A Simulation Tool for Evaluating the Constraint-Based Allocation of Storage Resources for Multi-Tenant Cloud Applications

Pieter-Jan Maenhaut*†, Hendrik Moens†, Bruno Volckaert†, Veerle Ongenae* and Filip De Turck†
*Ghent University, Faculty of Engineering and Architecture, Dept. of Information Technology
Valentin Vaerwyckweg 1, 9000 Ghent, Belgium
†iMinds – INTEC, Ghent University, Dept. of Information Technology
Gaston Crommenlaan 8 bus 201, 9050 Ghent, Belgium
Email: pieterjan.maenhaut@intec.ugent.be

Abstract—Cloud computing is closely related to multi-tenancy, as it relies on resources that are shared among multiple clients. The provisioning and management of storage resources for cloud applications is an interesting research topic, as reallocation of data over time should be minimised, and the developed strategy should guarantee both data separation and performance isolation for every tenant. In this demo, we present a simulation tool for evaluating and comparing different data allocation strategies. Evaluation using real implementations can be very expensive and time consuming, and is not always possible, due to the scale and complexity of the infrastructure on which they are intended to run. The simulator aids as a tool for inexpensive and rapid evaluation of new techniques, and to validate and finetune new data allocation strategies.

I. INTRODUCTION

Multi-tenancy enables the sharing of a common set of resources among multiple client organisations, referred to as tenants. Cloud computing itself is a form of multi-tenancy as it relies on virtualization, and thus physical hardware resources are being shared among multiple clients. A typical multi-tenant application hosted on the cloud consists of two types of resources, computational resources, used to execute the application logic, and storage resources, used to store the application (tenant) data. The provisioning and management of storage resources is an interesting research topic, as reallocation of data over time should be minimised. This is especially true for heterogenous environments where application servers are geographically distributed, making migration of data both expensive and time consuming. Furthermore, although resources are being shared among multiple tenants, the storage system should behave like a private instance towards every tenant by guaranteeing both data separation and performance isolation.

In this demo, we present a simulation tool for evaluating different data allocation strategies implemented in a Software-Defined Storage (SDS) [1] system responsible for the management of storage resources in multi-tenant cloud environments. These algorithms are difficult to evaluate in real-life deployments due to the scale and complexity of the infrastructure

on which they are intended to run, making such evaluations both expensive and time consuming. The proposed simulator allows for rapid and inexpensive evaluation of different data allocation techniques. Once a new strategy has been validated and evaluated using the tool, it can be implemented in a (small-scale) real-life infrastructure for further experimentation.

II. HIERARCHICAL RESOURCE PROVISIONING

In previous work [2], we proposed the design of a scalable multi-tenant SDS system, responsible for the dynamic (re-)allocation of tenant data over the available storage resources. The system invokes the Hierarchical Bin Packing (HBP) algorithm, introduced in the paper, to determine a feasible allocation of tenant data. In order to evaluate the system and the proposed algorithms, the tool presented in this demo was built for rapid simulations based on existing datasets.

In the presented approach, tenants can have multiple scenario-specific attributes assigned, such as the selected Service-Level Agreement (SLA), the geographical location, average usage, and so on. This enables the development of a flexible system suitable for many heterogeneous scenarios. Based on these attributes, the tenants are hierarchically clustered and mapped to storage resources. This leads to a very different approach for the allocation of tenant data, with a strong focus on data isolation, and support for custom constraints and requirements from the tenant's perspective.

The simulation tool presented in this demo not only allows for evaluation of the system and the HBP algorithm, but it also allows for quick comparison with existing methods and algorithms such as the First-Fit Bin Packing (FFBP) algorithm and the CRUSH [3] algorithm, used by Ceph [4].

III. DEMO SETUP

Figure 1 provides an overview of the simulator, and illustrates the interaction between the different components. The main component portrayed here is the storage elasticity component, which is a binary executable invoking the selected strategy. The allocation strategies are designed as pluggable
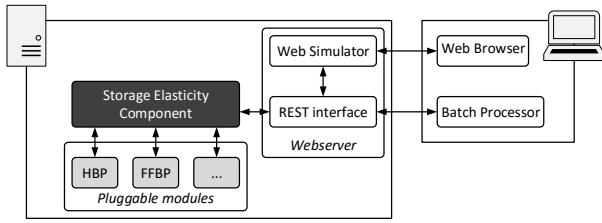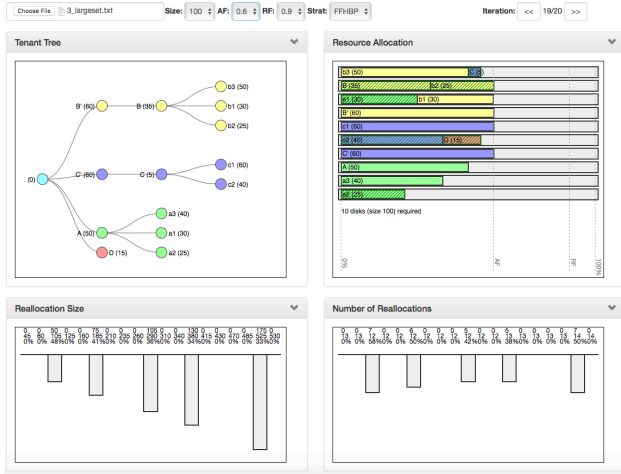
Fig. 1. Overview of the simulator setup.

Fig. 2. Screenshot of the web GUI of the simulator.

modules, making it easy to add, remove or update the algorithms to be evaluated. As the front-end of the simulator is mainly developed using HTML5[1] and JavaScript with jQuery[2], a REST interface was designed and implemented using PHP5[3] to allow communication with the Storage Elasticity Component. The REST interface and static HTML are running on top of a Apache HTTP Server[4].

There are two ways to interact with the simulator, depending on the type of experiment. First of all, there is a web GUI, illustrated in Figure 2, allowing the user to browse through the different iterations of an experiment, and presenting the different metrics in a user-friendly way. The web GUI is especially useful for smaller experiments or to gain insight in the developed algorithms. Next, there is a batch processor, useful for larger experiments (larger amount of tenants, higher number of iterations and larger execution times). The batch processor produces dataset files which can be easily imported in plotting tools such as Gnuplot.

Multiple configuration parameters can be easily adjusted in the simulation tool, such as the preferred allocation factor, denoting the maximum allowed percentage of a single storage bin that can be assigned by the system during (re-)allocation.

The simulator currently computes the following metrics:

- *Average Utilization*: measures the average utilization of the provisioned storage resources. Our research goal is to optimize the algorithm to reach an average utilization close to the selected allocation factor.
- *Number of Reallocations*: the simulator calculates and differentiates between the number of reallocations within a single resource pool (inexpensive reallocations) and between different resource pools (expensive reallocations).
- *Reallocation Size*: similar to the previous metric, this metric calculates the total reallocation size.
- *Average Distance*: calculates the average distance of the nodes (tenants) allocated to the same bin. A lower value for this metric indicates better clustering of tenants.

Figure 3 illustrates the output of the simulator for the average utilization metric for a single experiment over multiple iterations.
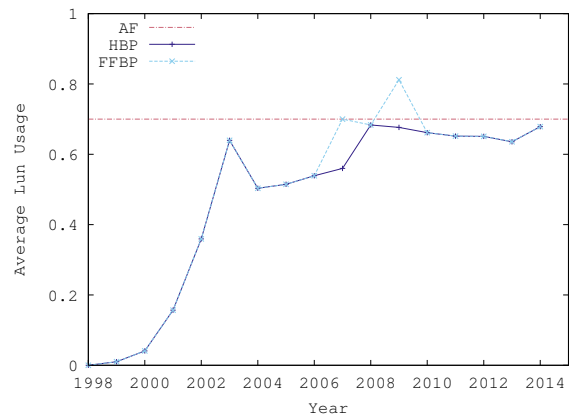

Fig. 3. Illustration of the average utilization for a fast-growing dataset over multiple iterations using two different data allocation strategies.

REFERENCES

[1] M. Carlson, A. Yoder, L. Schoeb, D. Deel, and C. Prattr, "Software defined storage," SNIA, Tech. Rep., Mar. 2014. [Online]. Available: http://www.snia.org/sds
[2] P.-J. Maenhaut, H. Moens, B. Volckaert, V. Ongenae, and F. D. Turck, "Design of a hierarchical software-defined storage system for data-intensive multi-tenant cloud applications," in *Proceedings of the 11th International Conference on Network and Service Management 2015 (CNSM2015)*, Barcelona, Spain, Nov. 2015.
[3] S. A. Weil, S. A. Brandt, E. L. Miller, and C. Maltzahn, "Crush: Controlled, scalable, decentralized placement of replicated data," in *Proceedings of the 2006 ACM/IEEE Conference on SuperComputing (SC '06)*, 2006.
[4] S. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *Proceedings of the 7th Conference on Operating Systems Design and Implementation (OSDI '06)*, Nov. 2006.
[5] (2015) DeCoMAdS: Deployment and Configuration Middleware for Adaptive Software-as-a-Service of Software Services in the cloud. [Online]. Available: https://distrinet.cs.kuleuven.be/research/projects/DeCoMAdS

---

[1]HTML5 - http://www.w3.org/TR/html5/

[2]jQuery - https://jquery.com

[3]PHP: Hypertext Preprocessor - https://www.php.net

[4]The Apache HTTP server project - https://httpd.apache.org