos e reatribuídos dinamicamente, de acordo com os pedidos dos utilizadores. *Multi-tenancy* é uma característica essencial de um sistema *Cloud*, na medida em que proporciona o isolamento de diferentes utilizadores (*tenants*), enquanto maximiza a partilha de recursos entre eles. No que toca à localização dos recursos, os utilizadores geralmente não possuem controlo ou conhecimento sobre a sua exata localização. Apenas têm acesso a informações com um alto nível de abstração, como o país, estado ou *datacenter* onde estes se encontram.

  Como exemplo, ao nível das aplicações, a característica *multi-tenancy* permite que uma única instância possa satisfazer vários utilizadores ao mesmo tempo. Este processo resulta na capacidade de consolidação de vários utilizadores dentro de uma base de dados e no aproveitamento de custos.

Para além destas características essênciais, é possível identificar um outro conjunto de características relevantes, como a disponibilidade, fiabilidade, modelo de pagamento, flexibilidade, interoperabilidade e portabilidade.

- **Disponibilidade**:

  Dependendo do serviço e do respetivo fornecedor em questão, os utilizadores estão perante um serviço com alta disponibilidade. Tendo como exemplo o caso em que um servidor deixar de funcionar, os restantes servidores que constituem o sistema asseguram a continuação do funcionamento do serviço.

- **Fiabilidade**:

  Os utilizadores de serviços *Cloud* procuram que os seus serviços tenham uma QoS fim-a-fim assegurada, assim como altos níveis de fiabilidade e de disponibilidade contínua da parte dos fornecedores. Os SLAs devem ser estruturados de maneira a abranger estes aspetos, para além de outros igualmente relevantes.

- **Modelo de pagamento**:

  O modelo de pagamento associado a *Cloud Computing* é o flexível e atrativo modelo *pay-as-you-go*. O utilizador apenas paga pelos recursos que utilizar ou pelo tempo de utilização. Não existe a necessidade de pagar por uma licença de uso integral, tal como sucede com o fornecimento de *software* tradicional. Este modelo contribui assim para um melhor controlo dos custos, quando estão envolvidas aplicações *Cloud* que não sejam gratuitas.

- **Flexibilidade**:

  A arquitetura *Cloud* tem a capacidade de suportar múltiplos modelos de implementação, assim como diversas categorias de modelos de serviço.

- **Interoperabilidade**:

  A interoperabilidade diz respeito à capacidade dos sistemas comunicarem. É expectável que estejam disponíveis especificações bem documentadas e testadas, de modo a possibilitar o trabalho conjunto entre sistemas heterogéneos de um ambiente *Cloud*.

- **Portabilidade**:

  A portabilidade está relacionada com a capacidade de execução de componentes ou sistemas escritos para um ambiente em outro ambiente. Os ambientes podem ir desde *hardware* físico ou virtual até *software*.

## 2.3 Benefícios Gerais

Os benefícios associados a *Cloud Computing* podem ser considerados de diferentes perspetivas, tendo em conta os diversos intervenientes num ambiente *Cloud*. De seguida são abordados alguns dos benefícios na perspetiva dos fornecedores, parceiros e utilizadores dos serviços [7].

### 2.3.1 Benefícios da perspetiva do fornecedor de serviço

Da perspetiva dos fornecedores de serviço, os benefícios de que usufruem podem ser os seguintes:

**Redução de custos**: processos como a virtualização e o fornecimento de *software* permitem uma alocação de recursos computacionais, diminuindo assim os custos despendidos em *hardware*. Os fornecedores podem localizar ainda as instalações em locais de baixo custo.

**Melhoria do custo total de propriedade e de risco**: os investimentos são deslocados da despesa de capital inicial (CapEX) para as despesas operacionais (OpEX) no consumo de recursos TI. Existe um aumento da utilização da capacidade dos ativos TI.

**Infraestruturas escaláveis e flexíveis**: um mecanismo altamente escalável permite a construção de serviços escaláveis para os utilizadores e parceiros. Uma infraestrutura escalável permite satisfazer os pedidos para picos de carga e para variações sazonais.

**Eficiência e flexibilidade na gestão de recursos**: os fornecedores podem utilizar recursos mais eficientes e flexíveis (recursos das TI, armazenamento, rede ou servidores), através da utilização da tecnologia de virtualização na *Cloud Computing*.

**Agilidade negocial com implementação de um serviço rápido**: a prestação de serviços é efetuada com um menor custo através do uso eficiente e gestão dos recursos envolvidos. Quanto mais rápido um fornecedor executar uma tarefa administrativa, mais expedito se move um negócio, aliado a uma redução dos custos.

**Fiabilidade do serviço com alta disponibilidade**: a redundância resultante das cargas de trabalho poderem ser distribuídas por várias instalações, ou mesmo através de *Clouds*, contribui para o aumento da disponibilidade de um serviço, assim como reduz o tempo de inatividade. As estratégias de distribuição de recursos podem ainda ajudar na recuperação em caso de ocorrência de desastres naturais (e.g., terramotos, tsunamis, inundações, etc.).

**Suporte a negócios com terceiros**: um fornecedor de serviço está inserido num mercado que permite a inclusão de diversos intervenientes independentes (fornecedores de *software*, programadores, integradores, clientes corporativos, utilizadores finais, entre outros).

**Eficiência energética**: resultante do potencial da tecnologia envolvida no ambiente *Cloud Computing* em poupar quantidades de energia significativas. Os aspetos relacionados com o *hardware* e virtualização têm sido focados neste âmbito, havendo ainda espaço para explorar a operação dos sistemas e os aspetos de rede.

**Capacidades de agregação**: os fornecedores podem utilizar uma plataforma *Cloud* para agregar as suas capacidades às de terceiros, integrando a cadeia da indústria e aumentando a fidelidade dos clientes.

### 2.3.2 Benefícios da perspetiva dos parceiros

Num ambiente *Cloud Computing* existem diversos parceiros dos fornecedores de serviço, destacando-se os fornecedores de rede, *software*, equipamentos e integradores de serviços. Os benéficos dos respetivos parceiros encontram-se descritos de seguida.

**Fornecedor de rede**: fornecendo um acesso simples de rede a serviços integrados e atualizados, surgem novas oportunidades de negócios. Os utilizadores para aumentarem a sua dependência de acesso vão melhorar a largura de banda de acesso, trazendo novos rendimentos.

**Fornecedor de Software**: muitos dos principais fornecedores de soluções de *software* desenvolvem extensões a partir de serviços existentes. As suas linhas de produção estão assim perante um crescimento de novas receitas.

**Fornecedor de equipamento**: as atualizações ao nível do *hardware* em equipamentos (servidores, armazenamento, rede) e a aquisição de equipamentos relacionados vão aumentar. Os fornecedores de equipamentos devem reforçar as suas reservas, assim como investir na pesquisa, com vista a fornecerem sistemas cada vez mais completos e que suportem as exigências dos diversos serviços *Cloud*.

**Integradores de sistemas**: a necessidade de soluções em sistemas de integração para os fornecedores de serviços *Cloud* vai abrir novos mercados para os integradores de sistemas.

### 2.3.3 Benefícios da perspetiva dos utilizadores

No que toca aos utilizadores, os seus benefícios podem resumir-se nos seguintes:

**Fornecimento otimizado e rápido**: os utilizadores estão perante uma disposição imediata dos serviços mais recentes através da rede. Existe uma oferta de serviços capaz de satisfazer em específico cada processo de indústria ou negócio.

**Aplicações disponíveis em qualquer local**: os serviços possuem a característica de poderem ser utilizados *online* em qualquer dispositivo. Os dispositivos móveis também são suportados, abrangendo assim clientes com elevados níveis de mobilidade e possibilitando a realização de

negócios em qualquer parte do mundo.

**Preços conforme a utilização**: num ambiente *Cloud Computing* deixam de existir despesas iniciais significativas resultantes da aquisição e manutenção do *hardware*. Os utilizadores pagam apenas consoante os recursos que utilizam ou tempo de utilização, ou seja, é adotado o modelo *pay-as-you-go* anteriormente referido.

**Baixos custos de migração**: em caso de insatisfação dos utilizadores, é relativamente fácil para eles mudarem para uma solução concorrente, bastando assinar um novo contracto, transferir os dados e atualizar os intervenientes. Este aspeto contrasta com as soluções instaladas localmente, onde seria necessário pagar por novas licenças e passar por novas e longas implementações.

**Proteção de dados importantes**: a *Cloud Computing* permite que seja mais fácil efetuar *backup* e armazenamento de dados importantes em vários locais. Mesmo perante a ocorrência de desastres naturais, desde terramotos a inundações e tsunamis, os dados continuam armazenados em segurança.

## 2.4 Modelos de Serviço

Os serviços *Cloud Computing* estão divididos em modelos de serviço, de acordo com a sua natureza. Os três principais modelos que estruturam a arquitetura *Cloud* são os seguintes: IaaS, PaaS e SaaS (Figura 2.2). Os serviços relativos às infraestruturas são considerados a camada inferior, seguidos pelas plataformas de desenvolvimento. As aplicações são disponibilizadas ao utilizador final e residem no topo da pilha *Cloud*. Devido aos três modelos de serviço referidos serem os mais comuns, estes serão os modelos abordados ao longo desta dissertação. Contudo, recentemente os *standards* estabelecidos pelo *Focus Group on Cloud Computing* do *International Telecommunications Union - Telecommunication Standardization Sector* (ITU-T) definem ainda mais dois modelos de serviço, nomeadamente CaaS (*Communications as a Service*) e NaaS (*Network as a Service*). De seguida seguem-se as definições e as principais características dos modelos de serviço referidos, assim como alguns exemplos de serviços.

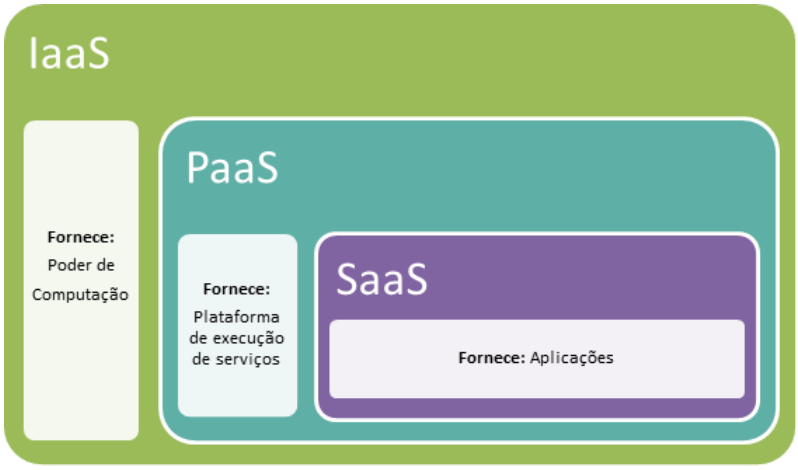



Figura 2.2: Modelos de Serviço.

### 2.4.1 IaaS

**Infrastructure as a Service (IaaS)**: neste modelo de serviço, o fornecedor do serviço *Cloud* disponibiliza aos clientes recursos computacionais virtualizados. Os recursos fornecidos podem ir desde poder de processamento, armazenamento, comunicação e serviços de conexão de rede (e.g., *Virtual Local Area Network* (VLAN), Firewall, balanceadores de carga, etc), a outros recursos computacionais fundamentais de uma infraestrutura *Cloud*. Os utilizadores podem desenvolver e executar *software* arbitrário, variando desde aplicações até *Operating Systems* (OS). Este modelo de serviço pode operar em *Clouds* do tipo *Public*, *Private*, *Community* ou *Hybrid* (ver Secção 2.5). O utilizador do serviço *Cloud* não pode controlar nem gerir os recursos da infraestrutura *Cloud* subjacente, mas possui controlo sobre OS, armazenamento, aplicações desenvolvidas e alguns componentes de rede [2, 9, 11].

### 2.4.2 PaaS

**Platform as a Service (PaaS)**: neste tipo de modelo de serviço é fornecido um conjunto de ferramentas para o desenvolvimento e disponibilização de aplicações *online*, sem que haja preocupações com a sua hospedagem. Os utilizadores podem desenvolver e implementar as suas próprias aplicações (adquiridas ou desenvolvidas por si) na infraestrutura *Cloud*. As ferramentas disponibilizadas fornecem serviços bem integrados e altamente especializados, incorporando tudo o que um programador necessita nas áreas de desenvolvimento, teste, hospedagem, publica-

ção e manutenção de aplicações. O utilizador possui controlo sobre as aplicações desenvolvidas, e possivelmente sobre algumas configurações do ambiente de hospedagem, mas não controla nem gere a infraestrutura *Cloud* subjacente (OS, *hardware* e rede). Resumidamente, a plataforma é uma *framework* de aplicação típica e ações como desenvolver, compilar, *debug* e testar uma aplicação, são agora executadas dentro de um ambiente disponibilizado pelo fornecedor do serviço [2, 9, 11].

### 2.4.3 SaaS

**Software as a Service (SaaS)**: neste modelo de serviço o objetivo passa por disponibilizar aos clientes, aplicações que correm sobre uma infraestrutura *Cloud*. O ambiente de execução é a Internet e as aplicações estão acessíveis através de vários dispositivos clientes, com recurso a interfaces como um *web browser*. As aplicações podem ser de diversos tipos, desde as aplicações TI até às comerciais. Com a exceção de algumas configurações de aplicações específicas dos utilizadores, estes não controlam nem gerem a infraestrutura *Cloud* subjacente. O utilizador não está necessariamente a pagar pela compra de um sistema, ou seja, está apenas a adquirir o direito de utilizar um serviço, o que na sua essência é um *software* como muitos outros existentes [2, 9, 11].

### 2.4.4 CaaS e NaaS

**Communications as a Service (CaaS)**: nesta categoria de modelos de serviço, os serviços disponibilizados aos clientes possuem características que permitem a utilização de comunicação em tempo real e de serviços colaborativos. Como exemplo deste tipo de serviços encontra-se o *Voice over IP* (VoIP), videoconferências ou mensagens instantâneas em diferentes dispositivos [9].

**Network as a Service (NaaS)**: modelo de serviço onde são fornecidos aos utilizadores serviços de transporte ou de conectividade de rede dentro da *Cloud*. Estão aqui incluídos serviços como *Virtual Private Networks* (VPN) flexíveis, largura de banda a pedido, entre outros [9].

### 2.4.5 Exemplos de serviços comerciais

Devido a ser um mercado emergente, a oferta de serviços baseados na *Cloud* tem aumentado, verificando-se a existência de diversos fornecedores. Contudo, quando existe a necessidade de procurar uma solução, nem sempre é fácil ter uma noção real da oferta disponível e da respetiva qualidade do serviço. A forte concorrência que se começa a fazer sentir leva também à necessidade de um estudo mais detalhado do mercado. Na Tabela 2.1 estão ilustrados alguns exemplos de serviços classificados segundo os modelos discutidos, assim como a relação Vendedor/Comprador de cada modelo de serviço. De seguida são abordadas em maior detalhe algumas das soluções comerciais mais populares disponíveis para os vários modelos de serviço.

**IaaS:**

**Amazon Web Services (AWS):** quando estamos perante o modelo de serviço IaaS é praticamente impossível desassociar a AWS [12], uma vez que foram os pioneiros no desenvolvimento deste tipo de soluções. Atualmente são os líderes do mercado neste sector. A AWS oferece um conjunto de serviços que no seu todo formam uma plataforma *Cloud Computing*. Os seus serviços centrais e mais populares estão relacionados com as áreas da computação (EC2 - *Elastic Compute Cloud*) e do armazenamento (S3 - *Simple Storage Service*) [12].

**Computação**: o serviço disponibilizado pela AWS na área da computação é o EC2. São disponibilizadas *Virtual Machines* (VM) que podem ser customizadas de maneira semelhante a servidores físicos e permitem aos utilizadores correrem as suas próprias aplicações. Através de uma interface *web*, o utilizador pode iniciar instâncias com uma variedade de sistemas operativos. Este tipo de serviço oferece uma capacidade de computação redimensionável na *Cloud*, resultante das suas características escaláveis e elásticas. Elasticidade que permite aos serviços ”esticarem” de acordo com as necessidades dos utilizadores. O pagamento é efetuado com base no modelo *pay-as-you-go*, evitando assim despesas com planeamento, aquisição e manutenção de *hardware*, encarregues por sua vez à Amazon.

**Armazenamento**: na área do armazenamento a AWS disponibiliza o S3, caracterizado por ser um serviço de interface *web* onde é possível armazenar e recuperar qualquer tipo de dados. Os dados podem ter de tamanho desde 1 Byte até 5 TBytes e estão disponíveis em qualquer lugar a qualquer altura. A região onde os dados serão armazenados pode ser escolhida pelo utilizador, podendo este usufruir de uma otimização da latência e redução de custos, para além

de questões legais. Existem mecanismos de autenticação para assegurar que os dados apenas são acedidos por utilizadores autorizados, conforme os direitos que lhes foram atribuídos. Os dados podem ser acedidos com simples pedidos HTTP ou através do protocolo BitTorrent. Através deste tipo de serviços, os utilizadores deixam de ter preocupações relacionadas com os espaços de armazenamento, manutenção, acesso ou segurança dos dados.

**PaaS:**

**Google App. Engine:** relativamente ao modelo de serviço PaaS, um dos serviços pioneiros foi precisamente o Google App. Engine [13]. Através do Google App. Engine os utilizadores podem executar as suas aplicações *web* na infraestrutura do Google. As aplicações são fáceis de criar, manter e escalar à medida que o tráfego e armazenamento de dados precisem de crescer. Após a criação de uma aplicação, esta pode ser disponibilizada a partir do domínio do proprietário ou de um nome livre no domínio *appsot.com*. Existe ainda a possibilidade de ser partilhada ou ser restrita a um grupo de membros de uma organização. Na criação das aplicações são suportadas várias linguagens de programação, sendo o ambiente de execução em Java e Python.

Por sua vez, o modelo de pagamento também é um modelo *pay-as-you-go*, onde o utilizador apenas paga em conformidade com a utilização. Não existem preços predefinidos nem taxas recorrentes. A utilização deste serviço pode ser gratuita, porém com algumas limitações. Os utilizadores podem usufruir de armazenamento até 500 MBytes e de CPU e largura de banda suficientes para suportar uma aplicação com eficiência (até cerca de 5 milhões de visualizações por mês). Ao ativar a versão paga, os limites gratuitos aumentam, sendo apenas pagos os recursos que ultrapassarem esses limites.

**SaaS:**


**Microsoft Office 365:** esta ferramenta desenvolvida pela Microsoft [14] oferece um conjunto de serviços *online* e está disponível a empresas de todos os tamanhos. Na aquisição desta ferramenta não existe a necessidade de investimentos em infraestruturas de suporte a TI, como servidores. Pode ser acedida praticamente em qualquer lugar e dispositivo.

Os produtos disponibilizados são o *Office Web Apps*, *Exchange Online*, *SharePoint Online*, *Lync Online* e o *Office Professional Plus*. O *Office Web Apps* oferece através de um *browser*

versões do Microsoft Excel, Word, PowerPoint e OneNote, permitindo a edição e visualização de documentos *Office* em navegadores *web*. O *Exchange Online* fornece email (até 25 GBytes de armazenamento por utilizador), calendários e contactos. Quanto ao *SharePoint Online*, tem como base a colaboração, partilha e edição de documentos. O *Lync Online* fornece recursos e funcionalidades avançadas de comunicação para as organizações. Por sua vez o *Office Professional Plus* disponibiliza o mesmo *software* cliente que a Microsoft oferecia através dos seus volumes licenciados.

Quanto ao modelo de pagamentos, existem diversos planos, entre os quais para pequenos negócios, médios negócios e empresas, educação e organizações governamentais.

Tabela 2.1: Exemplos de Serviços.

| **Modelos de Serviço** | **Exemplos** | **Vendedores** | **Compradores** |
|---|---|---|---|
| IaaS | Amazon Web Services, Rackspace, Microsoft Hyper-VGoGrid, Proofpoint, RightScale, VMWare VCloud, EMC, IBM (Blue Cloud), Sun (Project Carloine), HP Adaptative Infra. as a Service, Windows Azure... | Fornecedores de Datacenters | Empresas |
| PaaS | Google App. Engine, Windows Azure, Salesforce, dotCloud, Redhat, AT&T, Cloudera, Oracle, Cloud Foundry ... | Fornecedores de Plataformas de Serviço | Companhias de Desenvolvimento de Software |
| SaaS | Office 365, Salesforce - CRM, Google Apps., Yahoo (Zimbra), Concur, Taleo, Netsuite, Dropbox, Proofpoint, Workday, Hotmail... | Companhias de Software | Utilizadores Finais |

### 2.4.6 Exemplo de uma solução IaaS open source

No contexto do modelo de serviço IaaS, esta secção apresenta uma solução *open source* e *freeware* que permite o desenvolvimento e implementação de um serviço IaaS. Existe atualmente um conjunto de ferramentas populares com este tipo de características, como por exemplo o Openstack, Eucalyptus, OpenNebula e Nimbus. De seguida são descritas as principais características da ferramenta Openstack, uma vez que é a solução utilizada na parte prática desta dissertação, por sua vez abordada no Capítulo 5.

Openstack [15] é um sistema operativo *Cloud open source* que controla grandes quantidades de recursos de computação, armazenamento e de rede num *datacenter*. Estes recursos são geridos através de um painel, providenciando controlo aos administradores, enquanto capacita os seus utilizadores para fornecerem recursos através de uma interface *web*. Na Figura 2.3 está ilustrada a arquitetura Openstack, onde é possivel identificar os diversos blocos que constituem a ferramenta.

No componente de computação (*Nova*), a ideia passa por permitir às empresas e fornecedores de serviços oferecerem recursos de computação a pedido, fornecendo e gerindo grandes redes de máquinas virtuais. Este tipo de recursos são acessíveis através de uma *Application Programming Interface* (API) para os programadores de aplicações *Cloud*, e através de interfaces *web* para os administradores e utilizadores.

Quanto à componente de armazenamento (*Swift*), face à variedade de necessidades (diferentes requisitos, desempenho ou preço), a Openstack suporta o armazenamento por objetos e por blocos. O armazenamento por objetos fornece uma API totalmente distribuída e pode ser integrada diretamente nas aplicações ou ser utilizada para *backup*, arquivo ou retenção de dados. O armazenamento por blocos permite que os dispositivos em bloco sejam expostos e conectados a instâncias de computação para armazenamento expandido, permitindo a integração com plataformas de armazenamento de empresas.

Relativamente à componente de rede, o Openstack fornece uma API orientada para a gestão de redes e de endereços IP, onde os utilizadores podem efetuar as suas configurações de rede. O objetivo deste módulo passar por evitar que a rede seja um potencial *bottleneck* ou um fator limitante na implementação de uma *Cloud*.

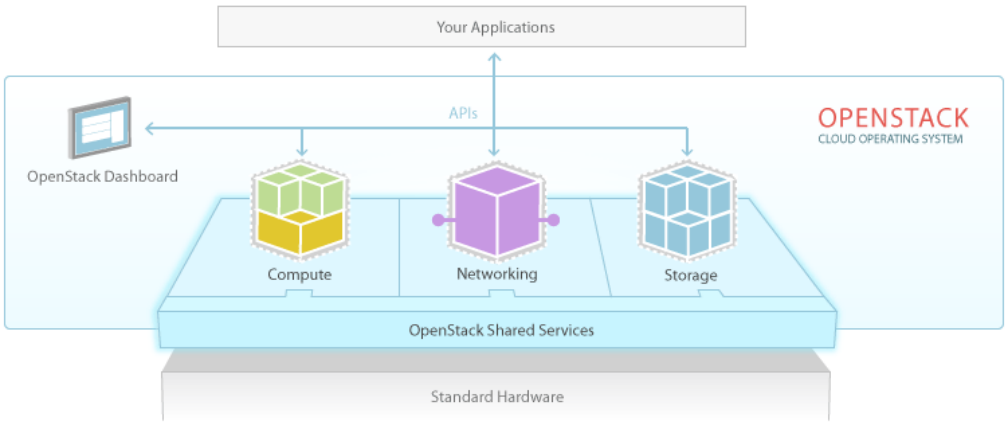



Figura 2.3: Arquitetura Openstack [15].

## 2.5 Modelos de Implementação

Uma vez abordados os principais modelos de serviço que constituem a pilha *Cloud*, seguem-se as formas básicas em que os serviços *Cloud* podem ser implementados, de uma perspetiva mais organizacional. Os modelos de implementação mais conhecidos são os quatro seguintes: *Public Cloud*, *Private Cloud*, *Community Cloud* e *Hybrid Cloud* (Figura 2.4). Nas *Public Clouds*, a principal atividade passa pela venda de serviços *online*, enquanto as *Private Clouds* são implementadas no interior das empresas e as *Community Clouds* são partilhadas por várias organizações. As *Hybrid Clouds* são uma combinação das características dos modelos anteriores. Seguidamente são abordados em maior detalhe cada um dos modelos de implementação.

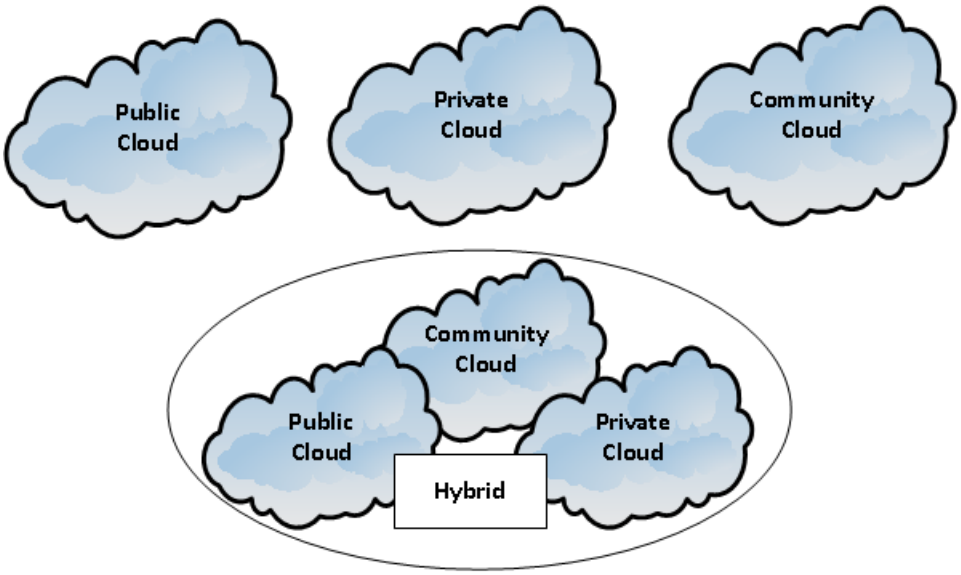



Figura 2.4: Modelos de Implementação.

### 2.5.1 Public Cloud

As *Public Clouds* são caracterizadas por disponibilizarem os seus recursos ao público em geral ou a um grande grupo de indústrias, através da Internet. Os utilizadores deste modelo utilizam serviços que são disponibilizados por organizações especializadas na venda de serviços *Cloud*. Neste caso o termo "*public*" nem sempre significa que o serviço não tem custos, caracterizando apenas o modo de acesso à sua interface. Os recursos são partilhados por todos os utilizadores,

que por sua vez não possuem controlo nem gerem a infraestrutura *Cloud* subjacente onde os serviços estão hospedados. As *Public Clouds* tiram partido dos benefícios gerais associados a *Cloud Computing*, nomeadamente da escalabilidade, elasticidade e do modelo de pagamento. Os clientes alugam o acesso aos recursos baseado num modelo *pay-as-you-go*, sem a necessidade de um grande investimento financeiro em infraestruturas como um *datacenter*.

Uma das preocupações inerentes a este modelo é a segurança, uma vez que o ambiente é partilhado e gerido pelo fornecedor do serviço *Cloud*. O utilizador possui pouco controlo sobre este aspeto. Contudo, o conceito por detrás de uma *Public Cloud* não implica que os dados e informações do utilizador estejam acessíveis publicamente. Os fornecedores deste tipo de serviços normalmente implementam mecanismos de controlo de acesso nos seus serviços.

Os primeiros tipos de *Clouds* foram na sua maioria *Public Clouds*, onde a Google, Amazon e Salesforce são alguns dos mais notáveis exemplos de empresas que começaram a explorar a área [9, 10, 11].

### 2.5.2 Private Cloud

Inicialmente eram muitos os cépticos que defendiam que o conceito de *Private Cloud* se tratava de um conceito impossível e sem sentido prático. Atualmente, esta é uma ideia ultrapassada e são poucos os que acreditam nela, onde a grande maioria, por razões estratégicas, operacionais ou simplesmente por razões culturais, reconhecem que uma empresa pode optar por construir e operar a sua própria *Private Cloud* [11].

Neste tipo de implementação, os serviços e os recursos computacionais estão exclusivamente dedicados a uma organização particular e não são partilhados com outras organizações. As *Private Clouds* oferecem muitos dos benefícios presentes nas *Public Clouds*, como a elasticidade e escalabilidade, aliando uma otimização do investimento em *hardware*. Existe uma consolidação e otimização do desempenho do *hardware* físico através da virtualização, o que se traduz numa melhoria da eficiência do *datacenter* e consequente redução dos custos operacionais.

A principal diferença entre uma *Private Cloud* e uma *Public Cloud* reside no facto de, num serviço de uma *Private Cloud*, os dados e processos serem geridos dentro de uma organização. Não existem restrições ao nível da largura de banda da rede, exposições a falhas de segurança ou requisitos legais inerentes à utilização de uma *Public Cloud*.

O principal desafio passa por lidar apropriadamente com a complexidade operacional deste tipo de *Clouds*, uma vez que é um ambiente que está hospedado e é gerido pelos recursos internos.

Por sua vez, a segurança deixou de ser uma preocupação tão relevante, quando em comparação com uma *Public Cloud*. Isto deve-se ao facto da *Private Cloud* estar instalada dentro do *data-center*, estando acessível apenas por interfaces internas, para além de estar interligada por redes privadas e protegida pela Firewall da organização. Uma *Private Cloud* é assunto da organização física, eletrónica e da segurança interna da mesma, garantindo assim um maior grau de segurança sobre os dados sensíveis [9, 10].

Uma referência ainda para as *virtual private clouds*, que segundo [11] são *Private Clouds* controladas e operadas por uma *third party*, neste caso um fornecedor de hospedagem e *outsourcing*.

### 2.5.3 Community Cloud

Neste tipo de *Clouds*, a infraestrutura *Cloud* é partilhada por várias organizações e suporta uma comunidade que partilha alguns interesses (e.g., uma missão comum, requisitos específicos de segurança, políticas entre outras considerações). Os membros da comunidade partilham o acesso aos dados e aplicações da *Cloud* em questão. As *Community Clouds* podem estar localizadas tanto no local como fora do estabelecimento e a sua gestão pode ser levada a cabo por uma organização ou por uma *third party* [9, 10].

Segundo [11], este tipo de *Clouds*, também conhecido como *Vertical Clouds*, são *Public Clouds* organizadas em torno de um grupo competitivo e cooperativo de negócios, sendo dado o exemplo dos serviços financeiros. Nas *Community Clouds*, as organizações podem beneficiar das vantagens de uma *Public Cloud*, tais como a elasticidade e o modelo de cobrança *pay-as-you-go*, mas conjugando também aspetos associados a uma *Private Cloud* como a segurança e privacidade.

### 2.5.4 Hybrid Cloud

Tal como o nome sugere, uma *Hybrid Cloud* é uma combinação de duas ou mais *Clouds* que utilizam diferentes modelos de implementação (*public*, *private* ou *community*). Os intervenientes permanecem entidades únicas, apesar de estarem unidos por tecnologia proprietária e normalizada, permitindo a portabilidade de dados e aplicações. Os benefícios e os desafios deste tipo de implementação acabam por ser uma combinação dos citados nos modelos acima.

Esta abordagem permite a uma organização utilizar a melhor ferramenta para cada trabalho,

embora com um inerente aumento de complexidade. Uma *Private Cloud* pode complementar a sua infraestrutura local com recursos computacionais de uma *Public Cloud*. De outro ponto de vista, os utilizadores deste modelo podem lidar com informação de negócios que não seja crítica numa *Public Cloud*, enquanto os serviços/dados importantes e críticos são mantidos debaixo do seu controlo (*Private Cloud*). Por estas razões, é provável que esta abordagem seja adotada pela maioria das empresas na criação da sua própria *Cloud* [11, 9, 10].

## 2.6 Questões Relacionadas com a Monitorização Cloud

Nesta secção são abordadas algumas questões relevantes relacionadas com a monitorização em *Cloud Computing*, identificando-se características técnicas associadas a este processo. A análise das várias perspetivas descritas de seguida facilita a perceção dos aspetos que caracterizam a monitorização deste tipo de ambientes. Esta análise e o estudo das principais características das plataformas e ferramentas de monitorização (apresentadas no Capítulo 4), sustentam a definição do modelo estratificado proposto para a monitorização de serviços *Cloud* apresentado no capítulo seguinte.

### 2.6.1 Monitorização fim-a-fim

Muitas das entidades que tentam definir o conceito de *Cloud Computing* são unanimes em considerar que não existe mais uma conexão ponto-a-ponto entre os utilizadores e a infraestrutura de computação. Os dados e as aplicações não estão alojados em apenas um computador ou servidor, mas sim num aglomerado de recursos computacionais. Portanto, a monitorização associada a este tipo de tecnologia é caracterizada por ser essencialmente fim-a-fim. Segundo [16], requisitos como a segurança e a privacidade são também responsáveis pela monitorização ser fim-a-fim. Neste caso, os autores abordam a monitorização ao nível dos modelos de serviço IaaS, da qual podem ser retiradas algumas indicações. No caso da monitorização de uma rede envolvendo um modelo IaaS, as interfaces de acesso devem incorporar a abstração associada a uma VLANs. Os dispositivos que constituem a rede em questão estão agrupados em VMs e por sua vez associados a uma ou mais VLAN. Através da engenharia de tráfego são interconectados e acessíveis a administradores de rede para fins de monitorização. Contudo, a monitorização a este nível para os utilizadores tem algumas limitações. A monitorização do tráfego entre dispositivos intermediários (como *routers* ou *switches* virtuais) por parte dos utilizadores é inaplicável,

na medida em que estes dispositivos estão fora do seu âmbito. Um utilizador que pretenda inspecionar as suas operações e configurações, por exemplo com recurso ao uso do *Simple Network Management Protocol* (SNMP), tem de se sujeitar a funcionalidades implementadas e disponibilizadas pelo fornecedor de serviço. Porém, os fornecedores de serviço não estão dispostos a dar aos utilizadores acesso aos dispositivos de rede intermediários, principalmente pelos riscos de segurança que esse processo acarretaria.

Para além das questões de segurança, existem ainda limitações técnicas na monitorização de dispositivos intermediários por parte dos utilizadores. As características técnicas que estão na origem destas limitações são a utilização da agregação de *links* e de tráfego entre os servidores físicos. Para evitar este tipo de inconvenientes, a monitorização da rede deve operar nos pontos terminais das ligações fim-a-fim. Estas são as principais razões pela qual a monitorização deve ser fim-a-fim, onde os pontos terminais são as interfaces de rede virtuais associadas a *hosts* virtuais, que por sua vez estão alocados aos utilizadores finais.

### 2.6.2 Arquitetura centralizada e distribuída

Devido à natureza dos ambientes *Cloud*, a complexidade da sua gestão cresce proporcionalmente ao número de recursos envolvidos. No que toca ao tipo de arquitetura subjacente, a monitorização pode ser centralizada ou distribuída. Na realidade existem várias maneiras de controlo da monitorização, dependendo essencialmente da granularidade dos pedidos de monitorização, da maneira como os recursos são disponibilizados e dos serviços oferecidos pelos fornecedores em si. Ainda relativamente a [16], os autores defendem que a arquitetura de monitorização de toda uma complexa infraestrutura deve ser distribuída, ou seja, não deve estar concentrada em apenas um servidor ou registo. As razões que levam à adoção de uma gestão distribuída prendem-se principalmente com uma resiliência a falhas de rede, aumento da redundância e escalabilidade, assim como uma melhoria da eficiência e da segurança (evitar *bottlenecks* por exemplo). Contudo é necessária uma coordenação maior. Por outro lado, a utilização de uma arquitetura de monitorização centralizada também pode ser vista como viável. Tal como referido anteriormente, este é um aspeto que depende da maneira como os recursos são disponibilizados. Este tipo de arquitetura pode ser levada em consideração numa *Private Cloud*, onde uma empresa possuí o controlo dos recursos e *software*. Um exemplo deste tipo de abordagem é o PCMONS [17], um sistema de monitorização de *Private Clouds*, abordado na Secção 4.1.1. Este sistema leva em consideração a arquitetura de monitorização centralizada no seu desenvolvimento, adequando-se ao modelo cliente/servidor.

### 2.6.3 Metodologias de medição

No que diz respeito às metodologias de medição, as opções recaem em medições de tráfego ativas ou passivas. A medição de tráfego ativa é caracterizada por recorrer a tráfego intrusivo, onde pacotes de prova ou *probes* são injetados na rede com o propósito de medição. Este método tem a vantagem de ser versátil e de estimar QoS com mais facilidade. Contudo é necessário controlar o *overhead* do tráfego de prova para este não interferir no normal funcionamento da rede. Por outro lado, a técnica de medição passiva usa somente tráfego real para realizar a medição. Não existe intrusão de tráfego de prova, mas em contrapartida a aplicabilidade é mais limitada, devido ao grande volume de dados envolvidos [18]. A metodologia de medição passiva recorre frequentemente a dispositivos específicos e a mecanismos embebidos nos dispositivos de rede, situados perto ou dentro dos pontos terminais [16]. Na Secção 4.1.2 é apresentada a Lattice [19], uma *framework* de monitorização desenhada para monitorizar recursos e serviços *Cloud* em ambientes virtuais. A *framework* de monitorização foi concebida de modo a ter uma ”pegada” mínima e que não seja intrusiva. Como consequência, existe a necessidade de assegurar que os componentes de gestão apenas recebem dados relevantes, de modo a que aspetos como o desempenho da rede ou a execução de aplicações não sejam afetados.

Relativamente aos tipos de monitorização, em [20] são considerados dois tipos, nomeados por *pooling* e por subscrição. A monitorização por *pooling* é um processo simples, pois permite ao gestor solicitar o estado atual de uma dada propriedade, sempre que necessário. Por sua vez, a monitorização por subscrição permite ao componente de observação ser notificado sobre mudanças em certas propriedades monitorizadas. A monitorização por subscrição pode ser configurada dos dois seguintes modos: subscrição por mudança, onde é especificado que o componente subscrito é notificado sempre que existirem alterações no valor da propriedade; ou subscrição por intervalo, onde é especificado que o componente subscrito é notificado após um intervalo de tempo configurável.

### 2.6.4 Monitorização dos modelos de implementação

Outro tópico importante passa pela monitorização dos diferentes modelos de implementação de serviços *Cloud*. De facto, cada modelo de implementação possuí as suas próprias características e necessidades, logo necessitam de diferentes abordagens de monitorização. As diferenças existem principalmente entre as *Private Clouds* e as *Public Clouds*. Nas *Private Clouds*, uma empresa apenas tem de lidar com os seus próprios recursos. Devido às suas políticas de segurança,

os dados relevantes estão sob o controle da organização. Por outro lado, as *Public Clouds* exigem um maior investimento na monitorização do tráfego, devido sobretudo à difusão geográfica e aos grandes conjuntos de recursos envolvidos, assim como à necessidade de mais flexibilidade, escalabilidade e segurança. As questões de segurança, tais como configurações de *firewalls*, podem afetar e limitar aspetos relacionados com a monitorização entre os fornecedores de serviços *Cloud*. As *Public Clouds* necessitam ainda de fornecer informações sobre a monitorização aos seus clientes, o que requer mais flexibilidade, segurança e customização.

As características das *Public Clouds* levantam ainda preocupações relacionadas com a distribuição geográfica dos recursos que constituem a base da infraestrutura *Cloud*. À primeira vista, pode parecer que em *Cloud Computing* já não existe a preocupação sobre a localização geográfica. Essa ideia advém da sua natureza permitir um amplo acesso à rede, característica herdada da Internet e na qual se baseia. Contudo, a *Cloud Computing* não pode sobrepor-se às leis da física e os atrasos consequentes numa transmissão de dados podem tornar-se um verdadeiro problema ao fornecimento de um serviço de qualidade. Utilizadores em localizações remotas podem estar sujeitos a latências inaceitáveis, limitando o grau de interatividade e interferindo nos parâmetros relacionados com QoS e QoE. Relativamente à QoE, na Secção 3.2 as questões relativas à distribuição geográfica serão novamente abordadas. No caso dos serviços de armazenamento de dados, existem algumas questões legais que devem ser levadas em consideração na adoção de uma solução *Cloud*, pois é necessário especificar a localização física das infraestruturas do fornecedor (incluindo *third parties* e parceiros) e onde é que os dados vão ser armazenados. A localização geográfica dos dados tem efeitos significativos em aspetos como a sua confidencialidade e privacidade, uma vez que o local físico dos meios utilizados (um ou mais países) está sujeito às leis e regulamentos locais [21]. Neste caso a solução passa por o utilizador tentar adotar um serviço localizado ”relativamente perto” do seu ponto de acesso. Existem serviços que permitem ao utilizador especificar a localização geográfica dos dados (e.g., Amazon). Atualmente, os fornecedores de serviço e de conteúdos procedem à otimização do tráfego das suas redes, apesar de muitas vezes não terem grandes preocupações com o estado da rede ou a localização do cliente. Isto é realizado, por exemplo, com recorso a um mecanismo onde os pedidos são redirecionados para servidores mais próximos do utilizador [22]. A maioria dos fornecedores de serviços de armazenamento específica no SLA que os dados vão residir numa dada localização geográfica. Contudo, os fornecedores podem violar os SLAs intencionalmente ou por acidente, alterando a localização dos dados dos utilizadores, com o intuito de reduzirem custos. Posto isto, a solução passa por uma auditoria por parte dos utilizadores, onde verificam que os dados se encontram no local especificado no SLA. Os SLAs assumem assim um papel

relevante perante questões relacionadas com a distribuição geográfica. Este é aliás um tópico abordado na Secção 3.1.4, no contexto da relação entre cliente e fornecedor. Os utilizadores devem ainda certificarem-se que o fornecedor continua a cumprir as suas obrigações geográficas.

Deste modo, a localização geográfica é uma das chaves para o fornecimento de serviços mais sustentáveis. Qualquer estratégia *Cloud* sofisticada deve levar em consideração a localização física dos recursos, assim como os riscos críticos associados e fornecer mecanismos relativamente transparentes e controláveis para dispor os dados e recursos mais perto de onde sejam necessários.

### 2.6.5 Classificação de métricas

No que diz respeito às métricas envolvidas na monitorização *Cloud*, em [23] é feita uma classificação abstrata, cuja descrição é apresentada de seguida.

Num modelo baseado na aplicação, são descritas duas classes, nomeadamente métricas genéricas (métricas que podem ser medidas para todas as aplicações) e métricas específicas (métricas dependentes de informação adicional para além da fornecida pelas aplicações). Num modelo de medição, a classificação das métricas pode ser efetuada em métricas diretas (métricas que são medidas e utilizadas sem posterior processamento) e métricas calculadas (métricas que são calculadas a partir de duas ou mais métricas). Quanto à sua implementação, o correspondente modelo define as métricas em partilhadas (podem ser implementadas para suportar todas as aplicações) e em individuais (são implementadas para cada aplicação separadamente). No que diz respeito à sua natureza, a classificação é feita em duas classes: quantidade (métricas definidas como um conjunto de recursos a serem fornecidos ao cliente) e qualidade (métricas que representam a qualidade do serviço).

## 2.7 Sumário

Neste capítulo foi apresentado o conceito de *Cloud Computing*, referindo-se brevemente o processo evolutivo que resultou no crescimento exponencial da tecnologia. Foram apresentadas as principais características envolvidas no modelo *Cloud*, bem como os benefícios associados à tecnologia, segundo as perspetivas dos vários intervenientes (fornecedores, parceiros e utilizadores de serviços). Posteriormente foram descritos os vários modelos de serviço, apresentando-se exemplos de serviços mais comuns (agrupados segundo os modelo discutidos, assim como a res-

petiva relação Vendedor/Comprador), e os vários modelos de implementação existentes. Por fim, foram abordadas algumas questões relacionadas com a monitorização *Cloud*, tais como as arquiteturas de monitorização centralizada e distribuída, as metodologias de medição de tráfego ativas e passivas, características próprias de cada modelo de implementação, o impacto da distribuição geográfica, entre outras características técnicas.

# Capítulo 3

# Monitorização Estratificada de Serviços Cloud

Este capítulo centra-se na principal contribuição desta dissertação. Através da análise de um vasto conjunto de referências, abrangendo as diversas áreas, componentes e intervenientes envolvidos num ambiente de *Cloud Computing*, é efetuada uma abordagem estratificada à monitorização dos serviços *Cloud*. Como resultado desta abordagem, é proposto um modelo estratificado que visa abranger as diversas dimensões envolvidas na monitorização deste tipo de serviços. As camadas que constituem o modelo proposto são abordadas individualmente, onde são propostos parâmetros, métricas e boas práticas a levar em consideração na sua monitorização. De seguida são abordadas as questões ao nível da QoE, discutindo-se alguns aspetos relacionadas com qualidade de serviço presenciada pelo utilizador final. Por fim é estabelecida uma relação entre o modelo proposto e os modelos de serviço existentes.

## 3.1 Modelo Estratificado Proposto

O modelo proposto está estratificado em 4 camadas principais, que por sua vez se subdividem em algumas categorias. As 4 camadas principais que constituem o modelo de monitorização correspondem às Infraestruturas, à Rede, ao Serviço/Aplicação e à relação Cliente/Fornecedor, conforme ilustrado na Figura 3.1. A camada referente às Infraestruturas abrange tanto os recursos físicos como os recursos virtuais envolvidos no complexo ambiente de *Cloud Computing*. Para além da necessidade de monitorizar os diversos componentes que constituem toda uma infraes-

trutura, existem ainda outras questões que devem ser monitorizadas a este nível, como questões energéticas e de segurança. Na camada de Rede são abrangidos aspetos relacionados sobretudo com o serviço IP, como o débito e as questões de desempenho, disponibilidade e fiabilidade. Ao nível da camada de Serviço/Aplicação, a monitorização incide em questões que permitem avaliar a disponibilidade, fiabilidade, desempenho e segurança de um serviço, entre outros aspetos. Por fim a relação Cliente/Fornecedor de serviço deve ser alvo de uma monitorização no que diz respeito à auditoria dos SLA, à contabilização do uso/custo de um dado serviço e também às questões de segurança a este nível. Esta subdivisão está disposta na Tabela 3.1, de forma a tornar mais percetível a sua compreensão.

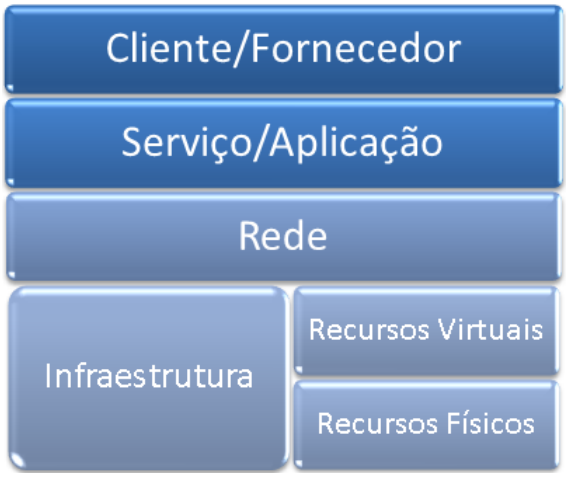



Figura 3.1: Modelo estratificado proposto para a monitorização de serviços *Cloud*.

Antes de se avançar para uma abordagem mais detalhada de cada camada e respetivas categorias, é importante destacar a inclusão de dois tópicos no modelo proposto, respeitantes às questões energéticas e à segurança. Sem retirar importância ou protagonismo às restantes categorias, a inclusão destes dois tópicos revela-se bastante pertinente, na medida em que são dois fortes focos de investigação atual, dada a sua relevância.

Tal como referido, as questões energéticas são cada vez mais um tópicoa ter em consideração na monitorização *Cloud*. As mudanças do clima e o aquecimento global são dois dos problemas mais relevantes para o nosso planeta, onde o aumento das temperaturas está diretamente relacionado com a quantidade de dióxido de carbono produzida. Posto isto, nos vários ramos da ciência e da tecnologia, nos últimos anos tem existido um investimento na investigação e desenvolvimento de soluções "amigas" do ambiente. Neste contexto surgiram termos como *Green*

Tabela 3.1: Modelo estratificado.

| | | | |
|---|---|---|---|
| **Cliente/Fornecedor** | | Auditoria | Métricas CF |
| | | Contabilização | |
| | | Segurança | |
| **Serviço/Aplicação** | | Disponibilidade/ Fiabilidade | Métricas SA |
| | | Desempenho | |
| | | Segurança | |
| | | Outras | |
| **Rede** | | Débito | Métricas RD |
| | | Desempenho | |
| | | Disponibilidade/ Fiabilidade | |
| **Infraestrutura** | **Recursos Virtuais** | Componentes | Métricas RV |
| | | Segurança | |
| | **Recursos Físicos** | Componentes | Métricas RF |
| | | Energia | |
| | | Segurança | |

*IT* e *Green Cloud Computing*. O desafio em *Green Cloud Computing* passa por minimizar a utilização de recursos e continuar a satisfazer a qualidade dos serviços requisitados e a sua robustez, contribuindo não só para uma redução dos custos operacionais como também do impacto ambiental [24, 25]. Posto isto, as questões energéticas são sobretudo um aspeto a ser associado à monitorização das infraestruturas ao nível dos recursos físicos.

Outra das grandes preocupações inerentes à monitorização prende-se com as questões de segurança. Este é um tópico em que tem havido grande investigação por ser crucial para a implantação do paradigma pois relaciona-se com a confiança e adesão dos clientes. Dada a elevada importância que as questões de segurança acarretam, relacionadas nomeadamente com integridade, disponibilidade, privacidade e autenticidade dos dados e dos utilizadores, torna-se pertinente incluir uma categoria "Segurança" em algumas das camadas definidas. Para este tópico serve de base o trabalho proposto em [26, 27], onde são recomendadas algumas restrições e auditorias à segurança *Cloud*. Os autores baseiam-se no trabalho efetuado pela *Cloud Security Alliance* (CSA), onde a *Cloud* é modelada em sete camadas, nomeadamente: *Facility, Network, Hardware, OS, Middleware, Application e User* (ver Figura 3.2). Tendo em conta a abordagem estratificada à monitorização dos serviços *Cloud* proposta, é necessário agrupar algumas destas camadas. Como é possível observar na Tabela 3.1, as questões relacionadas com a segurança

foram agrupadas e associadas às camadas dos "Recursos Físicos" e "Recursos Virtuais" nas Infraestruturas, assim como nas camadas de "Serviço/Aplicação" e "Cliente/Fornecedor".

De seguida serão apresentados em maior detalhe alguns dos parâmetros, métricas[1] e boas práticas a ter em consideração em cada camada de monitorização *Cloud* definida.


Figura 3.2: Modelo estratificado para a Segurança *Cloud*, segundo a CSA.

### 3.1.1 Infraestrutura

Como base e suporte de toda uma arquitetura complexa que envolve o ambiente *Cloud Computing*, as infraestruturas físicas são um dos focos principais a ter em conta no processo de monitorização. Neste contexto são consideradas as categorias referentes aos Componentes, Energia e Segurança (ver Tabela 3.2).

Todos os componentes físicos, desde dispositivos de processamento, armazenamento, até aos de rede (*switches*, *routers*) devem ser monitorizados. A maioria das referências na área é unânime em considerar como métricas mais relevantes a percentagem de utilização de CPU, RAM, memória de armazenamento, assim como as estatísticas das interfaces de rede das máquinas físicas [19, 24, 25, 28, 29]. Na Tabela 3.2 são ainda apresentados outros tipos de métricas relevantes neste contexto. Como já referido, os dispositivos de rede devem ser igualmente monitorizados, pois problemas ao nível dos *switches*, *routers* ou mesmo dos *links* podem afetar a conetividade

[1] Ao longo deste capítulo o conceito de métrica é por vezes usado de uma forma relaxada, para indicar parâmetros em avaliação. De uma forma mais rigorosa, para cada parâmetro podem ser definidas várias métricas específicas, como é o caso das métricas do *IP Performance Metrics* (IPPM) e ITU-T para o desempenho da rede.

da topologia. Uma topologia instável pode acarretar problemas que influenciem todo um conjunto de aspetos, como, por exemplo, a engenharia de tráfego, o débito, a disponibilidade de um serviço, violações dos SLA, questões económicas, entre outros.

No que toca às questões energéticas, uma parte significativa da energia elétrica consumida pelos recursos de computação é transformada em calor, o que por sua vez acarreta alguns problemas. As altas temperaturas reduzem o tempo de vida dos dispositivos/componentes e acabam também por influenciar a fiabilidade e disponibilidade do sistema. Por sua vez, os procedimentos de gestão de energia podem afetar o desempenho do sistema de uma maneira complexa, dado que a taxa de computação global é resultado da velocidade e da coordenação de múltiplos elementos dentro de um sistema [30]. Assim, tendo em atenção as questões energéticas na monitorização *Cloud*, sobretudo ao nível das infraestruturas, na Tabela 3.2 estão indicadas algumas das métricas a considerar, retiradas de referências na área que podem ser associadas a este tópico. Segundo [30], onde são levados em consideração aspetos ecológicos e de desempenho no sistema de gestão de recursos (métricas, técnicas, modelos, políticas, algoritmos), o consumo de energia é uma boa métrica para abordar as questões relacionadas com a energia. Também em [25], o consumo de energia é uma métrica levada em consideração. Os autores apresentam uma *framework* arquitetural e princípios para uma eficiência energética na gestão em ambientes *Cloud Computing*. O algoritmo proposto é caracterizado pelas preocupações energéticas na alocação de recursos e utiliza uma consolidação dinâmica das VMs. Comparativamente com as técnicas estáticas de alocação de recursos, através dos resultados obtidos na avaliação da *framework*, os autores constataram que houve uma redução substancial nos consumos de energia. Por sua vez, em [24] são propostas também métricas ao nível do controlo das temperaturas e sistemas de *backup* de energia (geradores e UPS - *Uninterruptible Power Supply*). Estas são métricas que surgem no contexto da solução proposta pelos autores para a gestão de recursos, baseada em modelos organizados e compostos por agentes autónomos. O objetivo passa sobretudo por otimizar a utilização de energia e reduzir a emissão de dióxido de carbono.

Relativamente à segurança das infraestruturas ao nível dos recursos físicos existem algumas restrições e auditorias que devem ser levadas em consideração. Visto que nesta etapa são abordados os recursos físicos *Cloud*, podem ser associadas as três primeiras camadas propostas pela CSA e abordadas em [26]. Portanto, tendo em conta a análise feita às camadas *Facility, Network* e *Hardware* podem ser extraídas métricas de alguns dos procedimentos propostos, conforme os exemplos apresentados na Tabela 3.2. No que toca às instalações (*Facility*) a segurança é sobretudo ao nível físico, onde podem ser implementados controlos de acesso através de videovigilância, vários sistemas de autenticação, sistemas de alarmes e sensores, entre outros. Os

objetivos principais passam por evitar infiltrações maliciosas, manipulações de dados e assegurar a própria integridade das instalações e componentes. Ao nível do *Hardware* propriamente dito, as medidas de segurança estão em conformidade com as adotadas nas instalações, onde devem ser seguidos os protocolos de segurança. No que diz respeito à camada *Network*, devido à sua natureza, que pode ser descrita como a "fronteira" entre os dados dos clientes e os próprios clientes (inseridos por vezes em redes sujeitas a ameaças), podem ser adotadas *Firewalls*, *Intrusion Detection Systems* (IDS), *Intrusion Prevention Systems* (IPS), entre outros mecanismos.

Tabela 3.2: Exemplos de Métricas RF.

| **Layer** | **Categoria** | **Exemplo de Métricas RF** |
|---|---|---|
| Infraestrutura<br><br>Recursos Físicos | Componentes | Estatísticas do CPU (percentagem de utilização global, por CPU ou por tarefa, nº de cores), RAM (percentagem de utilização), memória de armazenamento (percentagem de utilização, velocidade de leitura/escrita), estatísticas da memória virtual, *paging* e falhas, interrupções (globais, por CPU, por *hardware* e *software*), estatísticas de *swapping*, das *socket* e da utilização das tabelas internas do *kernel*, estatísticas das interfaces de rede, conectividade da topologia. |
| | Energia | Consumos de energia, temperaturas, estados dos geradores e UPS, frequência do *clock* do CPU (instatânea e média). |
| | Segurança | Alarmes/sensores de incêndio, vigilância, controlo de acessos, sistemas de autenticação, monitorização de firewalls, IDS, IPS. |

Ainda a nível das infraestruturas, os recursos virtuais assumem um papel muito interventivo num ambiente de *Cloud Computing*, pelo que a sua monitorização se torna um aspeto essencial. As categorias abordadas estão relaciondas com os Componentes e com a Segurança (ver Tabela 3.3).

Os processos envolvidos na virtualização são constituídos por algumas operações importantes, como a suspensão/reinício/migração e início/paragem de VMs. Estas são operações abordadas em muitos dos recentes trabalhos de investigação, tendo em vista o desenvolvimento de métricas, como por exemplo a "utilização" [30]. Várias referências apontam os diversos componentes dos recursos virtuais como aspetos a monitorizar. As métricas mais comuns a este nível estão relacionadas sobretudo com a percentagem de utilização do CPU, RAM e memória de armazenamento das VMs (ver Tabela 3.3), entre outras apresentadas na Tabela 3.3. As estatísticas das interfaces de rede das VMs são igualmente relevantes. Operações relacionadas com os pro-

cessos de criação e migração de VMs ou número de instâncias ativas são também informações úteis [19, 24, 25, 28, 29].

Quanto à segurança nos recursos virtuais das infraestruturas, podem ser associadas as camadas *OS* e *Middleware* abordadas em [26, 27]. Neste caso as métricas a serem levadas em consideração devem ser extraídas da monitorização dos Sistemas Operativos ao nível dos eventos assim como do sistema de chamadas entre as VMs e o *Hardware*. O objetivo passa sobretudo por evitar a cópia e modificações de dados. A camada *Middleware*, segundo os autores em [27], é considerada um potencial ponto fraco, pois encontra-se entre as camadas do OS e das Aplicações, envolvendo assim bastantes componentes, conforme o serviço e respetiva arquitetura em questão. Ao ser um tópico bastante vasto e dado que as métricas que poderiam ser obtidas não estão ainda bem definidas, não foram retiradas grandes conclusões acerca desta camada. Contudo, as métricas que podem surgir devem estar associadas à monitorização da virtualização e dos sistemas de segurança em arquiteturas *Cloud* heterogéneas. Ainda relativamente ao *Middleware*, e dado que esta camada foi agrupada nos recursos virtuais, em [31] propõem um IDS que opere a este nível. Os autores defendem que devido à natureza distribuída da *Cloud Computing*, cada nó deve ser monitorizado e, quando um ataque ocorrer, os restantes nós devem ser alertados. Este tipo de comunicação requer compatibilidade entre os vários componentes, vários mecanismos de comunicação e permissões de controlo sobre sistemas de manutenção e atualizações. Estas características são fornecidas geralmente pelo *middleware Cloud*. Posto isto, os autores propõem um IDS que integra análises de conhecimentos e de comportamentos para detetar intrusões específicas, onde o sistema de auditoria foi projetado para cobrir ataques que não são detetados ao nível da rede e das infraestruturas físicas.

Tabela 3.3: Exemplos de Métricas RV.

| **Layer** | **Categoria** | **Exemplo de Métricas RV** |
|---|---|---|
| Infraestrutura<br><br>Recursos Virtuais | Componentes | Estatísticas do CPU (percentagem de utilização global, por CPU ou por tarefa, nº de cores), RAM (percentagem de utilização), memória de armazenamento (percentagem de utilização, velocidade de leitura/escrita), estatísticas da memória virtual, *paging* e falhas, interrupções (globais, por CPU, por *hardware* e *software*), estatísticas de *swapping*, das *socket* e da utilização das tabelas internas do *kernel*, estatísticas das interfaces da VM, migrações de VM, número de instâncias ativas. |
| | Segurança | Monitorização de eventos e do OS ao nível do sistema de chamadas entre VMs e o *hardware*, IDS. |

### 3.1.2 Rede

Passando para a camada de "Rede", a classificação das categorias que a caracterizam e as respetivas métricas são sobretudo ao nível do Serviço IP. A subdivisão nesta camada é efetuada nas seguintes categorias: Débito, Desempenho e Disponibilidade/Fiabilidade (ver Tabela 3.4). As métricas RD propostas para estas categorias estão associadas às métricas tradicionais das redes de computadores e telecomunicações, provenientes sobretudo dos esforços dedicados a este tópico pelo ITU-T e pelo grupo de trabalho IPPM do *Internet Engineering Task Force* (IETF).

No que diz respeito ao Débito são várias as referências que o classificam como essencial no processo de monitorização *Cloud* [32, 33]. Para além da importância ao nível das decisões relacionadas com a engenharia de rede, as questões económicas também estão "atentas" a este aspeto. Devido à sua natureza, o débito pode variar constantemente, o que leva a que as métricas que lhe estão relacionadas sejam monitorizadas de perto e tendo em vista o cumprimento dos SLA [34]. Na análise do volume de tráfego por unidade de tempo, uma monitorização ao nível de classes de serviço pode trazer benefícios, nomeadamente para a otimização da utilização da rede, identificação de classes com problemas, etc. A largura de banda quantifica o volume de dados que um *link* ou caminho pode transferir por unidade de tempo. A largura de banda disponível, representa assim uma métrica variável no tempo, onde é identificada a capacidade disponível, levando em consideração a carga atual. A capacidade, representando o limite máximo à largura de banda disponível, é também uma métrica que se enquadra neste contexto.

Em [19, 28] as estatísticas referentes ao tráfego de rede são também apontadas como fontes de dados importantes à monitorização. Esta informação pode ser útil também na camada de rede ao nível do serviço IP, para além da camada das Infraestruturas físicas e virtuais, tal como referido anteriormente.

Quanto às métricas referentes ao Desempenho ao nível da rede, estas englobam as tradicionais métricas de QoS como a duplicação de pacotes, perda de pacotes (OWPL - *One-way packet loss*, OWLP - *One-way loss pattern*, IPLR - *IP packet loss ratio*), atraso (OWD - *one-way delay*, RTT - *round-trip time*, IPTD - *IP packet transfer delay*), *IP packet error ratio* (IPER), *Spurious IP packet ratio* (SPR), *jitter*, entre outras [29, 35].

No que toca à Disponibilidade/Fiabilidade de uma rede, esta pode apresentar períodos de inatividade provocados por problemas que podem ter origem nos componentes de rede, configurações de routing, entre outros aspetos. Face a esta possibilidade, torna-se relevante monitorizar a (in)disponibilidade de uma rede, assim como o estado da conectividade. O tempo de resposta a

uma configuração de rede também pode ser um indicador relevante para avaliar a disponibilidade da rede. Perante a ocorrência de falhas na rede, o tempo médio entre a ocorrência de falhas ou o tempo médio de restauro são bons fatores de avaliação da fiabilidade de uma rede.

Uma referência ainda para o estudo efetuado em [36], do qual podem ser retiradas algumas conclusões sobre aspetos associados à camada de rede. Relativamente ao débito, a sua variação tem como principais fontes a tecnologia de acesso, as políticas do *traffic shaping* dos ISPs e a congestão durante as horas de ponta. Medições efetuadas num único instante de tempo, se forem efetuadas num intervalo de tempo inadequado, podem levar à obtenção de números enganosos e que não representam o desempenho em períodos de tempo extensos. Por outro lado, a latência é afetada principalmente por aspetos como a qualidade do *link* de acesso, *modem buffering* e o tráfego cruzado dentro de casa. A infraestrutura de rede na casa dos utilizadores pode afetar o desempenho, o que torna recomendável proceder a um bom planeamento da mesma.

Ainda referente ao mesmo estudo, os autores concluem que devido à adoção de diferentes políticas e comportamentos dos ISPs, é difícil comparar as medições efetuadas através dos mesmos. É defendida ainda a ideia de que não existe um ISP ”melhor” para todos os utilizadores. Os utilizadores devem escolher um ISP de acordo com os seus interesses pessoais e os perfis de utilização oferecidos.

Tabela 3.4: Exemplos de Métricas RD.

| **Layer** | **Categoria** | **Exemplo de Métricas** |
|---|---|---|
| Rede | Débito | Volume de tráfego por unidade de tempo, largura de banda utilizada e disponível, capacidade. |
| | Desempenho | Duplicação de pacotes, perda de pacotes (OWPL, OWLP, IPLR), atraso (OWD, RTT, IPTD), IPER, SPR, *jitter*. |
| | Disponibilidade/ Fiabilidade | *UP time*, (in)disponibilidade da rede, conetividade (*one* ou *two-way*), tempo de resposta (médio e máximo), tempo médio de restauro em caso de falhas, tempo médio entre falhas. |

### 3.1.3 Serviço/Aplicação

Na camada de Serviço/Aplicação, a natureza dos parâmetros monitorizados e a maneira como estes devem ser recolhidos depende essencialmente do *software* a ser monitorizado e não da infraestrutura *Cloud* em que está inserido. As categorias consideradas neste contexto referem-se

à Disponibilidade/Fiabilidade, Desempenho, Segurança, entre outras mais específicas conforme o tipo de serviço (ver Tabela 3.5).

Uma das principais preocupações a ter em conta passa pela disponibilidade de um Serviço/Aplicação. Um Serviço/Aplicação *Cloud* está sujeito a um conjunto de aspetos de diversas naturezas que podem afetar a sua disponibilidade, devido sobretudo à complexidade do ambiente *Cloud*. Posto isto, devem ser associadas métricas à disponibilidade de um Serviço/Aplicação, onde são registados os períodos de tempo em que um serviço está em funcionamento e quando se encontra indisponível. Este é um tópico que envolve questões económicas, pois em caso de indisponibilidade de um Serviço/Aplicação existem violações de SLA e posteriores penalizações do lado do fornecedor, uma vez que a qualidade do serviço foi afetada. Nas méticas SA da Tabela 3.5 estão ainda indicadas algumas métricas associadas à Disponibilidade e Fiabilidade de um Serviço/Aplicação. Quando estamos perante um cenário de falhas de serviço (indisponibilidade de um serviço ou quebra significativa da qualidade de serviço), a capacidade de recuperação e o tempo utilizado deve ser do conhecimento dos clientes ou de *third-parties* responsáveis pela monitorização. Para além dos aspetos relacionados com a recuperação de um dado Serviço/Aplicação, os intervalos de tempo entre a ocorrência de falhas também funcionam como indicadores da sua fiabilidade e eficiência.

Por sua vez, o tempo de resposta de um dado serviço pode funcionar como um fator de medição do seu desempenho. Neste contexto, em [34] são abordadas métricas referentes ao tempo de resposta médio e máximo num cenário de jogos *online* em *Cloud Computing*. Em [9] são ainda referidas métricas associadas ao desempenho de um serviço como o tempo de processamento *batch*, para além do tempo de resposta *online*.

Devido à natureza insegura do ambiente onde alguns dos Serviços/Aplicações são disponibilizados, a segurança torna-se um aspeto relevante a controlar. Tal como indicado nas métricas SA da Tabela 3.5, o número de vulnerabilidades de segurança deve ser uma métrica importante, uma vez que é necessário monitorizar comportamentos para detetar possíveis violações. Outros aspetos ao nível da camada da Aplicação que podem ser monitorizados e salvaguardados são sobretudo os certificados digitais, chaves privadas, *Domain Name System Security Extensions* (DNSSEC), etc. O comportamento do utilizador também pode ser associado a esta camada e as métricas relevantes prendem-se sobretudo com os processos de login, padrões de acesso, IPs associados, entre outras. A monitorização deve incidir ainda na gestão de passwords, onde são fornecidos dados como o formato das passwords e frequência com que devem ser renovadas [32].

Para além das métricas e aspetos da monitorização referidos anteriormente, podem ainda

ser acrescentadas métricas associadas especificamente ao tipo de Serviço/Aplicação em questão. Dependendo do tipo de serviço em questão, este tópico pode ser uma grande e importante fonte de dados. Por outro lado, pode ser útil o registo de um histórico, onde podem constar os IPs de acesso e registos dos tempos de login referentes aos diversos clientes.

Tabela 3.5: Exemplos de Métricas SA.

| Layer | Categoria | Exemplo de Métricas SA |
|---|---|---|
| Serviço/Aplicação | Disponibilidade/ Fiabilidade | *UP time*, (in)disponibilidade do serviço, Tempo de restauro em caso de falhas, tempo de tempo médio entre falhas. |
| | Desempenho | Tempo de resposta (médio/máximo), processamento *batch*. |
| | Segurança | Número de vulnerabilidades de segurança, padrões de acesso, processos de login, gestão de passwords. |
| | Outras | Registos dos tempos de login e IPs de acesso (histórico), métricas específicas do tipo de aplicação. |

### 3.1.4 Cliente/Fornecedor

Para esta camada vão ser abordadas categorias que dizem respeito à Auditoria, Contabilização e Segurança (ver Tabela 3.6).

A relação Cliente/Fornecedor envolve todo um conjunto de interesses comerciais, o que torna necessário o estabelecimento de um contrato onde sejam especificados todos os aspetos do serviço em questão. Neste contexto é importante esclarecer o conceito de SLA. Um SLA é um contrato estabelecido entre fornecedor e cliente e especifica quais as necessidades dos consumidores e o compromisso dos fornecedores para com eles. Num SLA estão contidos normalmente itens como: conjunto de serviços fornecidos, uma definição completa e específica de cada serviço, requisitos de QoS, tempo de atividade, segurança, privacidade, procedimentos de backup, responsabilidades de ambas as partes, entre outros [10]. Uma referência ainda para as questões relacionadas com a localização geográfica dos *datacenters* em relação às leis nacionais e internacionais. Este é tido como um critério importante pelas companhias que pretendem investir em soluções baseadas na *Cloud*. Para tal é necessário que os SLAs incluam e abranjam este tipo de parâmetros (abordados em maior detalhe na Secção 2.6). Face à sua importância, devem ser

efetuadas auditorias de modo a comprovar que os dados se encontram no local especificado no SLA. O estabelecimento de normas para *Cloud Computing*, que ainda não se encontram claramente definidas, podem ajudar a lidar com estes parâmetros geográficos e legais [37]. No que toca à gestão de serviços, o cliente deverá requerer sumários de todo um conjunto de auditorias, feitas pelo fornecedor do serviço, como parte da verificação do respetivo SLA. Os SLA funcionam assim como um dos instrumentos primários de controlo do utilizador. Portanto, uma das métricas vitais a este nível de monitorização passa pela auditoria dos SLA, onde são registadas todas as violações e incumprimentos dos mesmos. Posto isto, a verificação do cumprimento dos SLAs está diretamente relacionada com as camadas anteriores, uma vez que pode ser necessário recorrer a métricas estabelecidas ao nível das infraestruturas, rede ou serviços. Por exemplo, em [25] é referida uma métrica relativa à média de violações de SLA, que representa a média de desempenho de CPU que não foi alocada a uma aplicação quando requerida. Em caso de ocorrência deste tipo de incumprimentos, existem consequências. Dependendo dos parâmetros dos contractos estabelecidos entre clientes e fornecedores, podem ocorrer penalizações e compensações por parte dos fornecedores de serviço.

A monitorização da contabilização do uso do serviço também é um aspeto bastante importante, na medida em que existe a necessidade de assegurar os interesses económicos de ambas as partes. Devido à natureza elástica dos ambientes *Cloud*, aliado ao modelo comercial *"pay-as-you-go"*, a medição da utilização e o custo tornam-se aspetos vitais [29, 34]. A análise da contabilização dos serviços e respetiva receita, permite também aos fornecedores de serviço adaptarem os seus planos de preçários e estratégias comerciais conforme as necessidades do mercado. Este estudo pode fazer a diferença, numa altura onde a forte concorrência na área se notabiliza, fruto do aumento da oferta na *web* de ferramentas baseadas em *Cloud Computing*.

Relativamente à segurança, em [9] são abordados alguns parâmetros que devem estar incluídos num SLA, e que por sua vez se enquadram na relação entre cliente e fornecedor. Este tipo de parâmetros diz respeito sobretudo ao estado de aquisição e atualização dos padrões de segurança relevantes por parte do fornecedor, assim como dos certificados. Outros tipos de parâmetros indicados são o estado de certificação da parte responsável pela gestão; estado das restrições operacionais incluídas nas medidas de segurança impostas pelo sistema de gestão; estado da garantia de confidencialidade nas trocas de dados entre *Clouds*; localização dos dados; estado da aquisição de *logs* para a deteção de atos maliciosos e o período durante o qual estes são mantidos; estado do controlo da comunicação para bloquear comunicações maliciosas; estado das medidas que atuam contra o congestionamento da rede, evitando ataques *Denial of Service* (DoS)/*Distributed Denial of Service* (DDoS); estado das medidas contra *malware*.

Tabela 3.6: Exemplos de Métricas CF.

| Layer | Categoria | Exemplo de Métricas CF |
|---|---|---|
| Cliente/Fornecedor | Auditoria | Monitorização de violações/incumprimentos dos SLA, penalizações. |
| | Contabilização | Monitorização do uso e respetivo custo do serviço, receita. |
| | Segurança | Estado de aquisição e atualização dos padrões de segurança, certificados, localização dos dados. |

## 3.2 Questões Relacionadas com QoE

Em *Cloud Computing*, na perspetiva do serviço *Cloud* para o fornecedor e cliente, a gestão dos serviços e a garantia de QoS tornou-se uma das dificuldades do desenvolvimento da tecnologia. Com a migração de aplicações pessoais e comerciais para a *Cloud*, a qualidade do serviço prestado torna-se um importante diferenciador entre os diversos fornecedores. Um fator que está diretamente relacionado com a QoS é a qualidade que é presenciada pelo utilizador final, ou seja, a qualidade de experiência (QoE) resultante da utilização de um dado serviço. Posto isto, torna-se também imprescindível monitorizar a QoE. Neste tipo de monitorização são tidas em conta métricas como atraso, variações do atraso (*jitter*), perdas, latência, entre outras. Contudo, estes são aspetos que não fazem parte do conhecimento e do vocabulário comum dos utilizadores finais. Porém a sua opinião e *feedback* acerca da satisfação em relação aos serviços subscritos são um fator bastante relevante a ter em conta na avaliação de toda a infraestrutura. Devido aos diversos intervenientes de um ambiente *Cloud*, perceber e gerir a QoE dos serviços requer uma visão multidisciplinar, que integra a tecnologia, utilizador e aspetos comerciais da qualidade do acesso do utilizador final. O objetivo principal da gestão da QoE está assim relacionado com a intenção de fornecer uma aplicação *Cloud* de alta qualidade ao utilizador final e tentar minimizar os custos dos diversos intervenientes. Estes vão desde entidades relacionadas com os modelos de serviço da pilha *Cloud* (IaaS, PaaS e SaaS), até aos fornecedores das redes subjacentes (TELCOS - *Telecommunications companies e ISPs*).

No que toca aos atuais trabalhos de investigação da QoE na *Cloud*, estes focam-se sobretudo em aplicações multimédia, onde se encaixam serviços de *streaming* HTTP como o Youtube ou Netflix. O impacto dos tempos de espera na perceção do utilizador tem ganho especial atenção nas comunidades de investigação, dado o aumento de popularidade dos serviços multimédia *Cloud* [22]. A relevância dos tempos de espera pode ser também associada a aplicações inte-

rativas como *web browsing*. Quanto aos serviços *Cloud* mais complexos, como produtos *office*, edição colaborativa ou OS a correr na *Cloud*, os trabalhos de pesquisa relacionados com a QoE ainda estão a dar os primeiros passos. Existem ainda algumas questões em aberto como, por exemplo, o impacto da interatividade dos utilizadores e a sua influência na QoE ou o relacionamento da QoE com as expectativas dos utilizadores, resultantes do domínio do uso do serviço em questão, entre outras.

Relativamente à gestão da QoE em geral, em [22] são abordados os passos básicos a ter em consideração. Estes estão relacionados com o entendimento e mapeamento, monitorização e estimação, adaptação e controlo da QoE. Num primeiro passo, é necessário entender quais são os requisitos de uma aplicação e efetuar um mapeamento entre parâmetros mensuráveis e QoE. Um mecanismo típico de avaliação de QoE passa pelo cálculo de *Mean Opinion Scores* (MOS). O próximo passo consiste na monitorização (desde infraestruturas, condições da rede, SLAs e informações específicas das aplicações) e estimativa de QoE. A monitorização pode ser efetuada pelo fornecedor dentro da rede, onde são requeridas funções de mapeamento entre a QoS e QoE, ou ao nível de parâmetros específicos de uma aplicação, o que requer técnicas de *Deep Packet Inspection* (DPI). Como alternativa, existe ainda a opção da monitorização no utilizador final, dando a melhor perspetiva sobre a qualidade presenciada. Por fim, a adaptação e o controlo da QoE têm como objetivo possibilitar aos fornecedores atuarem antes que o utilizador possa notar algum problema e ficar insatisfeito ou abandonar o serviço.

Os mesmos autores identificam ainda alguns desafios que surgiram com a migração de serviços para a *Cloud* e que têm influência na qualidade presenciada pelos utilizadores finais. Os desafios identificados podem ir desde a distribuição geográfica do utilizador (aspeto abordado também na Secção 2.6), artefactos introduzidos com o aumento das distâncias da rede entre o utilizador e o serviço, problemas de gestão de recursos derivados das localizações geográficas, ou até a questão do envolvimento de diversas entidades no fornecimento de um serviço. No caso da localização geográfica, esta pode limitar o grau de interatividade, uma vez que utilizadores em localizações remotas podem estar sujeitos a latências inaceitáveis, levando em consideração a distância entre o *datacenter* e o local onde o serviço é acedido. A grande quantidade de utilizadores em várias localizações geográficas também podem ter influência direta num serviço, pois podem ser afetados requisitos como a escalabilidade e a velocidade de acesso. Uma referência ainda para a dependência da QoE para com as condições da rede e os SLAs, na medida em que é definido o caminho entre o *datacenter* e o utilizador final, atravessando diferentes domínios administrativos.

## 3.3 Relação com os Modelos de Serviço

Devido às diferenças significativas entre os três modelos de serviço mais populares, é consensual que não exista uma solução genérica de monitorização *Cloud*. Cada modelo de serviço possui diferentes áreas e graus de controlo, assim como as suas próprias características de gestão. Posto isto é compreensível que seja difícil alcançar uma solução de gestão *Cloud* genérica. Face a este paradigma, um sistema de monitorização necessita de ser planeado e desenvolvido, com o intuito de ser adequado aos objetivos da gestão. Este processo para além de cobrir os vários constituintes de todo um ambiente *Cloud*, deve ainda levar em consideração aspetos como a QoS e a QoE, os SLAs e características como a segurança, robustez, escalabilidade, elasticidade, entre outras [19].

Neste contexto torna-se útil relacionar o modelo estratificado proposto para a monitorização de serviços *Cloud* com os modelos de serviço. Essa relação está ilustrada na Figura 3.3, levando em consideração os três modelos de serviço mais populares (IaaS, PaaS e SaaS) e as camadas do modelo proposto (Infraestrutura, Rede, Serviço/Aplicação e Cliente/Fornecedor).

No que diz respeito à camada de monitorização das Infraestruturas, esta pode ser essencialmente associada ao modelo de serviço IaaS. Na base desta associação estão as características dos componentes envolvidos, uma vez que são comuns. Na camada das infraestruturas estão incluídos componentes dos recursos físicos e virtuais relativos ao processamento, armazenamento e comunicação em rede, ou seja, aspetos que também caracterizam o modelo de serviço IaaS. Por outro lado, as plataformas de desenvolvimento de serviços também podem estar interessadas em aspetos relativos à monitorização das infraestruturas. Um exemplo desta situação situa-se no processo de desenvolvimento de serviços, onde por vezes pode ser necessário integrar um sistema de monitorização de infraestruturas. Posto isto, o modelo de serviço PaaS também acaba por estar relacionado com a camada das infraestruturas. Do ponto de vista dos serviços enquadrados num modelo de serviço SaaS, a importância dos aspetos relacionados com a monitorização das infraestruturas dependem sobretudo do tipo de serviço. Esta questão é um pouco relativa e discutível, uma vez que podem haver serviços em que não haja o mínimo de interesse em conhecer-se o estado das infraestruturas que o suportam, assim como outros podem requerer este tipo de informação. Face a esta discussão, na Figura 3.3, a camada das Infraestruturas não contempla por completo o modelo de serviço SaaS.

Quanto à camada de Rede, se for levada em consideração uma monitorização fim-a-fim, esta relaciona-se com os três modelos de serviço direta ou indiretamente. Tendo em consideração

todo o ambiente *Cloud*, dependendo dos intervenientes e do serviço em questão, a monitorização da Rede pode ser efetuada de várias maneiras. No caso de um serviço fornecido a um cliente com base num modelo SaaS, as questões relacionadas com os aspectos de Rede (como Débito e Desempenho), devem ser monitorizadas desde a origem (infraestruturas, *datacenters*) até ao local onde o utilizador acede ao serviço. Neste caso estão envolvidos os modelos de serviço IaaS e SaaS. No caso de existirem intermediários, como as companhias de desenvolvimento de *software*, ou seja, fornecedores de plataformas de serviço inseridos num modelo de serviço PaaS, a monitorização da camada de Rede também está relacionada com este modelo. Portanto esta camada pode estar relacionada com os diversos intervenientes, uma vez que pode atravessar os vários modelos de serviço, tendo em consideração uma monitorização fim-a-fim.

Por sua vez, a camada Serviço/Aplicação está associada ao modelo de serviço SaaS. Na base desta relação está a natureza da camada, uma vez que os parâmetros monitorizados e a maneira como devem ser recolhidos dependem sobretudo do *software* e não da infraestrutura.

Por fim, a camada Cliente/Fornecedor ao abordar aspetos como contabilização e auditorias, está diretamente relacionada com os três modelos de serviço. Devido à existência de diversos intervenientes no complexo ambiente *Cloud* é normal existirem relações comerciais entre fornecedores e clientes a vários níveis. Os fornecedores tanto podem disponibilizar *datacenters*, como plataformas de serviço ou *software*, enquanto os clientes podem ser empresas de desenvolvimento de plataformas e *software* ou os utilizadores finais.

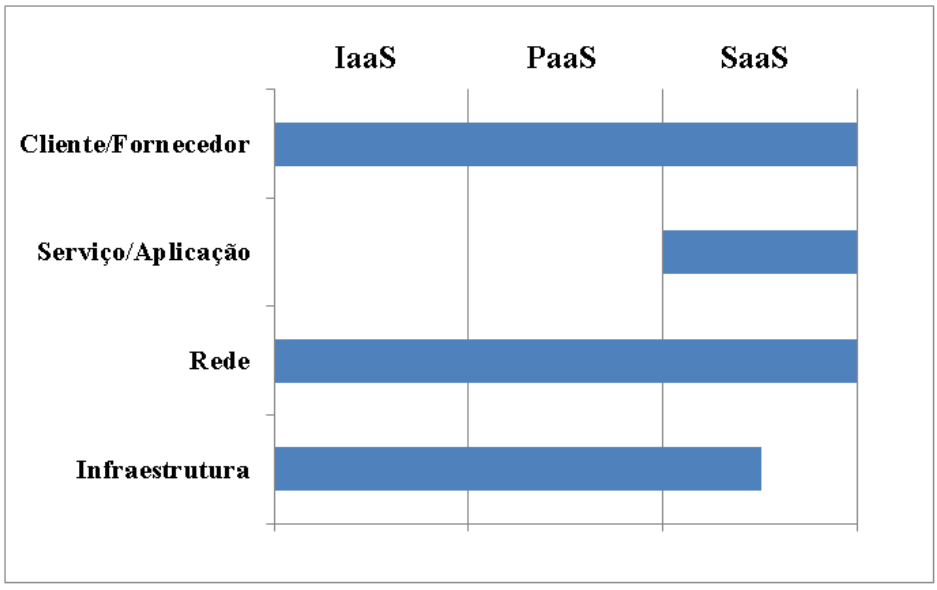



Figura 3.3: Relação com os modelos de serviço.

## 3.4 Sumário

Este capítulo centrou-se na proposta estratificada de monitorização de serviços *Cloud*, com base no estudo da literatura relacionada. Inicialmente foi apresentada a abordagem estratificada à monitorização deste tipo de serviços, onde foi proposto um modelo que tem como objetivo abranger as várias dimensões envolvidas. De seguida foram abordadas em maior detalhe as camadas que constituem o modelo (Infraestrutura, Rede, Serviço/Aplicação e Cliente/Fornecedor), onde são propostos parâmetros e métricas para a monitorização, bem como boas práticas a levar em consideração. Este processo enquadra-se num dos objetivos deste trabalho, onde a intenção é identificar métricas relevantes para a monitorização das várias dimensões. Estas são baseadas sobretudo na análise de diversas referências onde são apresentadas *frameworks*, metodologias, abordagens e conceitos ao nível da monitorização *Cloud* nas várias dimensões. Após uma análise da bibliografia relevante na área, é possível constatar que ainda não existe uma aceitação comum e adequada na classificação de métricas que satisfaçam todos os requisitos impostos pelos ambientes *Cloud*. Portanto, uma apropriada classificação das métricas e a sua normalização são um grande desafio, tendo em vista uma gestão eficiente e a otimização dos ambientes *Cloud*.

De seguida foram discutidas algumas questões relacionadas com QoE resultantes da utilização de um serviço *Cloud*. Por fim foi estabelecida uma relação entre o modelo proposto e os modelos de serviço existentes.

Segue-se uma descrição de plataformas e ferramentas de monitorização disponíveis que podem servir de suporte para a medição de alguns dos parâmetros mencionados no modelo estratificado proposto.

# Capítulo 4

# Plataformas e Ferramentas de Monitorização

A complexidade de um ambiente *Cloud Computing* leva a que o processo de monitorização seja controlado e implementado de várias maneiras. Fatores como o tipo de recursos em utilização, a natureza do tipo de serviços oferecidos (IaaS, PaaS, SaaS), a maneira como estes são implementados (públicos, privados ou híbridos) e os vários tipos de intervenientes (fornecedores, parceiros e utilizadores) levam à adoção de sistemas de monitorização diferentes. Um sistema de monitorização deve dar importância ao tipo de utilizador final, uma vez que estes têm diferentes necessidades. Por exemplo, um gestor de negócios necessita de dados de monitorização diferentes de um administrador de rede.

A *Cloud Computing* pode beneficiar de metodologias, conceitos e até de ferramentas existentes e consolidadas na gestão da computação distribuída [17]. O fornecimento de gráficos, estatísticas e outros aspetos relevantes da monitorização revela-se um processo essencial para os utilizadores gerirem e controlarem aspetos como a QoS e o cumprimento dos SLAs. Dependendo do serviço em questão, espera-se que uma infraestrutura *Cloud* forneça aos utilizadores capacidades para monitorizarem o serviço e efetuarem uma supervisão automática dos recursos que lhes foram atribuídos, em consonância com as permissões e acordos presentes no SLA.

É de salientar que os fornecedores de serviços *Cloud* muitas vezes disponibilizam serviços de monitorização (por vezes por um custo extra), tendo como objetivo avaliar a qualidade efetiva dos recursos entregues [38]. Um exemplo deste tipo de serviços é a Amazon CloudWatch.

Contudo, quando os fornecedores de serviços *Cloud* fornecem a sua própria monitorização,

os utilizadores devem estar atentos quanto à possibilidade da ocorrência de conflitos de interesse. Muitos dos fornecedores estão ansiosos por mostrarem tempos de disponibilidade maiores, por exemplo. Perante este cenário, a adoção de uma *third-party* neutra (ferramenta ou organização), responsável por monitorizar as infraestruturas e o desempenho de um fornecedor, parece ser a melhor solução. A monitorização passa a ser feita de uma maneira séria e independente. Por outro lado, é feita também uma auditoria independente aos SLAs, aumentando a credibilidade reivindicada pelos fornecedores. Esta solução beneficia tanto os utilizadores como a indústria desta tecnologia como um todo.

Neste contexto, este capítulo apresenta algumas das estratégias e ferramentas de monitorização de *Cloud Computing* existentes no mercado. O estudo dos conceitos, características, métricas e parâmetros presentes nas várias estratégias de monitorização esteve na base do modelo de monitorização estratificado proposto no Capítulo 3.

## 4.1 Frameworks

Esta secção apresenta algumas *frameworks* e sistemas de monitorização que procuram dar os primeiros passos na monitorização em *Cloud*. O estudo dos conceitos e características presentes nestas *frameworks* sustentou o modelo proposto no Capítulo 3. Estas abordagens são sobretudo viradas para o interior das organizações. Os sistemas de monitorização que se enquadram mais nos interesses dos utilizadores finais são abordados na Secção 4.2.

### 4.1.1 PCMONS

Nesta abordagem é tido como base o estudo efetuado em [17], onde os autores defendem que apesar das muitas soluções *Cloud* disponíveis, a monitorização e gestão ainda não está muito desenvolvida na área. A principal causa para este fator prende-se sobretudo com a falta de oferta de soluções *open source*. Perante esta situação, o objetivo dos autores passa por implementar um sistema de monitorização (PCMONS) em *Private Clouds*, com recurso a *software open source*, nomeadamente o Nagios (ferramenta descrita mais à frente neste capítulo). Este é um sistema que tem presente a ideia de que a monitorização pode beneficiar de ferramentas e conceitos já estabelecidos na gestão de computação distribuída. Antes de avançar para uma descrição mais detalhada do sistema de monitorização em si, é importante registar algumas das considerações iniciais que foram adotadas. Devido às características únicas de cada modelo de serviço, não é

possível chegar a uma solução de gestão genérica. Face a este facto, os autores tomaram a decisão de optarem por IaaS, devido sobretudo à sua flexibilidade, e por *Private Clouds*, uma vez que estão sob o controlo das políticas de segurança da respetiva empresa. Após uma análise das características deste tipo de *Clouds*, foi desenvolvida uma arquitetura de monitorização abstrata e genérica. A arquitetura do sistema de monitorização de *Private Clouds* é composta por três camadas, como é possível verificar na Figura 4.1. A camada correspondente às infraestruturas basicamente contem as instalações (*hardware* e rede), assim como *software* (OS, licenças, aplicações, *hypervisors*...). A camada do meio (*Integration Layer*) fornece uma separação abstrata entre os detalhes da infraestrutura e a informação de monitorização requerida pelos utilizadores. Esta é composta por vários módulos, que serão descritos em mais pormenor de seguida. A camada superior, correspondente à visualização, fornece uma interface onde através da análise das várias informações disponíveis, pode ser comprovado o cumprimento das políticas e SLAs estabelecidos.

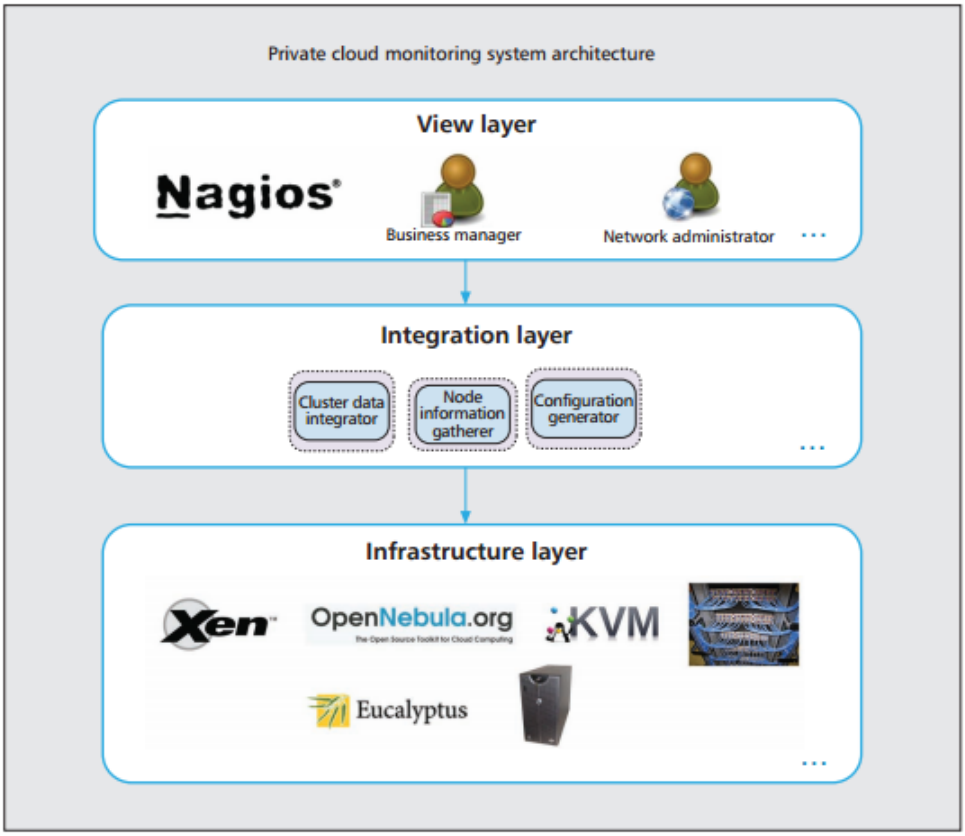



Figura 4.1: Arquitetura PCMONS [17].

Esta arquitetura equipara-se a um modelo centralizado onde é utilizada a ligação cliente/servidor. O sistema de monitorização PCMONS atua principalmente na camada de integração (*Integration Layer*), coletando informação relevante para disponibilizar na camada de visualização. O sistema é organizado em vários módulos, permitindo ao sistema ser adaptável e extensível (*plugins*) a outros cenários/ferramentas. Na Figura 4.2 está ilustrado um cenário típico da implementação da ferramenta de monitorização PCMONS. De seguida é feita a descrição das funções dos módulos ilustrados:

- **Node Information Gatherer**: responsável por recolher informações locais num nó da *Cloud*. Envia informações sobre as VMs locais para o *Cluster Data Integrator*.

- **Cluster Data Integrator**: como muitas das ferramentas *Cloud* organizam os nós em *clusters*, este módulo recolhe e prepara um conjunto de informações para a próxima camada. Evita a transferência de dados desnecessários de cada nó para o *Monitoring Data Integrator*.

- **Monitoring Data Integrator**: reúne e armazena dados numa base de dados para fins históricos. Fornece os dados ao *Configuration Generator*.

- **Configuration Generator**: recupera informações da base de dados e gera os ficheiros de configuração necessários às ferramentas de visualização.

- **Database**: local onde são guardados os dados necessários ao *Monitor Data Integrator* e *Configuration Generator*.

- **VM Monitor**: este módulo injeta *scripts* em VMs que enviam dados úteis a partir da VM para o sistema de monitorização (e.g., carga do processador ou memória utilizada).

- **Monitoring Tool**: módulo responsável por receber dados de monitorização de diversas fontes, assim como de armazená-los na base de dados para fins de consulta histórica.

- **User Interface**: interface para o utilizador, disponibilizada por algumas ferramentas de monitorização. Neste caso foi utilizada a interface do Nagios.

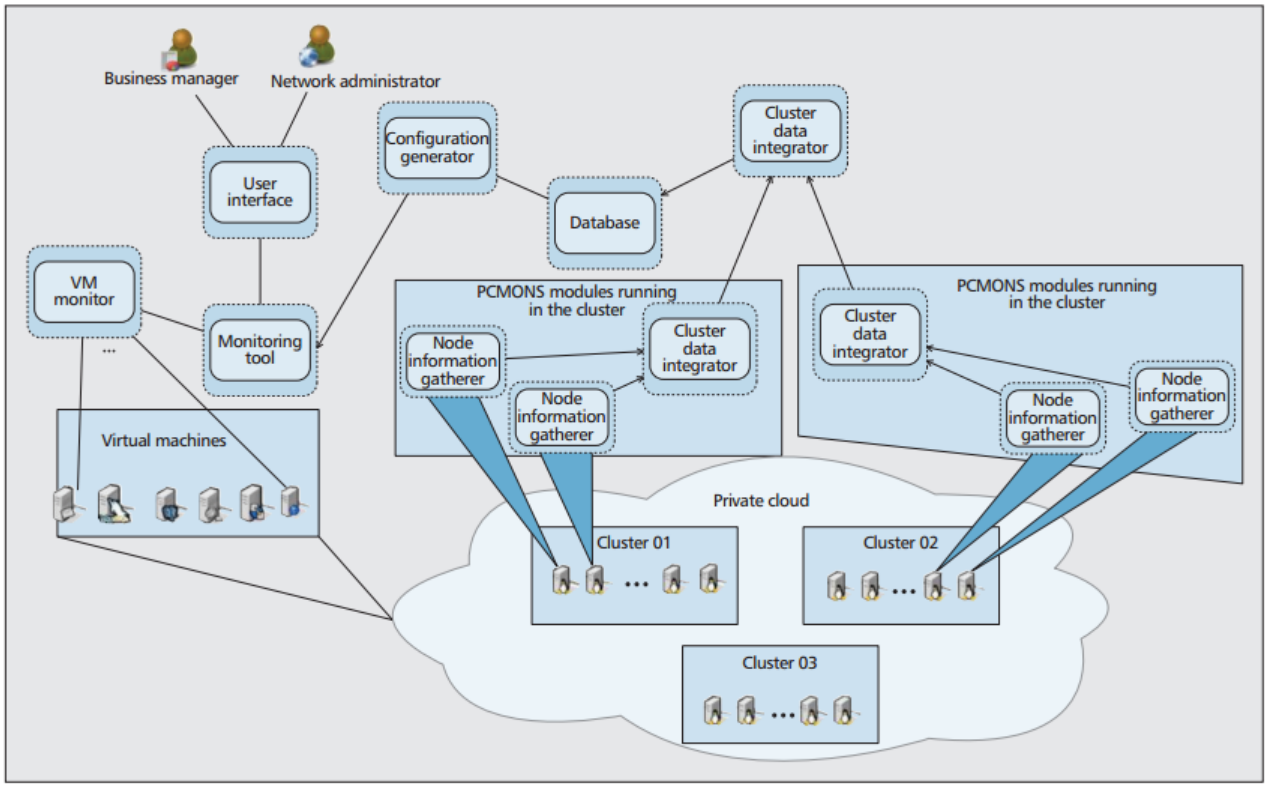



Figura 4.2: Cenário típico PCMONS [17].

### 4.1.2 Lattice

Para sustentar um funcionamento eficiente dos serviços *Cloud*, os autores em [19] desenvolveram uma *framework* de monitorização (Lattice), desenhada sobretudo para monitorizar recursos e serviços em ambientes virtuais. Para uma melhor compreensão do funcionamento e dos requisitos da conceção, são apresentados de seguida os pontos chaves da solução proposta. A *framework* de monitorização Lattice foi desenvolvida e implementada em conjunto com o projeto RESERVOIR. Resumidamente, o RESERVOIR é um serviço *Cloud* que distingue fornecedores de serviços dos fornecedores de infraestruturas e tem como objetivo aumentar a eficácia da computação, permitindo o desenvolvimento de serviços complexos. São abrangidas questões geográficas e de QoS, tentando também assegurar garantias de segurança. De seguida encontra-se descrita e ilustrada na Figura 4.3 a arquitetura do projeto RESERVOIR, que serviu de base para o desenvolvimento da *framework* Lattice.

**Arquitectura RESERVOIR**:

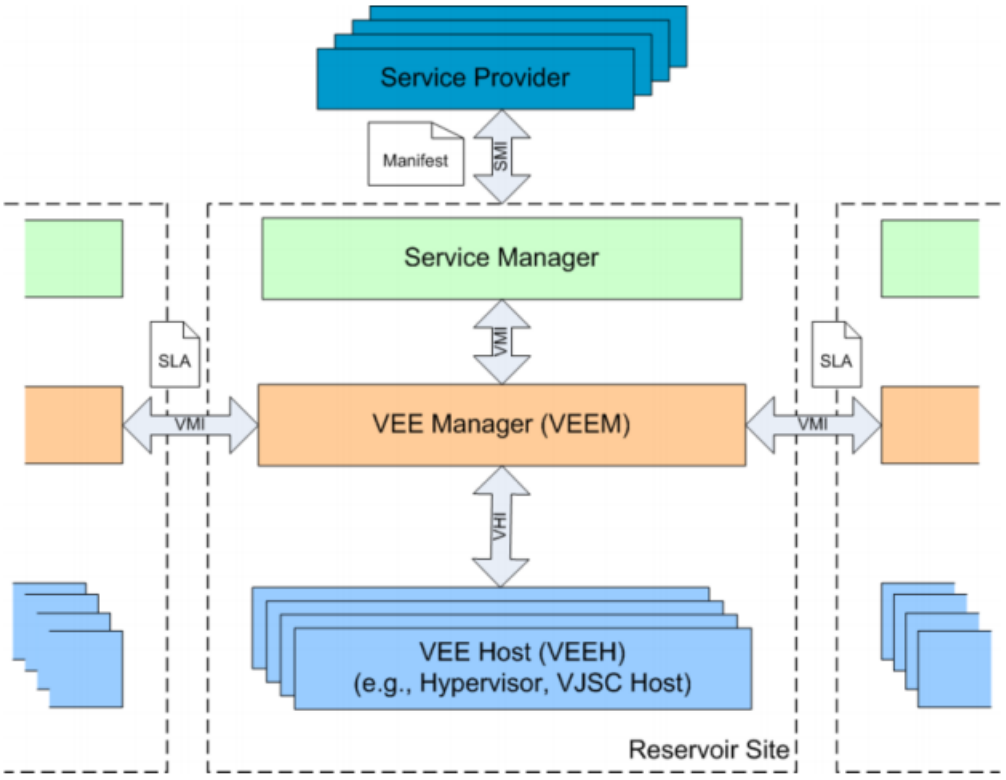



Figura 4.3: Arquitetura RESERVOIR [19].

- **VEE (Virtual Execution Environments)**: tal como o nome indica, são ambientes de execução virtual e o serviço *Cloud* RESERVOIR atua como uma plataforma para correr aplicações virtuais no VEE.
- **SP (Service Provider)**: específica os detalhes e requisitos da aplicação num *Definition Manifest*, assim como as regras de elasticidade e dos SLAs.
- **SM (Service Manager)**: responsável pela instanciação do serviço, solicitando a criação e configuração de VEEs para cada componente de serviço do *Manifest*. Também avalia e executa as regras de elasticidade estabelecidas pelo SP. Deve assegurar o cumprimento dos SLAs, recorrendo à monitorização em tempo real.
- **VEEM (VEE Manager)**: responsável por transformar VEEs em VEEHs (VEE hosts). Obedece ao SM, criando, movendo ou ajustando VEEs, tendo em vista a otimização do serviço.

- **VEEH (VEE Host)**: recursos que podem alojar certos tipos de VEEs, como máquinas físicas com Xen ou kvm *hypervisors*. O local de alojamento de um serviço *Cloud* normalmente possui vários VEEHs organizados num *cluster*.

Perante este tipo de sistema distribuído em grande escala, as fontes de medição que podem gerar dados são imensas. A recolha de toda essa informação ao mesmo tempo não se revela um processo eficiente. Os autores tiveram ainda o cuidado de implementarem um sistema que não fosse intrusivo, de modo a não afetar adversamente o desempenho do sistema ou de qualquer aplicação em execução. Perante estes fatores, a *framework* Lattice recorre a um sistema de monitorização de *probes* para coletar dados para o sistema de gestão. Para aumentar o poder e a flexibilidade da monitorização é introduzido o conceito de "fonte de dados" (*data source*). As fontes de dados podem conter de uma maneira dinâmica múltiplos *probes*. As medições estão programadas para serem tão pequenas quanto possível, assim como para transmitirem informações relevantes (ocorrências de alterações), tendo sempre em vista uma otimização do funcionamento da rede. Em conformidade com a arquitetura do projeto RESERVOIR foram delineadas três fontes de dados, nomeadamente:

- **Monitorização de recursos físicos**: são recolhidos três tipos de *probes*, onde mais especificadamente é monitorizada a utilização de CPU, memória e rede para cada VEEH.
- **Monitorização de recursos virtuais**: ocorre ao nível das máquinas virtuais que correm num *host* em particular. São recolhidos *probes* relativos à utilização do CPU, memória e rede.
- **Monitorização de aplicações de serviço**: os *probes* recolhidos neste tipo de monitorização são relativos às aplicações que correm no VEE.

Contudo, a *framework* Lattice não se limita somente a este tipo de fonte de dados. A sua conceção permite que tanto as fontes de dados como os próprios tipos de *probes* sejam desenhados e planeados conforme as necessidades e objetivos. Este é um fator que vem comprovar a mais-valia que é a flexibilidade da *framework*.

## 4.2 Ferramentas

Uma das preocupações constantes das empresas está relacionada com a monitorização e gestão dos serviços *Cloud*, pois é necessário identificar e reportar falhas, com vista a uma otimização dos serviços e respetiva satisfação dos clientes. A monitorização dos serviços *Cloud* é assim uma questão do interesse não só das empresas, mas também dos diversos intervenientes. Posto isto, é possível identificar no mercado algumas ferramentas de monitorização capazes de obedecer aos diferentes requisitos dos utilizadores. Os administradores de sistemas ou os próprios utilizadores finais podem monitorizar e gerir os seus recursos de várias formas. Na Tabela 4.1 estão indicadas algumas das ferramentas disponíveis, estando divididas de acordo com a técnica e paradigma a que obedecem. A monitorização local, tal como o nome indica, é feita localmente nos respetivos ambientes. Na monitorização remota, as ferramentas de monitorização são distribuídas e escaláveis, suportando sistemas de computação de alto desempenho, como *clusters* ou *grids*. Uma outra forma de monitorização é através de plataformas de gestão *web*, cuja oferta de mercado é maior, devido sobretudo à competitividade das empresas. De seguida são abordados alguns exemplos para os três tipos de ferramentas de monitorização.

Tabela 4.1: Algumas ferramentas de monitorização.

| **Tipo** | **Exemplos** |
|---|---|
| Local | Sysstat (Isag, Ksars), Dstat. |
| Remota | Nagios, Ganglia, GroundWork, Cacti, MonALISA, GridICE. |
| Plataformas de Gestão Web | RightScale, Landscape, Amazon CloudWatch, Gomez, Hyperic/Cloud Status, 3Tera, Zenos, Logic Monitor, Nimsoft, Monitis, Kaavo, TechOut, Tap in systems, CloudKick, Enstratus, YLastic, ScienceLogic, Keynote, NewRelic. |

### 4.2.1 Monitorização Local

#### Sysstat

O Sysstat [39] é uma ferramenta de monitorização local e fornece uma visão geral do desempenho atual de um sistema.

**Principais Características:**

- Inclui quatro grupos de ferramentas de monitorização (sar/sadc/sadf, iostat/nfsiostat/cifsiostat, mpstat, pidstat) para uma análise global do desempenho de um sistema.
- Pode monitorizar um grande número de métricas, entre as quais: taxas de transferência de entrada/saída (globais, por dispositivo, por partição, por rede e por tarefa); estatísticas do CPU (globais, por CPU e por tarefa), incluindo suporte para arquiteturas virtualizadas; estatísticas da utilização de memória; estatísticas da memória virtual, *paging* e de falhas; memória *Per-task (per-PID)* e estatísticas de falhas das páginas; estatísticas globais do CPU e de erros existentes em determinadas tarefas; atividade do processo de criação; estatísticas de interrupções (globais, por CPU, incluindo potenciais fontes de interrupção APIC - *Advanced Programmable Interrupt Controller*, interrupções por *hardware* e *software*); extensas estatísticas de rede; atividade do servidor e cliente *Network File System* (NFS); estatísticas *Socket*; estatísticas das filas e do sistema em execução; estatísticas da utilização de tabelas internas do *kernel*; alternância de tarefas do Linux e do sistema; estatísticas de *Swapping*; estatísticas da gestão de energia (frequência do *clock* do CPU instantânea e média, temperaturas dos dispositivos, entradas de tensão, dispositivos USB conectados ao sistema).
- Os valores médios das estatísticas são calculados ao longo de um período de amostragem.
- A maioria das estatísticas do sistema podem ser guardadas num ficheiro para futuras inspeções.
- Permite configurar o tamanho dos dados de histórico a serem guardados.
- Deteta novos dispositivos (discos, interfaces de rede, etc) que são criados ou registados dinamicamente.
- Suporta máquinas *symmetric multiprocessing* (SMP) e UP, incluindo máquinas com processadores *hyperthreaded* ou *multi-core*.
- Suporta CPUs *hotplug* (são detetados automaticamente processadores que são desligados ou ligados durante a execução) e *tickless* CPUs.
- Trabalha em diversas arquiteturas (32 ou 64 bit).
- Necessita de pouco processamento para correr (escrito em linguagem C).

- As estatísticas do sistema podem ser exportadas em vários formatos, como *Comma-separated values* (CSV), *Extensible Markup Language* (XML), entre outros.
- Suporta um grande número de línguas.
- Existe um grande número de programas que usam a informação contida no Sysstat e a disponibilizam em modo gráfico (isag - incluído no Sysstat, Ksars, entre outros).

### Dstat

O Dstat [40] é também uma ferramenta de monitorização local e fornece informações sobre os recursos em tempo real. Pode ser utilizada com o propósito de monitorizar testes de ajuste de desempenho, *benchmarks* ou mesmo para solucionar problemas. O Dstat essencialmente recolhe as mesmas informações que o Sysstat, mas exibe a informação num formato mais agradável para o utilizador. Esta ferramenta foi uma das utilizadas na monitorização do teste prático abordado no Capítulo 5.

**Principais Características:**

- Combina a informação dos módulos vmstat, iostat, ifstat e netstat.
- Mostra as estatísticas no mesmo espaço de tempo e de maneira precisa.
- Implementa contadores para ordenar as sequências de estatísticas, o que ajuda durante a análise das mesmas.
- É escrito em python e é facilmente extensível.
- Inclui muitos *plugins* externos.
- Pode fazer sumários dos números totais por grupos de blocos ou de dispositivos de rede.
- Pode mostrar interrupções por dispositivo.
- Mostra as unidades exatas e limita os erros nas conversões.
- Indica diferentes unidades com cores diferentes.
- Mostra os resultados intermédios quando o atraso é maior que 1 s.
- Permite a exportação dos dados para CSV, permitindo a geração de gráficos em Excel, ou outro tipo de ferramentas.

### 4.2.2 Monitorização Remota

#### Nagios

O Nagios [41] é uma das ferramentas de monitorização de rede mais populares. É uma ferramenta *open source*, distribuída sob a licença GPL e possui uma equipa de programadores que mantém a aplicação ativa, onde constantemente são lançados novos *plugins*, quer oficiais quer não oficiais. A sua versão atual e estável é a 3.4.1. A monitorização pode ser efetuada tanto em *hosts* como em serviços, permitindo detetar e reparar problemas nas infraestruturas das TI, antes que afetam os utilizadores finais.

**Principais Características:**

- Monitoriza vários componentes, como métricas do sistema, serviços de rede (SMTP, POP3, HTTP, NNTP, ICMP, SNMP), aplicações, serviços, servidores e infraestruturas de rede.
- Suporta a monitorização remota através de túneis criptográficos *Secure Shell* (SSH) ou *Secure Sockets Layer* (SSL).
- Notifica quando um serviço ou equipamento apresenta problemas e quando o problema é resolvido (via email, SMS ou qualquer outro meio definido pelo utilizador por *plugin*).
- Permite o desenvolvimento simples de *plugins*, facilitando aos utilizadores criarem os seus próprios modos de monitorização, dependendo das suas necessidades. Podem ser utilizadas diversas linguagens de desenvolvimento (Bash, C, Perl, Python, PHP, C#, etc.).
- Verifica serviços em paralelo, ou seja, em caso de vários serviços estarem a ser monitorizados em simultâneo não existe o risco de perda de informação.
- Possui a capacidade de definir tratadores de eventos que executam tarefas em situações pré-determinadas ou para a resolução pró-ativa de problemas.
- Permite assegurar que os SLAs da organização estão a ser cumpridos.
- Interface *web* intuitiva e prática, que permite uma boa visualização do estado atual da rede, notificações, histórico de problemas, arquivos de log, etc.

### Ganglia

O Ganglia [42, 43] é um sistema de monitorização distribuído e escalável para sistemas de computação de alto desempenho, como *clusters* e *grids*. É uma ferramenta *open source*, distribuída sob a licença BSD e a sua versão atual é a 3.5.2. Aproveita conceitos e tecnologias amplamente utilizadas, tais como XML para a representação dos dados ou *External Data Representation* (XDR) para um transporte de dados portátil e compacto. Para armazenamento e visualização dos dados, esta ferramenta utiliza o sistema RRDtool (Round Robin Database). Este sistema permite o armazenamento de sequências temporais de dados de forma compacta numa base de dados circular e de tamanho constante. Com base nos dados armazenados, o RRDtool gera gráficos que mostram a evolução de uma ou mais métricas ao longo do tempo. As métricas recolhidas pela ferramenta vão desde a percentagem de utilização de CPU, utilização de memória e rede, entre outras informações relativas ao *cluster*. Por outro lado, o sistema também suporta métricas definidas pelo utilizador, onde são recolhidas informações por uma aplicação externa. Esta característica confere ao Ganglia um carácter extensível, facilitando a sua modificação ou integração com outros sistemas.

Esta ferramenta utiliza uma estrutura de dados e algoritmos cuidadosamente projetados para não introduzirem *overhead* significativo em cada nó, contribuindo para uma boa escalabilidade do sistema. A sua implementação é feita duma forma robusta e já foi implementada em vários OS e arquiteturas, estando atualmente em utilização num grande número de *clusters* por todo o mundo.

O seu funcionamento (ilustrado na Figura 4.4) baseia-se no modelo cliente/servidor e divide-se em dois módulos: o gmond (Ganglia Monitoring Daemon) e o gmetad (Ganglia Meta Daemon). O gmond reside em cada um dos nós a serem geridos, monitorizando as mudanças de estado nos nós e anunciando mudanças relevantes. O gmetad normalmente está situado no servidor *web*, agrupando os dados provenientes dos nós. A interface de visualização fornece uma visão sobre a informação recolhida, com recurso a páginas *web* dinâmicas e em tempo real.

Esta ferramenta foi outra das soluções utilizadas no processo de monitorização levado a cabo no teste prático abordado no Capítulo 5.

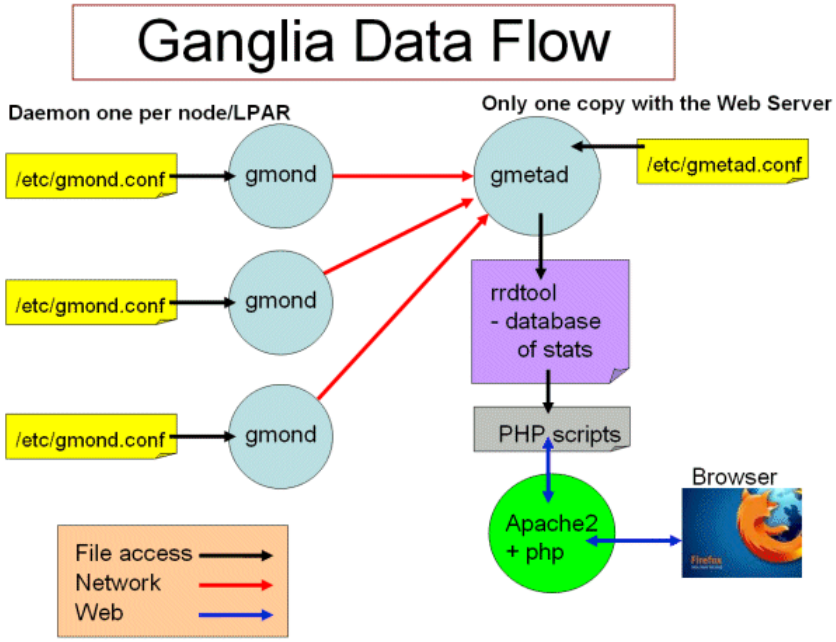



Figura 4.4: Arquitetura Ganglia [43].

### GroundWork

O GroundWork [44] é uma ferramenta de monitorização baseada numa combinação de tecnologias e ferramentas *open source* bem conhecidas como JBOSS, Apache, Nagios, Cacti e MySQL. Com recurso a este tipo de ferramentas é efetuada a recolha de dados. O sistema é complementado com outras ferramentas, de maneira a preencher lacunas. Com este tipo de abordagem, a arquitetura do sistema torna-se aberta e extensível, sem abdicar da segurança.

É utilizada uma arquitetura de três níveis para permitir a máxima potência e flexibilidade na implementação do GroundWork. Os três níveis estão incluídos num único servidor (GroundWork Enterprise Monitor), embora em grandes implementações a base de dados e o servidor de *polling* possam ser distribuídos por razões de desempenho. Os três níveis estão ilustrados na Figura 4.5 e descritos de seguida.

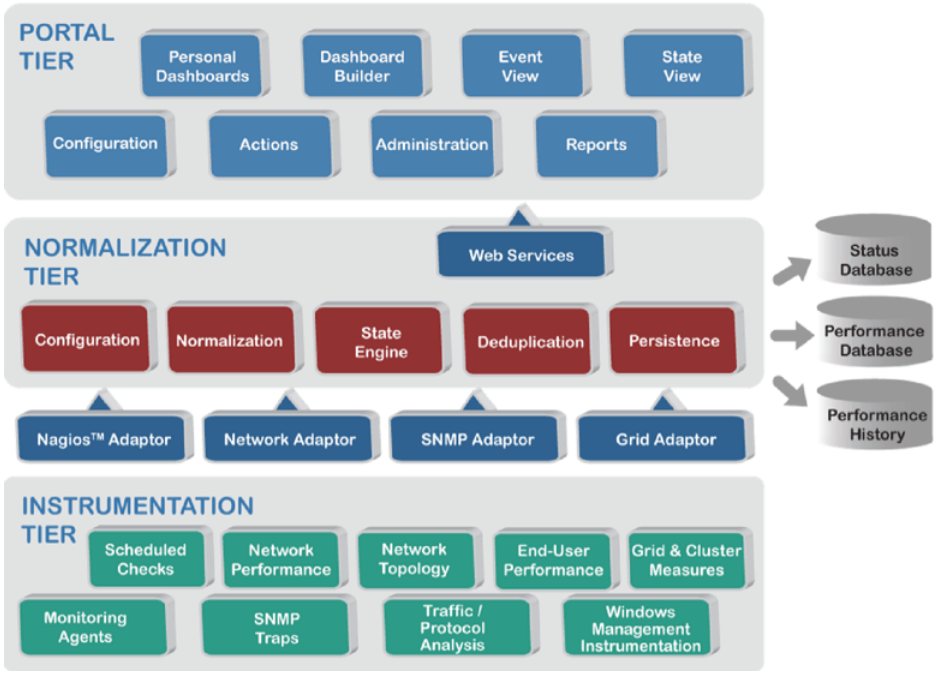



Figura 4.5: Arquitetura GroundWork [44].

**Nível 1 - Instrumentação:**

Esta camada utiliza uma variedade de componentes. A recolha de dados é realizada pelo Nagios, Cacti e por outros componentes opcionais que usam um padrão de recolha bem definido. Os dados são capturados quando ocorrem mudanças de estado, por eventos ou medidas de desempenho definidas.

**Nível 2 - Normalização:**

Por sua vez, esta camada armazena os dados recolhidos de uma forma estruturada e apresenta-os sob pedido através de serviços *web* ou de consultas na base de dados. Pode ser vista como a base de dados do nível seguinte.

**Nível 3 - Portal:**

Este nível corresponde à camada visualizada pelos utilizadores. Os dados armazenados na camada anterior são visualizados em tempo real e de maneira dinâmica, sob a forma de gráficos de desempenho, eventos agendados e representações do estado atual de serviços.

### 4.2.3 Plataformas de Gestão Web

#### RightScale

O RightScale [45] é a ferramenta líder de um conjunto de soluções de gestão de *Clouds* disponíveis. Esta ferramenta fornece uma versão completa paga, mas também disponibiliza uma versão gratuita com algumas limitações, desenhada para introduzir e publicitar o produto.

Através de uma solução única e integrada, o RightScale permite gerir toda uma infraestrutura *Cloud*, proporcionando visibilidade e controlo sobre todos os recursos em um só lugar. Esta plataforma é extremamente versátil e permite uma grande variedade de cenários e diferentes abordagens.

Os principais componentes da sua arquitetura estão ilustrados na Figura 4.6 e descritos de seguida.


Figura 4.6: Arquitetura RightScale [45].

**MultiCloud Platform:**

Esta camada fornece um controlo remoto universal (*Dashboard* ou API) para aceder a um conjunto de recursos.

- Suporta a gestão dos vários tipos de modelos de implementação *Cloud* (*Public*, *Private* e *Hybrid*). Os serviços compatíveis mais populares são os AWS, Rackspace, Windows Azure, CloudStack, Eucalyptus, OpenStack, entre outros.

- Suporta a gestão dos vários tipos de recursos (computação, armazenamento e rede) e a configuração dos recursos de rede, como endereços IP, redes de distribuição de conteúdo, *firewalls* e VPN. Permite a gestão dos volumes de armazenamento, permitindo a ativação do serviço *web*, distribuição de conteúdos e *backups*.

- Os recursos são organizados em grupos lógicos de servidores, permitindo configurações, desenvolvimentos, testes, preparação e automatização de servidores individuais ou implementações inteiras.

- Através do *Dashboard* e API é oferecida uma gama de recursos: recursos *Cloud* (configurar, editar e desativar); monitorização (gráficos de métricas personalizados); auditoria (acompanha e exporta *logs* de auditoria); provisionamento (gestão de contas, utilizadores e permissões).

**Configuration Framework:**

Na *framework* de configuração (*ServerTemplate*) são fornecidos planos *Cloud* inteligentes para configurar e operar os servidores de forma dinâmica e totalmente personalizável. O *ServerTemplate* é construído a partir de imagens modulares, *scripts* e *inputs* variáveis.

- A sua natureza modular e dinâmica permite aos *ServerTemplate* serem completamente personalizáveis. Permite escolher o tipo de *Cloud* e OS, tipos de instâncias, parâmetros de rede, alertas, *scripts* adequados aos pacotes instalados, entre outros aspetos.

- Cada elemento de um *ServerTemplate* tem a sua versão controlada, isto é, proporciona um comportamento reprodutível, previsível, repetitivo e fácil de solucionar. As configurações podem ser clonadas, personalizadas e republicadas de forma consistente para uso público ou privado.

- As operações podem ser rapidamente transpostas em *scripts* usando linguagens comuns, como Bash, Perl, Powershell, Python e Ruby.
- Existe uma abstração das diferenças específicas das *Clouds*, garantindo uma configuração multi-*Cloud* consistente através de recursos públicos, privados ou híbridos.

**MultiCloud Marketplace:**

Este é um mercado que oferece *ServerTemplates*, *scripts* e arquiteturas pré-construídas e publicadas pelo RightScale, parceiros ou os próprios utilizadores. Existe uma grande variedade de soluções rigorosamente testadas, entre as quais:

- App Servers (Apache, IIS, Nginx, Passenger, Tomcat, Websphere, Zend);
- Big Data (IBM Hadoop, Jaspersoft, Univa Grid);
- Content Management (BlogEngine, Drupal, Joomla, Mephisto, Wordpress);
- Load Balancers (aiCache, HAProxy, Varnish Cache, Zeus Traffic Manager);
- NoSQL Databases and Caches (Membase, Memcached, MongoDB);
- Operating Systems (CentOS, Red Hat Enterprise Linux (RHEL), Ubuntu, Windows Server);
- Security ( Microsoft Active Directory, Trend Micro, Zeus Traffic Manager);
- SQL Databases ( IBM DB2, MySQL, PostgreSQL, SQL Server);
- Web Servers (Apache, IIS, Nginx).

**Automation Engine:**

Esta camada fornece um conjunto de ferramentas que permitem monitorizar, dimensionar e gerir os servidores implementados de maneira eficiente, elástica e confiável. As suas funções permitem:

- Manter o controlo dos recursos com a monitorização dos servidores e aplicações. É possível criar visualizações personalizadas (gráficos, *plugins*, *widgets*);

- Visualizar grandes implementações, com centenas de servidores, através da monitorização de *clusters*;
- Atribuir alertas a qualquer tipo de métrica, com o intuito de notificar a existência de problemas;
- Dimensionar os recursos *Cloud* de maneira elástica com o recurso a matrizes que escalam automaticamente;
- Através da utilização de *macros* é possível activar arquitecturas inteiras;
- Automatizar de aplicações e bases de dados com recurso a etiquetas, bibliotecas e *scripts*.

**Governance Controls:**

Por sua vez, esta camada é responsável por permitir a vigilância de aspetos como os acessos, segurança, auditoria, relatórios e orçamentos.

- Através da gestão dos acessos e da segurança é possível controlar autenticações, permissões e credenciais.
- A auditoria permite resolver e rastrear problemas. Permite a análise dos *logs* do servidores, configurações chaves da *firewall* e SSH.
- Com o relatório de contas é possível controlar os recursos e os orçamentos. Existe a possibilidade de visualizar as infraestruturas utilizadas pela *Cloud* e o custo associado a cada item.

### Landscape

A ferramenta Landscape [46] é um sistema de gestão desenvolvida pela Canonical e permite monitorizar, gerir e atualizar toda uma infraestrutura a partir de uma única interface. Inclui características como acesso remoto, gráficos informativos, alertas em tempo real, gestor de tarefas, entre outros. Esta é uma ferramenta paga. Esta ferramenta é caracterizada pelo sistema de gestão e de monitorização e pela gestão de *Clouds*.

**Sistema de Gestão:**

Tarefas como o acompanhamento e atualização das configurações de todo um sistema ocupam bastante tempo aos administradores das TI. O Landscape por sua vez permite recolher toda a

informação detalhada de todas as configurações e atualizações efetuados, tornando o diagnóstico e a auditoria dos servidores um processo fácil. Esta ferramenta proporciona também a criação de scripts que automatizam rotinas como iniciar e parar serviços, realizar *backups*, traçar perfis de *hardware*, etc. Permite ainda:

- Gerir várias máquinas ao mesmo tempo, permitindo que uma operação possa ser aplicada a mais do que uma máquina;
- Agrupar máquinas em diferentes grupos, consoante os diferentes requisitos, permitindo dar resposta a diferentes tarefas e realizar a gestão apenas por uma interface;
- Gerir pacotes através da rede, onde através da consulta de um inventário, estes são geridos em cada máquina (instalar, atualizar ou remover);
- Gerir utilizadores de um ou mais sistemas de uma forma fácil;
- Controlar atualizações de segurança de uma forma eficiente e rápida;
- Suportar sistemas desconectados, sendo devidamente tratados quando ficarem *online*.

**Sistema de Monitorização:**

Através da plataforma do Landscape é possível observar informação detalhada sobre o *hardware* e as estatísticas de utilização. A monitorização do sistema é efetuada com recurso a uma aplicação cliente instalada em cada máquina. Por sua vez, o cliente envia informação (desde temperaturas, utilização de disco e de memória, carga de um sistema a métricas personalizadas) em tempo real para o Landscape. Permite ainda:

- Rápido conhecimento e compreensão do consumo dos recursos do sistema;
- Personalizar a interface apenas com os dados que interessam;
- Obter a informação que se deseja de uma forma automática cada vez que se entra na interface do Landscape;
- Gerir de uma maneira central os processos ativos;
- Mostrar de uma forma simples todo o *hardware* existente na infraestrutura;
- Simplificar as auditorias de segurança, através de *logs* de histórico detalhados.

**Gestão de Clouds:**

Permite gerir as instâncias do *Ubuntu Enterprise Cloud* (UEC) e da Amazon EC2 de uma forma fácil, utilizando ferramentas dedicadas que ajudam a observar quem é que está a utilizar determinados recursos e quando. Como as instâncias do UEC e a Amazon EC2 são suportadas, é possível correr as mesmas imagens quer numa *Public Cloud* ou numa *Private Cloud*. Permite ainda:

- Configurar, iniciar, parar e atualizar uma *Private Cloud* com o UEC ou uma *Public Cloud* com a Amazon EC2;
- Ver a topologia *Cloud* para uma melhor gestão do sistema e da rede;
- Configurar zonas disponíveis e de armazenamento seguro para chaves de acesso para *Private Clouds* ou *Public Clouds*;
- Configurar o *Elastic Block Storage* (EBS) na Amazon EC2;
- Alocar IPs ”elásticos”.

### Kaavo

O Kaavo [47] é uma companhia de gestão de *Cloud Computing* e o seu principal serviço é o IMOD. Esta ferramenta proporciona a gestão de aplicações distribuídas, cargas de trabalho e ambientes TI. A sua abordagem de gestão é centrada nas aplicações e a tecnologia resultante encontra-se com a patente pendente. Os autores defendem que as empresas tradicionalmente gerem os seus recursos e servidores recorrendo a uma abordagem centrada nas infraestruturas, onde os recursos são geridos individualmente. Este processo revela-se complexo e caro, dificultando o crescimento do negócio. Com uma abordagem centrada nas aplicações, a gestão incide em aplicações específicas, em vez de servidores e *routers*. Todos os recursos necessários a uma específica aplicação são geridos através dum sistema global.

Uma das principais inovações inseridas pelo Kaavo foi a capacidade de capturar o comportamento da gestão do tempo de execução de qualquer aplicação complexa ou carga de trabalho em um único documento XML. Outra das grandes características do serviço IMOD prende-se com a possibilidade de efetuar uma gestão multi-*Cloud*, ou seja, através de vários vendedores e dos vários modelos de implementação (*Public*, *Private* e *Hybrid*). A sua arquitetura de funcionamento está ilustrada na Figura 4.7. As *Clouds* e serviços suportados são a Amazon EC2,

CloudStack, Eucalyptus, HP, IBM, Logicworks, OpenStack, RackSpace, Terremark e vCloud Director. O serviço IMOD (pago) está disponível através de uma solução de gestão SaaS ou instalado diretamente no local para clientes corporativos.

**IMOD - Principais Características:**

- Implementação simples de aplicações seguras: com um único clique, o sistema IMOD disponibiliza sistemas de múltiplos servidores, configurações de dados e de segurança de acesso, *middleware*, implementa aplicações e cargas de trabalho e inicia serviços na ordem correta.
- Agendamento e automação: o "piloto automático" do IMOD permite executar tarefas complexas em determinados momentos ou em resposta a eventos pré-configurados, sem intervenção manual.
- Modelos de implementação de boas práticas: o IMOD simplifica a criação e partilha de configurações complexas usando um único sistema de definição de arquivo. Este processo torna as boas práticas um padrão e permite fornecer um ambiente consistente para o desenvolvimento, teste e produção.
- Contabilização: o IMOD rastreia a utilização dos recursos *Cloud* por cada aplicação, resultando numa medição e faturação precisa dentro e entre os departamentos.

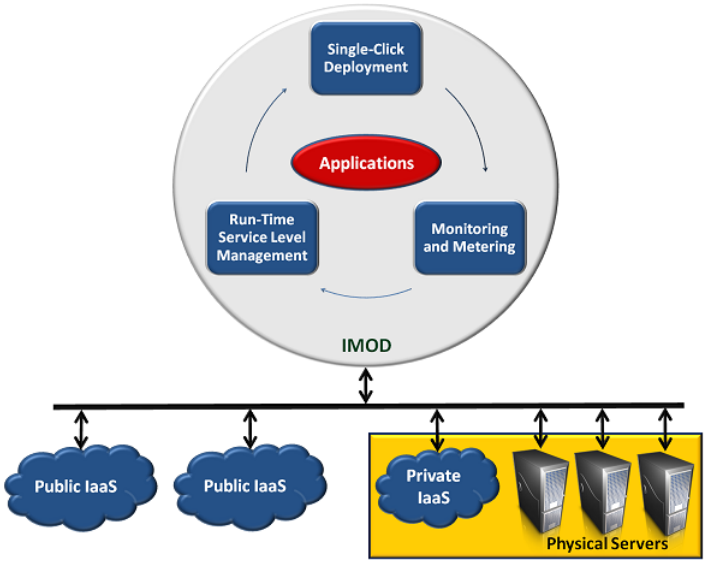



Figura 4.7: Arquitetura Kaavo [47].

## 4.3 Sumário

Neste capítulo foram apresentadas algumas das plataformas e ferramentas de monitorização de *Cloud Computing* existentes no mercado. Inicialmente foi feita uma contextualização acerca das razões que levam à existência de diferentes tipos de sistemas de monitorização. De seguida foram apresentadas duas estratégias de monitorização, nomeadamente o PCMONS e a Lattice. O PCMONS é um sistema de monitorização para *Private Clouds* e recorre a *software open source*. Por sua vez a Lattice é uma *framework* de monitorização desenhada sobretudo para monitorizar recursos e serviços em ambientes virtuais. Por último foram apresentadas algumas das ferramentas de monitorização existentes, agrupadas segundo o paradigma a que obedecem (monitorização local, remota ou plataformas de gestão *web*). Algumas dessas ferramentas foram descritas em maior detalhe. Em particular, as ferramentas Ganglia e Dstat vão ser usadas no capítulo que se segue.

# Capítulo 5

# Cenário Prático

Neste capítulo é apresentado um cenário prático que tem como objetivo permitir explorar a monitorização de um ambiente *Cloud*, classificando as métricas obtidas segundo o modelo proposto no Capítulo 3. Para o efeito foi utilizada a *Cloud* do Departamento de Informática da Universidade do Minho, assim como as ferramentas de monitorização aí instaladas[1]. De seguida é apresentado o ambiente de testes e a metodologia utilizada. Por fim é feita a análise dos resultados obtidos.

## 5.1 Ambiente de Testes

Tal como referido anteriormente, como ambiente de testes utilizou-se a *Cloud* implementada no Departamento de Informática da Universidade do Minho, onde se encontram a decorrer alguns projetos e testes. Deste modo, foi utilizado um dos testes em curso, onde estava a ser utilizada a base de dados HBase. Neste teste o objetivo passa por avaliar o HBase num ambiente *multi-tenant* utilizando o YCSB como um gerador de cargas de trabalho. Deste modo foram configuradas diferentes cargas de trabalho em simultâneo com o intuito de se simular um cenário *multi-tenant*, característico de um ambiente *Cloud*. Com a colaboração de um dos responsáveis por este projeto, chegou-se a uma solução para o processo de monitorização adequado aos obje-

[1]Este é um primeiro passo no sentido de explorar a utilização do modelo proposto. Futuramente estão previstos testes mais diversificados, explorando diferentes cenários de carga e respetivo impacto nas métricas de cada camada, mostrando em que medida a estratificação da monitorização clarifica a análise dos serviços prestados e a causa de possíveis problemas. Por motivos de limitação de tempo e logística, não foi possível implementar em tempo útil testes mais elaborados.

tivos desta etapa da dissertação. Para a recolha de dados estatísticos foram utilizadas ferramentas de monitorização abordadas no Capítulo 4, nomeadamente o Dstat e o Ganglia. Na Figura 5.1 está esquematizado o cenário de testes, onde é possível identificar os componentes que vão ser monitorizados. Como o objetivo do cenário prático passa por explorar a monitorização de um ambiente *Cloud*, através deste cenário espera-se obter métricas cuja classificação se enquadre no modelo proposto.

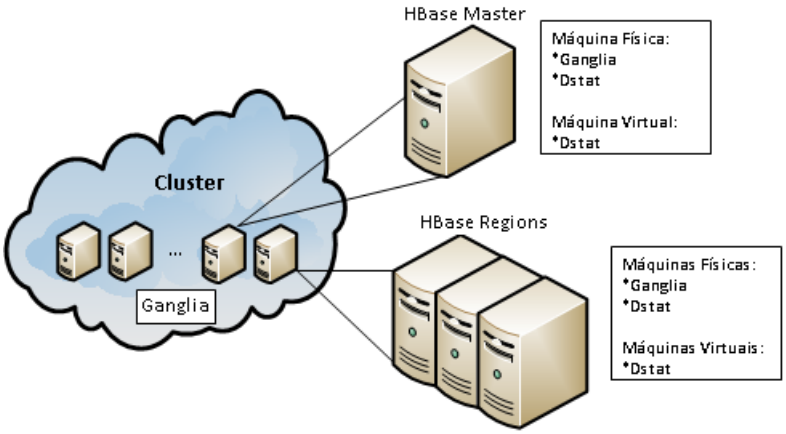



Figura 5.1: Cenário de testes.

Quanto às características do ambiente de testes, o *cluster* tem no total 36 máquinas com *dual cores* com *hyper threading* Intel i3 CPU de 3.1GHz, 4 GBytes de memoria e um disco local 7200 RPM SATA. No momento em que se efetuou o teste, 18 delas estavam a correr OpenStack (ferramenta de implementação de serviços IaaS abordada na Secção 2.4.6). A versão utilizada é a *diablo* e uma das máquinas estava a correr o *nova-controller* e as outras eram *nova-computes*. As VMs por sua vez, têm 3 Gbytes de memória, 10 GBytes de disco e 4 *virtual* CPUs.

No que diz respeito aos testes em curso, a monitorização do ambiente *Cloud* incidiu na altura em que estava a ser corrido o HBase sobre as máquinas físicas e virtuais (disponibilizadas pelo Openstack), de maneira a ser possível monitorizar a carga introduzida no *cluster*.

O HBase [48] é uma base de dados NoSQL distribuída e escalável. Os sistemas de base de dados NoSQL lidam com enormes quantidades de dados e a natureza dos dados não requer um modelo relacional para a estrutura de dados. Este tipo de base de dados é típico de um ambiente *Cloud*. A base de dados HBase é *open source*, escrita em Java e desenvolvida como

parte do projeto Hadoop da Apache Software Foundation. Este é um projeto que tem como objetivo hospedar tabelas muito grandes (bilhões de linhas por milhões de colunas) em cima de aglomerados de *hardware*.

Tendo em consideração o presente cenário, de uma maneira resumida podem identificar-se dois tipos de elementos que constituem o HBase, nomeadamente o *Master* e as *Regions*. O *Master* implementa um *Master Server*, responsável por monitorizar todas as instâncias *Regions Server* (por sua vez responsáveis pelo atendimento e gestão das regiões) no cluster, e é a interface para todas as alterações de metadados. As regiões (*Regions*) são o elemento básico de disponibilidade e de distribuição para uma tabela.

No presente caso, quando foi lançado o HBase foi utilizado um *Master* e seis *Regions Server*. Na Tabela 5.1 é possível identificar quais as características das instâncias onde correu o HBase, nomeadamente o ID das máquinas reais e endereços IP das respetivas máquinas virtuais. Esta informação é importante para o processo de identificação das máquinas do *cluster* envolvidas neste processo.

Tabela 5.1: Características das Instâncias.

| **Nome da Instância** | **ID Máquina Real** | **IP Máquina Virtual** |
|---|---|---|
| Master | Cloud05 | 10.0.108.3 |
| Region1 | Cloud17 | 10.0.108.5 |
| Region2 | Cloud35 | 10.0.108.9 |
| Region3 | Cloud01 | 10.0.108.7 |
| Region4 | Cloud33 | 10.0.108.27 |
| Region5 | Cloud25 | 10.0.108.13 |
| Region6 | Cloud20 | 10.0.108.15 |

## 5.2 Metodologia

No processo de monitorização do ambiente de testes descrito anteriormente, recorreu-se às ferramentas Ganglia e Dstat (abordadas no Capítulo 4). Devido à utilização de ambas as ferramentas nos projetos em curso, estas já se encontravam devidamente instaladas e configuradas, dada a sua utilidade para a monitorização do desempenho, *troubleshooting*, entre outros aspetos.

Todas as máquinas físicas que constituem o *cluster* são monitorizadas pelo Ganglia (ver Figura 5.1). Através da interface de visualização é possível observar as diversas estatísticas recolhidas de uma maneira global (*cluster*) ou em específico para cada máquina física.

Por sua vez, o Dstat foi ativado em todas as máquinas físicas e virtuais envolvidas (ver Tabela 5.1), antes de se dar início ao teste. Para tal, acedeu-se via SSH[2] a cada máquina física e virtual e digitou-se o seguinte comando:

- dstat - -output Log "ID da máquina".CSV

Este comando ativa o dstat na respetiva máquina e guarda os dados que vai recolhendo num ficheiro CSV. Os *logs* das máquinas físicas foram identificados pelo respetivo ID (e.g., LogReal05) e os das máquinas virtuais pelo último octeto dos seus endereços IP (e.g., LogVM3). Na Figura 5.2 está ilustrada a maneira como a informação é exibida, assim como as diversas estatísticas recolhidas. Estas dizem respeito à percentagem de utilização total do CPU (subdividida por *usr* - utilizador, *sys* - sistema, *idl* - desocupada, *wai* - em espera, *hiq* - *hardware interruptions* e *siq* - *software interruptions*), velocidades de escrita e leitura no disco, quantidade de dados de rede recebidos e enviados, *paging* (técnica de gestão de memória utilizada pelo sistema, onde pode ser observada a memória trocada de/para o disco), e estatísticas do sistema (interrupções e *context switches*).

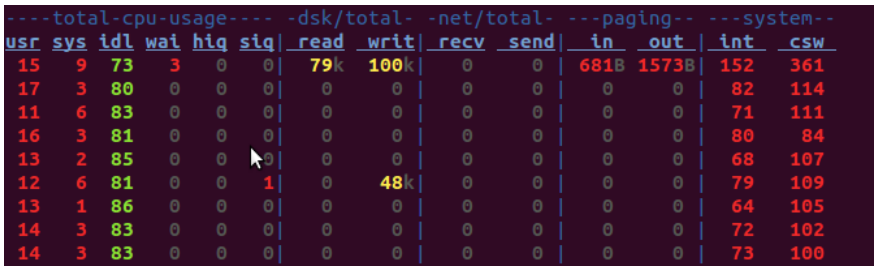



Figura 5.2: Exemplo da informação exibida pelo Dstat.

A partir deste momento estão reunidas as condições necessárias para dar início ao teste. O HBase foi lançado nas máquinas referidas anteriormente durante aproximadamente 15 minutos.

[2]Por motivos de confidencialidade relativos aos projetos em curso, não são ilustrados os procedimentos de acesso às várias máquinas.

Por fim fez-se a recolha dos *logs* gerados pelo Dstat e foi observada a informação gráfica disponibilizada pelo Ganglia. Nos *logs* do Dstat foi necessário fazer algum tratamento de dados de modo a serem gerados gráficos[3] coerentes. Para tal estabeleceu-se um instante temporal padrão (último *log* iniciado precisamente antes do início do teste), sendo os restantes *logs* editados de forma a iniciarem-se no mesmo período de tempo.

De modo a complementar o processo de monitorização levado a cabo pelas duas ferramentas de monitorização abordadas anteriormente, procedeu-se também à recolha de *logs* gerados pelas cargas de trabalho implementadas no teste. Os *logs* foram recolhidos nas máquinas clientes, a correr YCSB (*framework* e cargas de trabalho comuns projetadas para avaliar o desempenho de diferentes valores-chave e serviços de armazenamento *Cloud* [49]). O YCSB fornece seis cargas de trabalho pré-configuradas (A, B, C, D, E e F), simulando diferentes cenários de aplicação. Em função dos tipos de cenários são implementadas cargas de trabalho em processos como *update, scan, insert, read, modify,* e *write*. Deste modo, através dos *logs* recolhidos podem ser obtidas estatísticas referentes às cargas de trabalho utilizadas, nomeadamente sobre o *throughput* (operações por segundo) e latências médias.

## 5.3 Análise dos Resultados

Quanto à análise dos dados obtidos durante a monitorização, podemos constatar que existiu um aumento do consumo de recursos durante a realização do teste descrito anteriormente. Analisando os recursos globais do *cluster*, podemos verificar na Figura 5.3 que durante o lançamento do HBase (entre as 11 horas e 41 minutos e as 11 horas e 56 minutos), existe um aumento da percentagem de utilização de CPU (percentagem gasta pelo utilizador atinge um máximo 10.9%) e também um aumento do número de processos (carga). No que diz respeito às estatísticas de rede, estas mantêm-se constantes, destacando-se apenas um pico no tráfego recebido (73.7 kB/s) nos primeiros minutos do teste.

[3]Os gráficos obtidos a partir da informação dos *logs* do Dstat foram gerados com recurso a uma aplicação desenvolvida para este fim, disponibilizada em `http://www.michenux.net/vmstax/`.

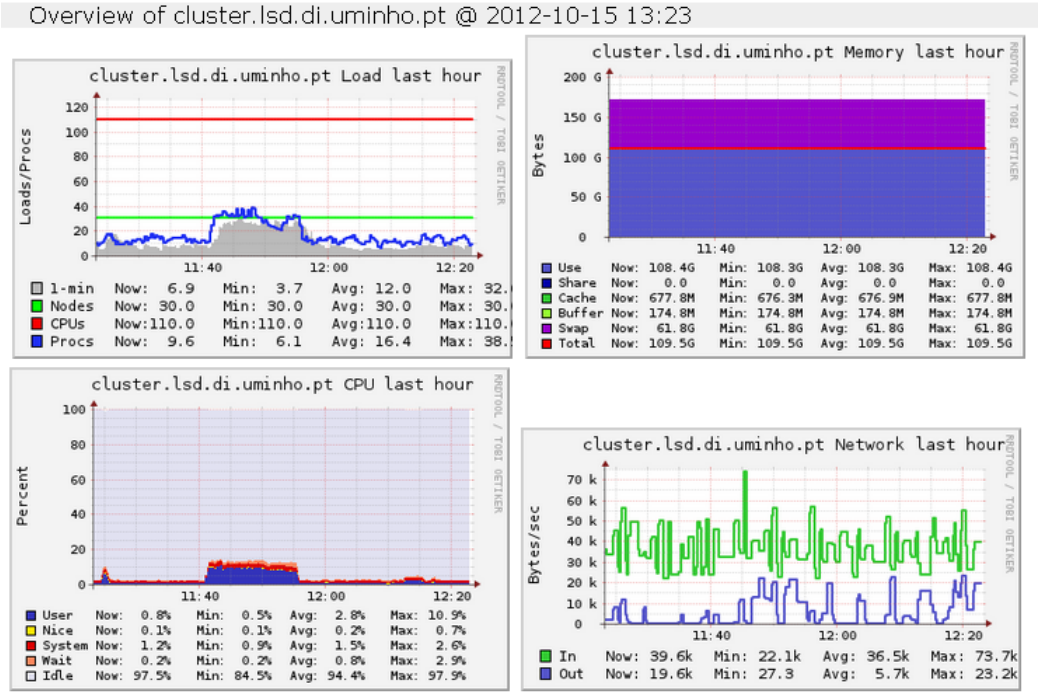



Figura 5.3: Ganglia - estatísticas do cluster.

Passando para uma análise individual das máquinas físicas e virtuais envolvidas (Tabela 5.1), é possível identificar também um aumento do consumo de recursos nas *Regions*, tal como esperado. Nesta etapa são levadas em consideração as estatísticas recolhidas pelo Dstat, uma vez que são praticamente recolhidos os mesmos tipos de informações que o Ganglia nas máquinas físicas.

Em primeiro lugar, no que diz respeito aos recursos utilizados pelo *Master*, verifica-se que ao contrário do que sucede nos recursos das *Regions*, não existe um aumento da percentagem de utilização do CPU. As percentagens de utilização de CPU mantêm-se baixas durante o teste, tanto na máquina física (ver Figura 5.4) como na máquina virtual (ver Figura 5.5). Por sua vez, em ambas as máquinas verifica-se que houve bastante atividade na escrita no disco. As estatísticas de rede revelam que existiu uma maior quantidade de dados recebidos do que enviados. No que toca aos dados relativos ao sistema, verificam-se baixos valores mas estatísticas, uma vez que não existe um grande número de processos a requerer utilização por parte do CPU.

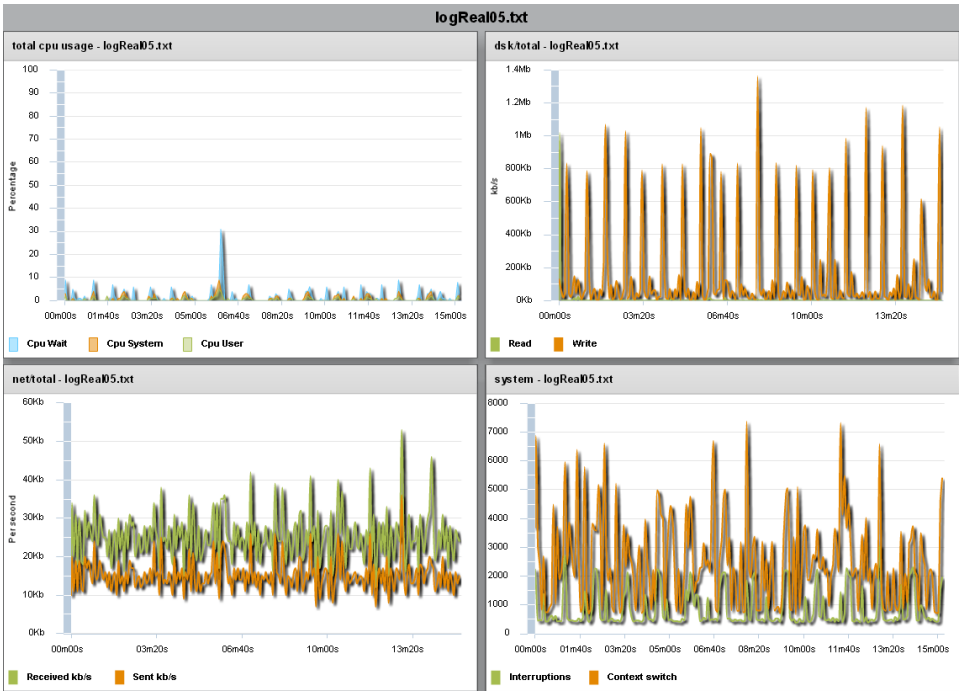



Figura 5.4: Estatísticas da Máquina Real 05.

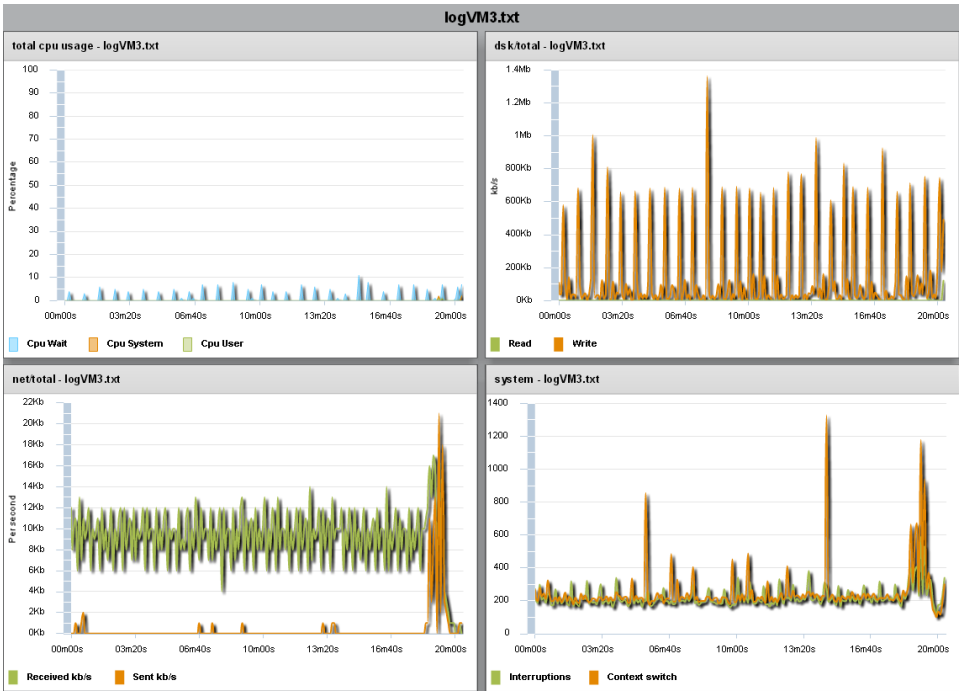



Figura 5.5: Estatísticas da Máquina Virtual 3.

No que diz respeito aos recursos utilizados pelas *Regions*, verifica-se que existiu um grande consumo de CPU durante a realização do teste. No final do teste as percentagens de utilização de CPU voltam a ser muito baixas. Tomando como exemplo as estatísticas relativas à *Region 3*[4], na máquina real (ver Figura 5.6) pode-se observar que existe uma grande percentagem de utilização do CPU por parte do utilizador. Na máquina virtual correspondente (ver Figura 5.7), existe igualmente um aumento do consumo de CPU durante o período em que foi lançado o HBase. Contudo a percentagem de utilização por parte do utilizador baixou. Em contrapartida, existe um aumento da utilização do CPU por parte do sistema e também da percentagem de CPU em espera. A causa do CPU estar em "espera" deve-se ao facto de este estar a aguardar uma resposta de um dispositivo (como memória, disco ou rede) e ainda não a recebeu. Analisando todos os gráficos dos recursos das *Regions* verifica-se que sempre que ocorre um aumento da percentagem do CPU em espera, existe um aumento da atividade nos discos relacionada com a leitura. Esta relação pode ser observada nas Figuras 5.6 e 5.7.

Quanto às estatísticas de rede, ao contrário do que sucede com os recursos utilizados pelo *Master*, nas *Regions* verifica-se que existe um aumento da quantidade de dados enviados durante o período em que decorreu o teste.

As estatísticas do sistema revelam ainda números elevados de interrupções e de *context switch* (processo computacional referente ao armazenamento e restauro do estado (contexto) do CPU, de forma a que múltiplos processos possam compartilhar uma única instância de CPU) durante a realização do teste. O elevado valor destas estatísticas indica que existe um grande número de processos a disputar a utilização do CPU, tal como se pode comprovar com as suas elevadas percentagens de utilização. É possível também observar que nas máquinas físicas existe um maior número de interrupções do que *context switch*, ao contrário das máquinas virtuais.

As estatísticas referentes ao *paging* não foram abordadas, uma vez que não se observaram dados relevantes.

[4] A *Region* 3 é escolhida de modo exemplificativo, uma vez que grande parte das características das suas estatísticas são comuns às várias *Regions*.


Figura 5.6: Estatísticas da Máquina Real 01.

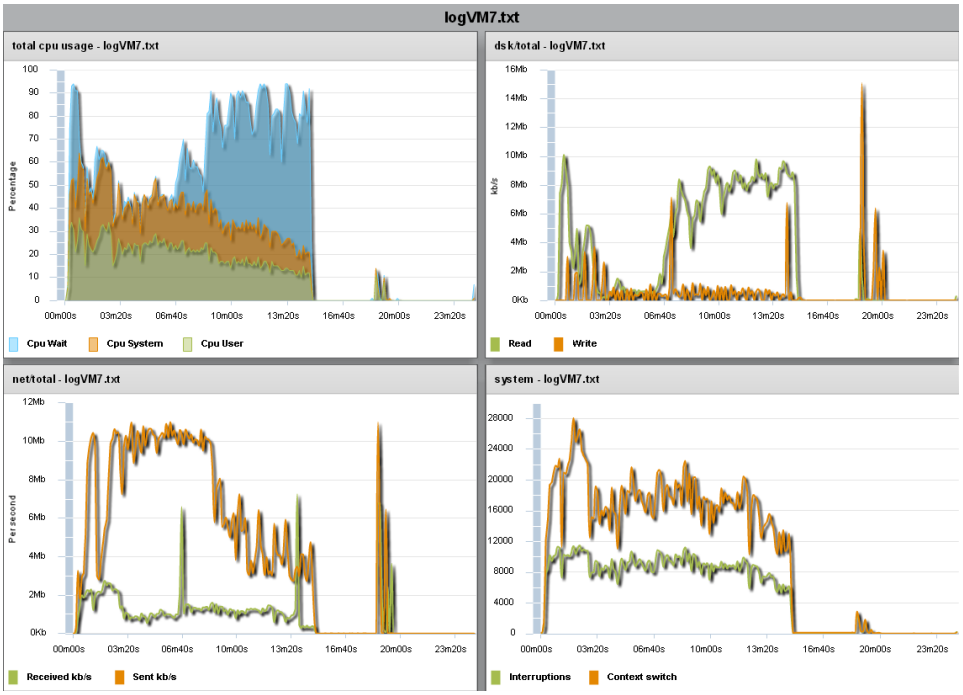



Figura 5.7: Estatísticas da Máquina Virtual 7.

Por fim, seque-se uma breve análise das estatísticas referentes às cargas de trabalho utilizadas na avaliação do HBase. Dado que este tipo de dados são úteis quando existe uma *baseline* para comparar, apenas será abordada de um modo exemplificativo a carga de trabalho ”A”. O cenário aplicativo desta carga de trabalho diz respeito ao ”armazenamento das acções recentes no contexto de uma sessão” e as proporções que utiliza são de 50% para *read* e 50% para *update*. Na Figura 5.8 estão ilustradas as estatísticas registadas no *log* gerado por esta carga de trabalho. O gráfico referente ao *throughput* ilustra as operações realizadas por segundo durante o período de teste. Por sua vez, o gráfico da latência ilustra as latências médias para os processos de *read* e de *update*. Só se começa a observar a ocorrência de latências nestes processos a partir do segundo minuto do teste, uma vez que no início ainda se estão a estabelecer ligações entre as máquinas envolvidas.

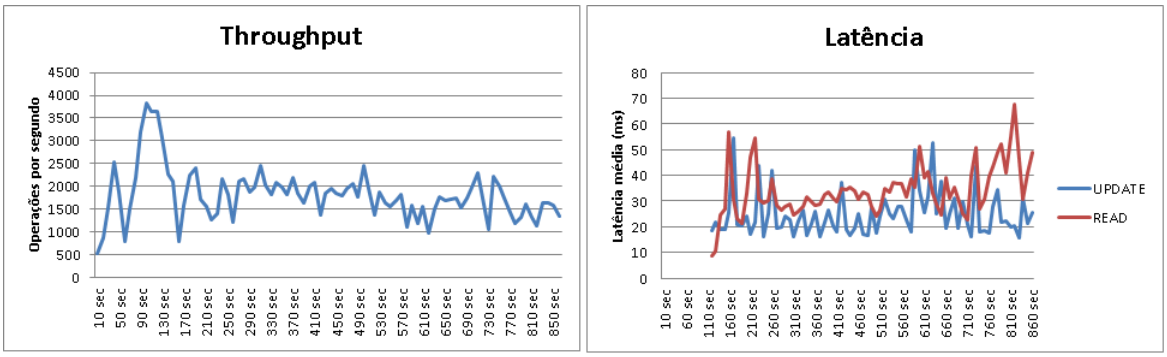



Figura 5.8: Estatísticas da carga de trabalho A

**Síntese de resultados:**

No que toca à análise dos resultados, foi possível observar que durante a execução dos testes envolvendo o HBase, houve um aumento do consumo de recursos ao nível do CPU. Esta situação ficou ilustrada tanto nas estatísticas globais do *cluster* como nas estatísticas das máquinas físicas e virtuais em específico (principalmente nos recursos das *Regions* do HBase). Existiu também durante o período de testes um aumento da atividade nos discos (leitura e escrita), assim como da quantidade de dados recebidos e enviados pelas interfaces de rede. Esta análise enquadra-se sobretudo na camada das Infraestruturas do modelo proposto no Capítulo 3. O tipo de parâmetros e métricas envolvidas dizem respeito aos recursos físicos e virtuais (categoria dos ”Componentes”), sendo possível estabelecer relações entre os mesmos. Ainda a nível dos recursos virtuais,

através da interface *web* do Openstack é possível identificar quais as instâncias ativas. A monitorização do estado das instâncias pode ajudar na identificação da fonte de possíveis problemas e respetivas causas.

Relativamente às estatísticas de rede, estas também podem ser enquadradas na camada de Rede do modelo proposto, para além da camada das Infraestruturas. Neste contexto é possível verificar que durante a realização do teste houve um aumento da troca de dados entre as várias máquinas envolvidas.

Por fim, as estatísticas relativas à carga de trabalho utilizada na avaliação do HBase estão associadas à camada Serviço/Aplicação do modelo proposto. O *throughput* e as latências médias dos processos executados pela carga de trabalho em questão enquadram-se na categoria "Outros", uma vez que são métricas específicas do tipo de serviço em questão. A existência de uma *baseline* de comparação torna-se útil na análise deste tipo de dados.

## 5.4 Sumário

Neste capítulo apresentou-se um pequeno cenário de testes, onde o objetivo foi o de explorar a monitorização de um ambiente *Cloud*, classificando as métricas obtidas segundo o modelo proposto. Em primeiro lugar apresentou-se o ambiente de testes, assim como a metodologia utilizada. Tal como referido, o processo de monitorização incidiu num dos testes em curso na *Cloud* do Departamento de Informática da Universidade do Minho. Foram utilizadas as ferramentas de monitorização Ganglia e Dstat, por sua vez abordadas no capítulo anterior. Na análise dos resultados obtidos foi possível enquadrar e classificar as métricas obtidas de acordo com o modelo de monitorização proposto. Foram identificadas métricas referentes à camada das Infraestruturas (componentes das máquinas físicas e virtuais), da camada de Rede (ainda que não muito aprofundadas) e da camada Serviço/Aplicação (estatísticas referentes aos processos utilizados na avaliação do HBase).

# Capítulo 6

# Conclusões

Neste capítulo é efetuada uma síntese do trabalho desenvolvido e apresentado ao longo desta dissertação. São apresentadas as conclusões e a análise crítica ao trabalho realizado, bem como as principais contribuições decorrentes do mesmo. Por fim, são referidos alguns possíveis tópicos para trabalho futuro.

## 6.1 Resumo do Trabalho Desenvolvido

O rápido crescimento de *Cloud Computing* como um novo modelo de prestação de serviços é um facto que não pode ser negado. De forma a se obter uma base de conhecimento que sustente a elaboração desta dissertação, inicialmente foi feito um estudo sobre a tecnologia. Um facto que se pode constatar é a falta de consenso na elaboração de uma definição de *Cloud Computing* genérica. Posto isto, foi levada em consideração uma definição resultante da extração das características essenciais consensualmente presentes em diversas definições. As principais características da tecnologia dão uma boa perceção sobre as suas mais-valias, sendo possível identificar benefícios resultantes da sua utilização na ótica dos diversos intervenientes (fornecedores, parceiros e utilizadores). Um ambiente *Cloud Computing* é bastante complexo e versátil, existindo diversos tipos de serviços, assim como várias maneiras de serem implementados e disponibilizados. Neste contexto, foram descritos os modelos de serviço, onde foram abordadas também algumas das soluções comerciais mais comuns, e os modelos de implementação existentes.

A noção da necessidade da existência de um sistema de monitorização para operar com eficiência um ambiente *Cloud* já está presente. Devido à falta de maturidade típica das novas

tecnologias, podem ser apontadas algumas limitações, nomeadamente no controlo e gestão de uma *Cloud*. Como área de pesquisa recente e ativa, na monitorização *Cloud* a falta de normas e padrões relacionados é notória. Este facto é particularmente crítico quando se tenta realizar a monitorização de serviços *Cloud* através de várias *Clouds*, envolvendo problemas de qualidade, aspetos jurídicos e de localização geográfica.

Devido à complexidade de um ambiente *Cloud*, a sua monitorização envolve diversos componentes e entidades. Com o intuito de abranger as várias dimensões envolvidas neste tipo de monitorização, o principal objetivo desta dissertação centrou-se na elaboração de um modelo estratificado de monitorização de serviços *Cloud*. O modelo proposto resultou de uma intensiva investigação das principais referências na área. Através da análise das propostas de sistemas, *frameworks* e estratégias de monitorização que procuram dar os primeiros passos e afirmarem-se nesta área, foi possível recolher alguns parâmetros e métricas a vários níveis. Ao longo deste processo de recolha de informação, o modelo de monitorização ganhou forma e ficou estratificado em 4 camadas principais. Os diversos parâmetros e métricas foram agregados por categorias segundo a sua natureza, de maneira a trazer clareza ao modelo.

Tal como foi referido, as 4 camadas principais centraram-se sobretudo nas Infraestruturas, Rede, Serviços/Aplicações e na relação Cliente/Fornecedor. As infraestruturas, ao serem a base de suporte de um ambiente *Cloud*, assumem um papel importante, uma vez que são abrangidos tanto os recursos físicos como os virtuais. Este nível de gestão abranje diversos componentes, o que por si só se traduz numa grande quantidade de fontes de dados para o processo de monitorização. Por outro lado, uma das mais-valias deste modelo a este nível, prende-se com a inclusão das questões energéticas e de segurança. Estes são dois tópicos onde tem havido bastante interesse e onde alguns trabalhos de investigação se começam a focar, fruto da sua importância. As questões energéticas focam-se sobretudo na tentativa de melhorar o rendimento e a otimização de um serviço, sem descorar as preocupações com o meio ambiente. A segurança por vezes é considerada um entrave à adoção em massa da tecnologia, uma vez que por vezes os utilizadores ficam reticentes a deslocar os seus dados para localizações desconhecidas (”nuvens”). Em relação à rede, dado que esta tecnologia é acedida através da Internet, a sua monitorização reside ao nível do serviço IP, onde já existem métricas e conceitos bem estabelecidos. Ao nível dos serviços e aplicações a monitorização deve incidir em aspetos que indiquem a sua disponibilidade, fiabilidade, eficiência, para além das questões de segurança, dado o possível caracter inseguro do meio onde são disponibilizados. Esta camada fica um pouco em aberto, uma vez que muitos dos aspetos que podem ser monitorizados dependem bastante do tipo de serviço ou aplicação em questão. Por fim a relação entre o cliente e o fornecedor de serviço também deverá ser gerida, uma vez

que estão em causa interesses económicos. Numa visão vertical do modelo proposto, visto que a relação Cliente/Fornecedor está no topo, esta pode ser associada às questões relacionadas com a QoE abordadas. Fatores como a QoS e a QoE funcionam como aspetos de diferenciação entre os diversos fornecedores.

Com o intuito de se fazer uma ponte de ligação entre o modelo de monitorização proposto e os modelos de serviços mais populares, foi estabelecida uma relação entre ambas. Este é um tópico passível de alguma discussão, na medida em que por vezes a perspetiva pode variar conforme o serviço em questão. No entanto existiu o esforço de se tentar obter uma relação o mais coerente possível.

Ao longo da análise das referências na área da monitorização *Cloud* foi possível constatar que a natureza complexa de um ambiente *Cloud* torna difícil chegar a uma solução de gestão genérica. Em grande parte, este facto deve-se aos diferentes graus de controlo e à natureza e características próprias de cada modelo de serviço (IaaS, PaaS e SaaS) e de cada modelo de implementação (*Public* e *Private*). As abordagens, mecanismos e algoritmos existentes na literatura abordam na sua maioria propósitos específicos. Uma abordagem completa por sua vez necessita de ser multi-nível, polivalente e holística, sendo assim projetada para cobrir todas as camadas dos paradigmas da computação e sistemas.

Como exemplos de estratégias de monitorização *Cloud*, foram descritos os sistemas PCMONS e Lattice. O sistema PCMONS centra-se na monitorização de *Private Clouds*, enquanto a Lattice é uma *framework* desenhada para a monitorização de recursos e serviços em ambientes virtuais. O estudo da Lattice deu um contributo inicial importante, nomeadamente na fase da estratificação do modelo proposto, devido sobretudo ao facto de identificarem os recursos físicos e virtuais, além das aplicações, como fontes de dados para a monitorização.

Ao longo do levantamento das ferramentas de monitorização que se podem enquadrar num ambiente *Cloud*, constatou-se também que a monitorização *Cloud* pode beneficiar de metodologias, conceitos e ferramentas já consolidados na gestão da computação distribuída tradicional.

Por fim, através do cenário prático abordado foi possível ter contacto com algumas das ferramentas de monitorização abordadas, assim como explorar a monitorização de um ambiente *Cloud*. Este será um tópico a explorar no futuro, de modo a se tentar demonstrar a utilidade e as vantagens do modelo estratificado de monitorização proposto.

## 6.2 Principais Contribuições

A principal contribuição do trabalho desenvolvido, cumprindo os seus objetivos, foi a de reunir, esclarecer e sistematizar as principais questões envolvidas na monitorização *Cloud* e plataformas relacionadas, a fim de fundamentar e fomentar o desenvolvimento de serviços de monitorização abrangentes e flexíveis.

A leitura desta dissertação permitirá identificar quais as principais camadas envolvidas na monitorização de serviços *Cloud*, assim como os respetivos parâmetros e métricas mais relevantes. Espera-se que através de uma visão estratificada do processo de monitorização dos complexos ambientes *Cloud Computing*, seja possível efetuar uma gestão mais eficiente dos mesmos. Para além disso, esta dissertação fornece uma lista de plataformas e ferramentas de monitorização de *Cloud Computing* existentes no mercado.

A monitorização *Cloud* é uma área de pesquisa recente e ativa, onde a falta de padrões relacionados é evidente. Outra das contribuições do trabalho desenvolvido enquadra-se nos esforços dedicados à normalização, propondo uma abordagem estratificada, identificando e sugerindo parâmetros, métricas e boas práticas para uma monitorização eficiente dos serviços e ambientes *Cloud Computing*.

Como fruto deste trabalho, foram elaborados e submetidos dois artigos científicos na "12$^a$ Conferência de Redes de Computadores" [4], já aceite para publicação, e na "IEEE ICC 2013 - Next-Generation Networking Symposium" [5], encontrando-se o mesmo em processo de revisão.

## 6.3 Trabalho Futuro

O trabalho futuro inclui validar e ajustar o modelo proposto recorrendo a cenários experimentais mais diversificados. Neste contexto espera-se uma evolução do cenário de testes de modo a serem testados outros tipos de serviços *Cloud* mais comuns. Espera-se também explorar vários cenários de carga, de maneira a comprovar a utilidade do modelo estratificado de monitorização proposto. Através da análise dos resultados obtidos espera-se estabelecer relações causa/efeito com maior clareza e eficiência.

Uma vez que a monitorização *Cloud* é um tópico em constante atualização, fruto dos vários esforços científicos e empresariais, o modelo de monitorização proposto pode ainda ser complementado com novas perspetivas e parâmetros que possam surgir.

# Bibliografia


[1] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *ELSEVIER, Future Generation Computer Systems*, 25:599 – 616, June 2009.

[2] Nick Antonopoulos and Lee Gillam. *Cloud Computing: Principles, Systems and Applications*. Springer, 2010.

[3] Yuanhui Sun, Zongshui Xiao, Dongmei Bao, and Jie Zhao. An architecture model of management and monitoring on cloud services resources. In *3rd Internacional Conference on Advanced Computer Theory and Engineering (ICACTE)*, pages 207–211, 2010.

[4] $12^a$ Conferência de Redes de Computadores. Novembro 2012. `http://crc2012.av.it.pt/`.

[5] IEEE ICC 2013 Next-Generation Networking Symposium. June 2013. `http://www.ieee-icc.org/2013/index.html`.

[6] Luis Vaquero, Luis Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev*, 39(1):50–55, January 2009.

[7] ITU-T FG Cloud. Part 7: Cloud computing benefits from telecommunication and ICT perspectives. Technical report, February 2012.

[8] NIST Working Definition Of Cloud Computing. `http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html`.

[9] ITU-T FG Cloud. Part 1: Introduction to the cloud ecosystem: definitions, taxonomies, use cases and high-level requirements. Technical report, February 2012.

[10] Cloud Computing Use Case Discussion Group. Cloud computing use cases white paper v4.0. Technical report, July 2010.

[11] Eric A. Marks and Bob Lozano. *Executive's Guide to Cloud Computing*. WILEY, 2010.

[12] Amazon. `http://aws.amazon.com`.

[13] Google. App. engine. `http://code.google.com/intl/pt-BR/appengine/`.

[14] Microsoft. Office365. `http://www.microsoftbusinesshub.com/Products/Microsoft_Office_365`.

[15] Openstack. `http://www.openstack.org`.

[16] Augusto Ciuffoletti. Monitoring a virtual network infrastructure, an IaaS perspective. *ACM SIGCOMM Computer Communication Review*, 40(5):47–52, October 2010.

[17] Shirlei Chaves, Rafael Uriarte, and Carlos Westphall. Toward an architecture for monitoring private clouds. *IEEE Communications Magazine*, 49(12):130–137, December 2011.

[18] Solange Lima and Paulo Carvalho. *Redes Multiserviço*. Departamento de Informática - Universidade do Minho, 2011.

[19] Stuart Clayman, Alex Galis, Clovis Chapman, Giovanni Toffetti, Luis Merino, Luis Vaquero, Kenneth Nagin, and Benny Rochwerger. Monitoring service clouds in the future Internet. pages 115–126. IOS Press, April 2010.

[20] Mohamed Mohamed, Djamel Belaïd, and Samir Tata. How to provide monitoring facilities to services when they are deployed in the cloud? In *CLOSER 2012 - 2nd International Conference on Cloud Computing and Services Science*, page 258 – 263, 2012.

[21] Aiiad Albeshri, Colin Boyd, and Juan Nieto. GeoProof: Proofs of geographic location for cloud computing environment. In *ICDCS 2012 - 32nd International Conference on Distributed Computing Systems*, pages 506–514, 2012.

[22] Tobias Hoßfeld, Raimund Schatz, Martin Varela, and Christian Timmerer. Challenges of QoE management for cloud applications. *IEEE Communications Magazine*, 50(4):28 – 36, April 2012.

[23] Toni Mastelic, Vincent C. Emeakaroha, Michael Maurer, and Ivona Brandic. M4CLOUD - generic application level monitoring for resource-shared cloud environments. In *CLOSER 2012 - 2nd International Conference on Cloud Computing and Services Science*, page 522 – 532, 2012.

[24] Jorge Werner, Guilherme Geronimo, Carlos Westphall, Fernando Koch, and Rafael Freitas. Simulator improvements to validate the green cloud computing approach. In *Network Operations and Management Symposium (LANOMS) 7th Latin American*, October 2011.

[25] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy aware resource allocation heuristics for efficient management of data centers for cloud computing. *ELSEVIER, Future Generation Computer Systems*, 28:755–768, May 2012.

[26] Jonathan Spring. Monitoring cloud computing by layer, part 1. *IEEE Security & Privacy Magazine*, 9(2):66–68, March/April 2011.

[27] Jonathan Spring. Monitoring cloud computing by layer, part 2. *IEEE Security & Privacy Magazine*, 9(3):52–55, May/June 2011.

[28] Ya-Shiang Peng and Yen-Cheng Chen. SNMP-based monitoring of heterogeneous virtual infrastructure in clouds. In *Network Operations and Management Symposium (APNOMS) 13th Asia-Pacific*, September 2011.

[29] Taesang Choi, Nodir Kodirov, Tae-Ho Lee, Doyeon Kim, and Jaegi Lee. Autonomic management framework for cloud-based virtual networks. In *Network Operations and Management Symposium (APNOMS) 13th Asia-Pacific*, September 2011.

[30] Mehdi Sheikhalishahi and Lucio Grandinetti. Revising resource management and scheduling systems. In *CLOSER 2012 - 2nd International Conference on Cloud Computing and Services Science*, page 121 – 126, 2012.

[31] Kleber Vieira, Alexandre Schulter, Carlos Westphall, and Carla Westphall. Intrusion detection for grid and cloud computing. *IT Professional, IEEE Computer Society*, pages 38–43, July/August 2010.

[32] Shirlei A. Chaves, Carlos B. Westphall, and Flavio R. Lamin. SLA perspective in security management for cloud computing. In *Sixth International Conference on Networking and Services, IEEE Computer Society*, pages 212–217, March 2010.

[33] Flávio R. C. Sousa, Leonardo O. Moreira, Gustavo A. C. Santos, and Javam C. Machado. Quality of service for database in the cloud. In *CLOSER 2012 - 2nd International Conference on Cloud Computing and Services Science*, page 595 – 601, 2012.

[34] Pankesh Patel, Ajith Ranabahu, and Amit Sheth. Service level agreement in cloud computing. Technical report, September 2009.

[35] Solange Lima. *A Distributed Admission Control Model for Class-based Networks*. PhD thesis, University of Minho, Braga, 2005.

[36] Srikanth Sundaresan, Walter Donato, Nick Feamster, Renata Teixeira, Sam Crawford, and Antonio Pescapè. Broadband Internet performance: a view from the gateway. SIGCOMM '11, pages 134–145. ACM, 2011.

[37] Katerina Stamou, Jean-Henry Morin, Benjamin Gateau, and Jocelyn Aubert. Service level agreements as a service - towards security risks aware SLA management. In *CLOSER 2012 - 2nd International Conference on Cloud Computing and Services Science*, page 663 – 669, 2012.

[38] Massimiliano Rak, Salvatore Venticinque, Tamás Máhr, Gorka Echevarria, and Gorka Esnal. Cloud application monitoring: the mOSAIC approach. In *Third IEEE International Conference on Cloud Computing Technology and Science, IEEE Computer Society*, pages 758–763, 2011.

[39] Sysstat. `http://sebastien.godard.pagesperso-orange.fr/features.html`.

[40] Dstat. `http://dag.wieers.com/home-made/dstat/`.

[41] Nagios. `http://www.nagios.org/`.

[42] Ganglia. `http://ganglia.sourceforge.net/`.

[43] IBM Ganglia HowTo. `http://www.ibm.com/developerworks/wikis/display/WikiPtype/ganglia`.

[44] Groundwork. `http://www.gwos.com/`.

[45] Rightscale. `http://www.rightscale.com/`.

[46] Landscape. http://www.canonical.com/enterprise-services/ubuntu-advantage/landscape.

[47] Kaavo. http://www.kaavo.com/.

[48] HBase. http://hbase.apache.org/.

[49] YCSD. https://github.com/brianfrankcooper/YCSB/wiki.

[50] Zookeeper. http://zookeeper.apache.org/.