

Lógica *fuzzy* aplicada à gerência de distribuição de carga em um *grid* de computadores

Gabriel Garcia do Nascimento¹

Luciana Alvim Santos Romani²

Os avanços tecnológicos têm levado a um aumento no volume de dados científicos disponíveis, seja por meio da coleta feita por diferentes sensores, como por modelos matemáticos que geram inúmeras saídas distintas. Especialmente nas pesquisas em mudanças climáticas, os especialistas têm que lidar com esse grande volume de dados, o que extrapola a capacidade do ser humano de cálculo e análise. Desta forma, o apoio computacional se torna fundamental para alavancar as pesquisas em diferentes áreas.

Em busca de uma solução economicamente viável e ecologicamente correta para este problema, e que ainda pudesse ser compartilhada por diversos projetos na empresa, a equipe do projeto SCAF adotou a computação em grade. Dessa forma, um *grid* computacional (PISKE, 2004), formado por um núcleo de máquinas dedicadas e por máquinas virtuais que aproveitam o tempo ocioso das demais máquinas da empresa, foi projetado e configurado. Essa solução foi adotada, uma vez que esse tipo de computação paralela é não homogêneo, ou seja, os computadores que integram a rede não precisam ser idênticos, o que facilita a integração de novas máquinas na rede.

Uma vez em operação, percebeu-se que se alguma máquina do *grid* parasse de funcionar ou apresentasse problema na execução dos processos, esses problemas só eram detectados tardiamente. Embora o *grid* esteja sendo usado há mais de 6 meses, o monitoramento das máquinas ainda

¹ PUC-Campinas, gabrieln@outlook.com

² Embrapa Informática Agropecuária, luciana.romani@embrapa.br

é feito manualmente, o que dificulta bastante o trabalho do administrador do sistema. Ele precisa acessar constantemente o sistema para assegurar que ele continua no ar e com todas as máquinas em funcionamento. A solução proposta para tornar mais eficiente o gerenciamento do *grid* foi o desenvolvimento de um sistema de monitoramento. Neste contexto, o objetivo deste trabalho foi desenvolver um sistema para monitorar, em tempo real, a atividade e as informações de estado do *grid* de computadores que vem sendo utilizado pela equipe do projeto SCAF, a fim de garantir a execução das aplicações identificando antecipadamente a interrupção ou queda das máquinas que compõem o *grid* computacional.

A metodologia definida para o monitoramento do sistema de *grid* baseia-se no modelo usado pelo sistema Ganglia (MASSIEA, 2004) que segue um padrão centralizado. O sistema de monitoramento de *grid* computacional (SMGD) foi desenvolvido em java, usando pacotes do sistema Unix para obter informações do hardware das máquinas do *grid*. A arquitetura do sistema SMGD pode ser vista na Figura 1, e segue um padrão centralizado. O sistema colhe informações sobre o uso da CPU, memória, disco e a temperatura de cada uma das máquinas ativas do *grid* e as armazena

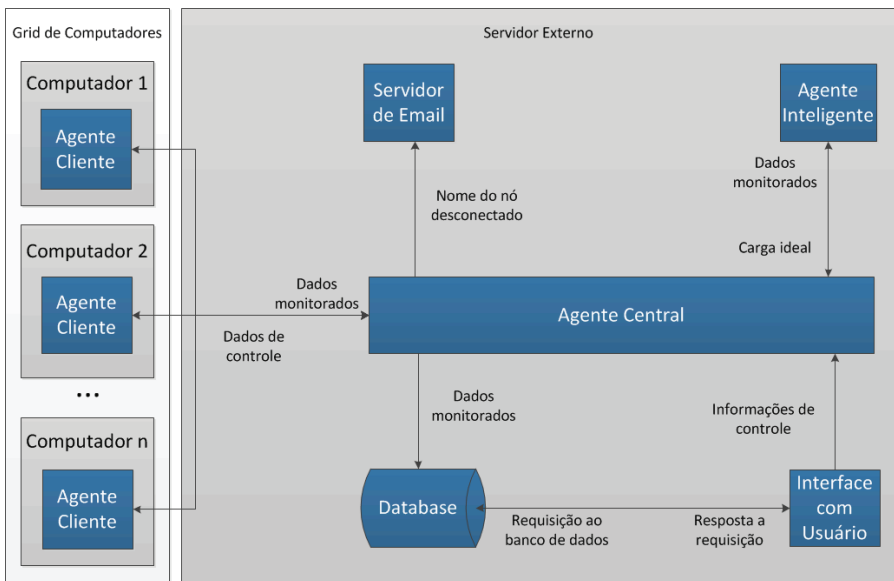


Figura 1. Arquitetura do Sistema de Monitoramento de computadores.

em um servidor central para análise posterior. Para isso foi necessário o desenvolvimento de um Agente Central que fica instalado no servidor, e um Agente Cliente que fica instalado em cada máquina e é responsável por coletar os dados de monitoramento em intervalos de tempo pré-determinados, enviando-os ao Agente Central. Esse agente recebe as informações de monitoramento de cada Cliente e armazena-as. Ele é sensível à atividade dos Clientes, reportando por meio de e-mail e/ou da interface gráfica do sistema, o status de cada máquina, como por exemplo, se uma máquina não está mais em funcionamento (possível queda).

A definição do número ótimo de processos a serem executados em uma determinada máquina do grid é feita por meio de um algoritmo baseado em lógica nebulosa (*Fuzzy*). O algoritmo de lógica *fuzzy* utiliza dados de temperatura, uso da CPU, memória e número de processos de um computador em um determinado momento, para definir qual seria o número de processos ideal para aquela máquina. Para isso, o algoritmo precisa primeiramente transformar essas informações de dados escalares para variáveis difusas. Com isso, o algoritmo define se uma determinada temperatura é baixa, excelente ou ótima, se o uso da CPU é pouco, bom ou ótimo, e se a memória está cheia ou vazia. Após esse processo o algoritmo transforma essas informações em um dado de saída chamado 'processos', ou seja, conforme as regras de produção do algoritmo, ele faz o processo inverso, transformando as variáveis em dados escalares para definir a quantidade de processos ideal para a máquina. As regras de produção definem 3 possíveis estados, o primeiro é um caso neutro, ou seja, não há alterações no número de processos, o que acontece quando o computador está trabalhando da melhor forma possível, isto é, com memória cheia, carga alta na CPU e com a temperatura baixa ou excelente. O segundo caso ocorre quando há a necessidade de retirada de processos, o que acontece somente quando a temperatura está alta demais. Por último, a adição no número de processos que ocorre quando a CPU está ociosa (pouca carga) e existe memória livre para ser utilizada.

A interface gráfica do sistema foi desenvolvida para web, o que possibilita uma visualização mais intuitiva do que o acesso por linha de comando sendo passível de uso por usuários sem conhecimentos aprofundados de computação. O uso de javascript e AJAX na interface possibilitou a criação de um ambiente dinâmico. As informações coletadas dos computadores podem ser exibidas em gráficos ou tabelas de dados. Essa forma de apre-

sentação dá uma visão geral do funcionamento de cada máquina, como pode ser visto na tela de atividade do *grid* na Figura 2, onde as máquinas em verde estão ativas, em vermelho estão em queda, e em cinza estão inativas.

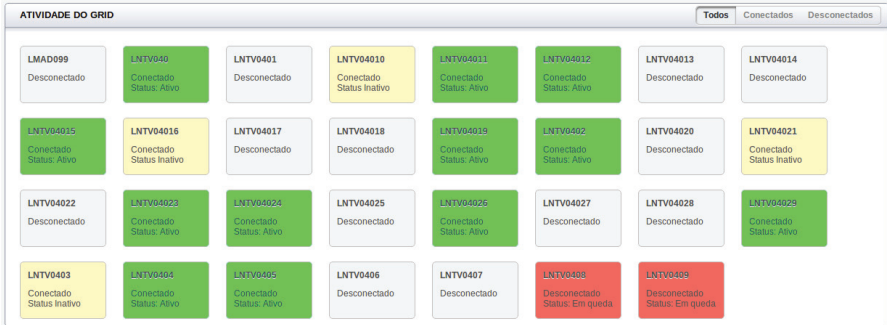


Figura 2. Tela de status das máquinas do *grid*.

Testes iniciais mostraram que o monitoramento por meio do sistema reduz a carga de trabalho do administrador e antecipa informações sobre o funcionamento das máquinas. Embora existam outros sistemas de monitoramento de *grids*, o desenvolvimento do SMGD contribui para as pesquisas em computação de alto desempenho usando lógica nebulosa para definição do status do sistema. Em trabalhos futuros serão realizados testes no sistema incorporando novas máquinas reais do *grid*, bem como um conjunto de máquinas virtuais.

Referências

PISKE, O. R. **Computação em grid Centro Universitário Positivo**. 2004. Disponível em: <http://www.angusyoung.org/arquivos/artigos/grid_computing.pdf>. Acesso em: 19 abr. 2012.

MASSIEA, M. L.; CHUN, B. N.; CULLER, D. E. The ganglia distributed monitoring system: design, implementation, and experience. **Parallel Computing**, v. 30, p. 817-840, 2004.