Real-Time Multi-Cloud Management Needs Application Awareness

John Chinneck
Carleton University
Ottawa, Canada
+1-613-520-5733
chinneck@sce.carleton.ca

Marin Litoiu
York University
Toronto ,Canada
1-416-485-4003
mlitoiu@yorku.ca

Murray Woodside
Carleton University
Ottawa, Canada
1-613-520-5721
cmw@sce.carleton.ca

ABSTRACT
Current cloud management systems have limited awareness of the user application, and application managers have no awareness of the state of the cloud. For applications with strong real-time requirements, distributed across new multi-cloud environments, this lack of awareness hampers response-time assurance, efficient deployment and rapid adaptation to changing workloads. This paper considers what forms this awareness may take, how it can be exploited in managing the applications and the clouds, and how it can influence cloud architecture.

Categories and Subject Descriptors
D.4.8 [**Performance**]: Measurements, Modeling and Prediction, Queuing Theory

Keywords
Cloud management; optimization; performance models; layered queueing.

1. INTRODUCTION
Multi-clouds comprise several geographically dispersed clusters with possibly separate management. An example is the SAVI multi-tier cloud [14], developed by a group of universities and companies in Canada. It consists of a core cloud filling a role similar to current clouds, with substantial resources, and small edge clouds, distributed geographically and integrated with the network elements such as routers. By minimizing the distance between the end user and the computing elements, low latency and high bandwidth applications such as real time or multimedia can achieve their minimum levels of quality of service. At the same time, the core cloud supports other requirements of the application, such as a high volume of computations and storage. Similar concerns apply to other multi-cloud architectures, such as hybrid clouds.

The common approach to resource management in IaaS clouds is for each application manager to determine its needs in terms of VMs and request them from the cloud, while the cloud manager determines their placement. However, to take advantage of the multi-cloud architecture, an application should be aware of cloud topology and resources. On the other hand, to fulfill applications requirements and to achieve its own objectives, a cloud infrastructure should be aware of the application objectives and the time-relationships of its components.

ICPE'14, March 22 - 26 2014, Dublin, Ireland

http://dx.doi.org/10.1145/2568088.2576763

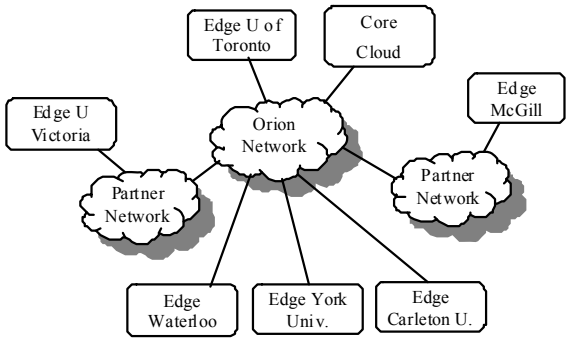

Figure 1: SAVI Edge and Core Clouds

We are motivated by mobile interactive applications in the cloud with a strong requirement for fast and time coordinated user responses, and heavy data streams in both directions. Examples include multiplayer games on mobile devices, image- handling interactive applications, flash crowds, etc. . The edge cloud can make these applications more responsive, while the core cloud to handles globally shared data and carries out data-intensive computations.

Awareness is Essential: For these applications running on multi-clouds, application awareness is not optional. It is essential for obtaining adequate responsiveness and efficient adaptation. Fundamentally, balancing the deployment of parts of the application over two sub-clouds may introduce unacceptable delays due to communications between the parts; the manager that decides the deployment must be aware of the communication delays, which are a combination of application properties (internal communications patterns) and cloud properties (such as the communication delay between sub-clouds).

2. MANAGEMENT ARCHITECTURES AND AWARENESS
This "application-awareness" applies in both directions, and can take several forms as discussed in Section 3 below. Consider a set of applications deployed over a set of clouds or sub-clouds, with representative members illustrated in Figure 2.

Figure 2(A) shows collaboration between separate managers for If each application and each cloud. Awareness data can flow between them as indicated by

- AiAD to represent "Application i Awareness Data"
- CjAD to represents "Cloud j Awareness Data".

Figure 2(B) shows a management architecture in which the application manager determines it awareness data but delegates the immediate adaptive decision-making to the cloud.

Figure 2(C) shows an all-knowing global manager which makes decisions for all applications over all clouds. It is generally true that the availability of more information allows better management of any system.

There are three potential gains from application awareness:

- More efficient deployment.
- QoS-based decisions.
- Adaptive deployment across multiple clouds, in order to provide lower user-to-cloud latency.

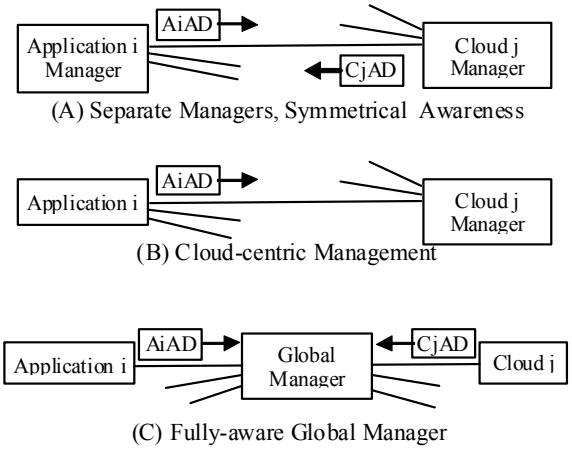

Figure 2 Management Architectures to exploit Awareness

Actual clouds use separate managers as in (A), with minimal mutual awareness. The cloud publishes its available resources, and the application tells it which ones it wishes to reserve. To adapt, the application tracks its status and changes its resource reservations over time, and the cloud executes these changes and also may redeploy VMs to consolidate workloads.

The simultaneous optimization of multiple applications is an ideal problem for large-scale techniques such as mixed-integer programming. The optimization may reflect the goals of the application and cloud managers by minimizing the number of hosts, their operating costs or energy costs, or some combination of these balanced against the cost of poor application response times. The constraints on the deployment can include the CPU and memory of each host (required for deployed VMs), the QoS guarantees or (more indirectly) the resource utilizations that result from the deployment, the number of replicas of certain tasks (due to license availability or cost), and the number of VMs per host.

Global decisions and large-scale optimization methods can provide a benchmark for simpler, more practical techniques. Taking advantage of full awareness, they can determine the best possible decisions. They can also be used to study the impact on the applications and the cloud of different optimization goals, for example does an energy-efficiency goal impact the applications compared to a total cost-saving goal?

2.1 Effectiveness of Different management Architectures

Separate Managers

This is the prevalent approach.

Most studies do not use an exchange of awareness data, but some of them at least use performance awareness within the application level management. Cardellini et al [1] considered formal optimization by the application manager of its requests for VM reservations. Wang et al. use a performance model within the application manager to determine what VMs to request and for what period.

Wu et al [11] propose a cloud manager with complex heuristics based on credit levels and future plans communicated from the application manager, which is a form of partial awareness.

Separate Managers and exchange of awareness data seems to be a future research topic, as discussed below.

Global Manager

The authors have experimented with a global manager for multiple applications running on a single cloud, with large-scale optimization techniques. Application awareness was provided by a detailed performance model which adapts to the application status [5][6][7][8]. Global optimization was compared to optimization of each application separately, and to using a simple rule that corresponds to separate management with low awareness.

Some challenges to using global optimization were evaluated: the model had to be extended to accommodate different kinds of constraints, such as license limits, and a two-stage approach combining bin-packing heuristics and mixed-integer programming (MIP)was needed for scalability. Scalability was still limited by the MIP solvers.

Other authors have also considered versions of full application awareness with different models For example, Ghanbari et al [4] optimize a utility function combining an application-level SLAs and resource costs with tunable parameters for the administrator to specify trade-offs between the two.

Cloud-centric Management

Van et al [9] describe experience with Cloud-centric architecture, for a single cloud. They maximized a global utility function combining the utility of each application with that of the cloud provider. Applications provided the cloud manager with a function to evaluate the effects of a deployment, or an adaptive change, on utility, so the cloud manager can evaluate trade-offs between applications.

Zhang et al [10] define a multi-cloud manager with minimal application awareness, and consider game-theoretic strategies to resolve resource conflicts between applications. They do not consider responsiveness in deriving the deployments.

For practical management the most promising way forward seems to be to improve the separate managers (architecture A) with greater awareness, particularly to ensure that response times can be guaranteed for a multi-cloud deployment. Awareness can take many forms.

3. FORMS OF AWARENESS

We use the term *awareness* for the information exported about an application or a cloud, to describe itself and its current state. It can take different forms, which impact how useful it is for management.

Awareness of the Application

Some examples of the awareness exported by an application to a cloud, in increasing order of complexity, are:

1. **VMs:** the number and the type of VMs required for the application (minimal awareness)
2. **Aggregate host demand**: The total CPU demand of the application, in CPU-sec/sec, that are to be provided (this is essentially the number of cores required, without providing any margin), plus the total network traffic generated say in MB/second (low application awareness).
3. **VM demands**: The list of VMs to be deployed (possibly in multiple replicas), with the CPU demand of each, and the network traffic between pairs of VMs. This determines the total cores required for each VM, and indicates the traffic pattern between them. It can be used to make decisions about the "size" of VMs. (partial application awareness).
4. **App-Opt Properties**: optimization-related properties indicating the value to the application users and application manager of additional resources, derived from a local optimization, combined with willingness to pay for them. Application details are not revealed. (partial application awareness).
5. **Predictive Awareness**: An performance model which predicts the response time as a function of the deployment(full application awareness).

Awareness can be regarded as a kind of *model* of the application. The first model above corresponds to what is provided by current cloud users in their requests for services. The second would allow the cloud manager to determine the number and size of the VMs. The third conveys additional but partial information that could be used by the cloud provider to determine the VM size and host to which each task should be deployed, taking into account the internal cloud communications structure for inter-VM capacity. At present the application manager deploys tasks to VMs without knowledge of the host location. The fourth model could relate to a decentralized optimization strategy where the awareness coordinates the separate decision-making optimizers. The fifth is essentially a performance model (more or less detailed) that could be used by the cloud manager to find deployments that satisfy application QoS requirements such as user-perceived latencies. Detailed predictive awareness is provided in our research by a Layered Queueing Network (LQN) model, which is illustrated in Figure 3. The blocks represent application deployable units (tasks), with sub-rectangles representing the operations performed by the tasks, with their CPU demands. Calls (service requests) between operations are indicated by arrows annotated by the number of calls per operation and the mean data transfer size for the request and the reply together. This model is fitted by a statistical model-tracking procedure [13]. There is a large literature on LQN models and their use to model distributed systems [1]. Other performance models could also be used for application awareness; but for reasons of space we do not address them here.

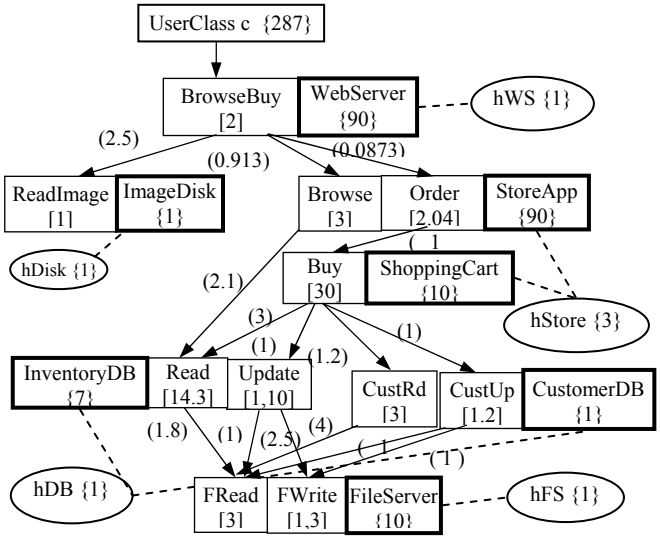

Figure 3 LQN model of a three-tier service system, deploying one replica per task

In [9] an empirical fitted linear response-time model was conveyed from the application manager to the cloud manager, based on total VM capacity provided by a single cloud.

Awareness of the Cloud

To guide an application manager which does not delegate its decisions, each cloud can provide awareness of its state. Some options, in increasing order of complexity and completeness, are:

1. **Available resources:** the cloud publishes the availability of VMs of different capacities (minimal awareness)
2. **Cloud-Opt Properties**: optimization-related properties of the cloud provider, such as the cost and availability of different resources (partial cloud awareness)
3. **Predictive Awareness:** A performance model for the total cloud infrastructure which can predict the delay effect of a deployment (full cloud awareness).

Again, awareness can be considered to comprise a model of the cloud resources, including communications delays. For a hierarchical cloud like SAVI, or a multi-cloud, it should include communications delays between the sub-clouds as well.

Awareness of cloud state is of more use to a global manager than an application manager, simply because the application manager does not make detailed deployment decisions. However it needs awareness of cloud delays to determine how to distribute the application across multiple clouds, to achieve response time goals.

4. EXPLOITING PARTIAL APPLICATION AWARENESS

Our vision is a flexible architecture combining a manager for each cloud and each application. Cloud managers would determine detailed deployment of VMs, and might make additional decisions (such as tuning the capacity allocated to a VM, or even moving a VM to another cloud). Application managers would request resources from the separate clouds. The managers would exchange a variety of forms of information according to their own capabilities and policies about revealing their critical information. The separate managers therefore should be capable of interpreting

as many kinds of awareness data as possible, in reaching their own decisions.

Partial application awareness has not been much addressed as yet. It may however be more realistic in the near term. Interesting research questions include:

1. What is the impact on deployment of optimization using different degrees of awareness?
2. What is the value of application awareness? That is, what is the improvement in various optimization cost functions such as monetary cost of operation of the cloud, energy cost of the cloud, aggregate penalty cost (of some kind) related to the provided QoS to all applications together?
3. Following from 2, what is the most useful form of application awareness?
4. Using App-Opt Properties, some kind of collaborative optimization may adequately coordinate two separate managers for the application and the cloud, without revealing to each other their inner workings. A decomposed optimization strategy might be considered, exchanging marginal resource pricing, and the value of additional capacity to the application users, expressed in suitable units.

5. IMPACT ON CLOUD ARCHITECTURE

A defining characteristic of cloud computing is that there is little concern for *where* in the cloud a computing task takes place. However, when communication latencies are vital to meeting response time requirements for modern cloud-based applications, where the task is allocated relative to the entities it communicates with becomes vital. This has a direct impact on cloud architecture.

For example, the SAVI concept envisions small clouds that are located close to the users (both physically and in the sense of latency). Given some awareness of the application (such as an LQN model), the cloud manager can then make sure to co-locate tasks that communicate heavily and are highly latency-sensitive. For example, if building a real-time 3-dimensional view from multiple cell phone camera inputs, then low latencies to the users, is critical, implying that tasks handling the video streams should be co-located on the edge cloud.

Providing full application awareness, as in the LQN model, allows the cloud manager to decide where to allocate tasks so that it can meet response time requirements. Given the LQN, it can identify tasks that are best handled on the edge cloud while other latency-insensitive tasks can be handled centrally in the main cloud, or even a much more physically remote but lightly loaded cloud.

Some of the cloud architecture issues that arise include: How much computing power should be available in an edge cloud? How many edge clouds should there be (e.g. several per city?)? Should there be a hierarchy of clouds, e.g. several small edge clouds in a city linked to an intermediate size city cloud, which is itself linked to the main cloud center?

6. CONCLUSIONS

Greater levels of awareness of the cloud by an application manager, or of its applications by a cloud manager, is a largely unexplored approach which appears to be necessary for managing real-time dynamically changing applications over multi-clouds. We are pursuing this approach in the context of the SAVI cloud.

ACKNOWLEDGMENTS

This research was supported by the SAVI Strategic Research Network (Smart Applications on Virtual Infrastructure, funded by NSERC (the Natural Sciences and Engineering Research Council of Canada).

REFERENCES

[1] V. Cardellini, E. Casalicchio, F. Lo Presti, L. Silvestri, “SLA-Aware resource management for application service providers in the cloud”, Proc 1st Int. Symp on Network Cloud Computing and Applications (NCCA), pp 20 - 27, Toulouse, Nov 2011

[2] S. Costache, N. Parlavantzas, C. Morin, S. Kortas, “An Economic Approach for Application QoS Management in Clouds”, Euro-Par 2011 Workshops Part II, LNCS 7156, Springer, pp. 426–435, 2012.

[3] Franks, G., T. Al-Omari, et al. (2009). "Enhanced modeling and solution of layered queueing networks." IEEE Transactions on Software Engineering 35(2): 148-161.

[4] H. Ghanbari, B. Simmons, M. Litoiu, and G. Iszlai, “Feedback-based optimization of a private cloud,” Future Generation Computer Systems , 2011.

[5] J. Li, J. Chinneck, C.M. Woodside, M. Litoiu, G. Iszlai, "Performance Model Driven QoS Guarantees and Optimization in Clouds", in Proc Workshop on Software Engineering Challenges in Cloud Computing @ ICSE 2009, Vancouver, May 2009.

[6] Li, J., Chinneck, J., Woodside, M., Litoiu, M, "Fast Scalable Optimization to Configure Service Systems having Cost and Quality of Service Constraints", Proc 6th Int Conf on Autonomic Computing (ICAC.09), Barcelona, June 2009.

[7] Li, J., Chinneck, J., Woodside, M., Litoiu, M, “Deployment of Services in a Cloud Subject to Memory and License Constraints", in Proc 2nd IEEE Intl Conf on Cloud Computing , Bangalore, India, September 21-25, 2009.

[8] Li, J., Chinneck, J., Woodside, M., Litoiu, M., “CloudOpt: Multi-Goal Optimization of Application Deployments across a Cloud", 7th Int. Conf. on Network and Service Management (CNSM 2011), October 24-28 2011, Paris, France

[9] H.N. Van, F. D. Tran, J.-M. Menaud, “SLA-aware virtual resource management for cloud infrastructures”, 9th IEEE Int. Conf. on Computer and Information Technology (CIT'09), Xiamen, China (2009)

[10] X. Wang, Y. Xue, L. Fan, R. Wang, Z. Du , “Research on adaptive QoS-aware resource reservation management in cloud environments”, Proc. 2011 IEEE Asia-Pacific Services Computing Conference (APSCC), pp.147 - 152, Dec. 2011.

[11] L. Wu, . S. K. Garg, S. Versteeg, and R. Buyya. "SLA-based Resource Provisioning for Software as a Service Applications in Cloud Computing Environments", IEEE Transactions on Services Computing preprint, 21 Nov. 2013

[12] Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba, “Dynamic service placement in geographically distributed clouds,” in Distributed Computing Systems (ICDCS), 32nd International Conference on . IEEE, 2012, pp. 526–535.

[13] T. Zheng, C.M.,Woodside, M. Litoiu, "Performance Model Estimation And Tracking Using Optimal Filters", IEEE Trans on Software Engineering, Vol. 34 , No. 3, pp. 391-406, 2008.

[14] SAVI Strategic Network, http://savinetwork.ca, Jan 2013