Locality-aware Cooperation in Distributed IaaS Infrastructures

Jonathan Pastor[1], Marin Bertier[2], Flavien Quesnel[1], Adrien Lebre[1] and Cedric Tedeschi[3]

[1] ASCOLA Research Group, Mines Nantes / Inria / LINA, Nantes, France
`firstname.lastname@mines-nantes.fr`
[2] ASAP Research Group, INSA / Inria / IRISA, Rennes, France
`marin.bertier@irisa.fr`
[3] Myriads Research Group, Université de Rennes 1 / Inria / IRISA, Rennes, France
`cedric.tedeschi@inria.fr`

Abstract. With the adoption of distributed cloud computing infrastructures as the new platform to deliver utility computing paradigm, new algorithms leveraging peer to peer approaches have been proposed for scheduling virtual machine (VM). Although these proposals considerably improve the scalability enabling the management of hundreds of thousands of VM upon thousands of physical machines (PMs), multisite infrastructures introduce network overhead, which can have a dramatic impact on performances when there is no mechanism in charge of favoring intra-site vs. inter site manipulations.
To reduce such an impact, locality properties should be considered as a key element, e.g. PMs should collaborate first with their neighbourghood from the same geographical site before contacing remote ones. As network bandwidth/latency fluctuate over time, using a static partitionning of the resources is not enough.
This paper introduces a new building block built on a vivaldi overlay that maximizes efficient collaborations between PMs. We combined this mechanism with DVMS , a large scale virtual machine scheduler and show its benefit by discussing several experiments performed on four distinct sites of the Grid'5000 tesbed. Thanks to our proposal and without changing the scheduling decision algorithm, the number of inter-site operations has been reduced by 66properties to improve performance of massive distributed cloud platforms.

Keywords: Cloud computing, locality, peer to peer, network overlay, vivaldi, DVMS, virtual machine scheduling

1 Introduction

Introduced few years ago [5], the new trend to deliver cloud computing resources, in particular Infrastructure as a Service solutions, consists in leveraging several infrastructures distributed world-wide. If such distributed cloud computing platforms deliver undeniable advantages to address important challenges such as reliability, latency or even in somehow jurisdiction concerns, most mechanisms that were previously used to operate centralized IaaS platforms must be revisited to offer the same level of transparency for the end-users. Keeping such an objective in mind, the use of P2P paradigm is strongly investigated. This is particularly true for instance for scheduling algorithms in charge of assigning VMs on top of PMs according to their effective needs (and reciprocally usages). Indeed and although major improvements have been done, centralized approaches [7] are not scalable enough and hierarchical solutions [4] face important limitations regarding the reactivity to take into account physical topology changes, an important criteria in such widely distributed infrastructures.

2 Background

2.1 DVMS

Overview DVMS [10] (*Distributed Virtual Machine Scheduler*) is a framework that enables VMs to be scheduled cooperatively and dynamically in large scale distributed systems.

DVMS is deployed as a set of agents that are organized following a ring topology and that cooperate with one another to guarantee the quality of service (QoS) for the VMs.

When a node cannot guarantee the QoS for its hosted VMs or when it is under-utilized, it starts an iterative scheduling procedure (ISP) by querying its neighbor to find a better placement; it thus becomes the initiator of the ISP. If the request cannot be satisfied by the neighbor, it is forwarded to the following free one until the ISP succeeds. This approach allows each ISP to send requests only to a minimal number of nodes, thus decreasing the scheduling time without requiring a central point. In addition, this approach allows several ISPs to occur independently at the same moment throughout the infrastructure; in other words, scheduling is performed on partitions of the system that are created dynamically, which significantly improves the reactivity of the system. Communications are handled efficiently, as each node involved in a partition can forward a request directly to the first node outside its partition, by means of a "first out" relation.

An example involving three partitions is shown on Figure 1; in particular, we can see the growth of partition 1 between two steps.

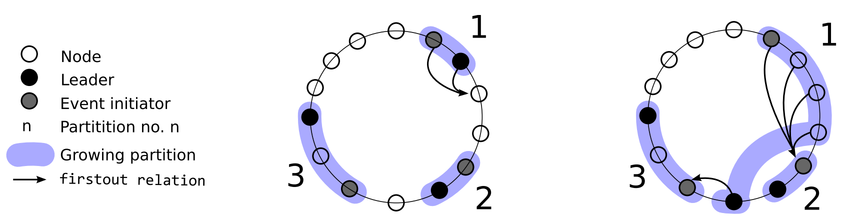

Fig. 1. Solving three problems simultaneously and independently with DVMS

Limitations

2.2 P2P - locality

La prise en compte de la localit dans la construction de rseaux logiques (*overlays*) a t initialement propos dans le rseau logique structur Pastry [12] afin d'optimiser la latence du routage. La recherche de ces nœuds proches est fonde sur un change pdiodique de rfrences de nœuds.

Le mme concept a t propos dans les rseaux logiques non structurs afin de permettre chaque nœud de dcouvrir des nœuds du systmes les plus *proches*. La notion de proximit peut recouvrir toute mtrique transitive entre deux nœuds, en particulier le temps de latence entre les nœuds [**?**].

Le protocole Vivaldi [2], sur lequel notre rseau logique est fond, a une approche particulire. En effet, il fournit chaque nœud des coordonnes dans un espace multi-dimensionnel refltant sa position dans le rseau physique. Initialement, chaque nœud prend une position alatoire de l'espace, et choisit un petit sous-ensemble de nœuds. Puis, il se rapproche dans l'espace, des nœuds avec lesquels il a une faible latence et s'loigne dans le cas inverse. Vivaldi ne permet donc pas de connatre les nœuds qui lui sont proches dans le rseau, mais de les reconnatre via leurs coordonnes.

Les approches prcdentes maintiennent constamment la connaissance des nœudes proches afin de fournir le meilleur nœud possible, au cot de communications priodiques (indpendamment de la quantit de requtes effectives.) Notre approche se distingue par une approche paresseuse consistant rechercher des nœuds proches (en s'appuyant sur les coordonnes Vivaldi) lors des requtes, adaptant ainsi la qualit de la rponse la frquence des requtes.

3 Contributions

3.1 Locality based overlay

- clustering
- Vivaldi + spirale

3.2 Dvms + PeerActor + locality

4 Experimentations

4.1 Implementation

A prototype of DVMS leveraging locality based overlay has been developed. The current version of DVMS are been developed over the PeerActor abstraction. PeerActor provides network abstraction that enables the design of distributed algorithm that are network overlay agnostic. We have developed two different overlay for the Peer Actor abstraction: Chord and a locality based overlay over Vivaldi.

The strength of this software architecture is that to enable an algorithm (like DVMS) to comply with a given network overlay, it only have to follow the Peer Actor API. This way, we were able to run DVMS over Chord or Vivaldi without any modification in its source code.

4.2 Grid5000' experiments

Objectives The prototype has been tested with a various number of experiments conducted on the Grid5000' testbed. The main objective of the experiments was to estimate impact of locality on the performance of a distributed scheduling algorithm. A significant portion of the reconfiguration time is spent in live migration of virtual machines, which depends of network parameters such as latency and bandwidth. One way to improve performance of distributed scheduling algorithm is is to promote collaboration between close ressources, which can be reach by maximising this ratio:

$$\frac{number\ of\ intrasite\ migrations}{number\ of\ migrations}$$

Experimental protocol For each experiment, we booked 40 compute servers spread on 4 geographical sites and 1 service server. The compute servers were used to run virtual machines and DVMS while the service node is used to stress several parameters of virtual machines.

Each compute node will host a number of virtual machines proportional to the number of CPU cores it has. In our case:

$$number\ of\ virtual\ machines = 1.3 \times number\ of\ cores$$

Results The impact of locality on DVMS is significant: using a Vivaldi based network overlay leads to an average number of 83% of intrasite migrations while using a Chord based DVMS leads a ratio of 50% of intrasite migrations, as depicted in the following table:

network overlay	average number of intrasite migrations	average number of migrations
Vivaldi	83	100
Chord	40	80

Complete this section with more experimentation results.

5 Related work

5.1 DVMS

Centralized approach [6]

Hierarchical approach [4]

Distributed approaches [1, 3, 8, 9, 11, 13]

5.2 P2P

6 Conclusion

References

1. Barbagallo, D., Di Nitto, E., Dubois, D., Mirandola, R.: A bio-inspired algorithm for energy optimization in a self-organizing data center. In: Self-Organizing Architectures, Lecture Notes in Computer Science, vol. 6090, pp. 127–151. Springer, Berlin/Heidelberg, Germany (2010)
2. Dabek, F., Cox, R., Kaashoek, M.F., Morris, R.: Vivaldi: a decentralized network coordinate system. In: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 15–26. SIGCOMM '04 (2004)
3. Feller, E., Morin, C., Esnault, A.: A Case for Fully Decentralized Dynamic VM Consolidation in Clouds. In: CloudCom '12: 4th IEEE International Conference on Cloud Computing Technology and Science. IEEE Computer Society, Washington, DC, USA (Dec 2012)
4. Feller, E., Rilling, L., Morin, C.: Snooze: A Scalable and Autonomic Virtual Machine Management Framework for Private Clouds. In: CCGRID '12: Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. pp. 482–489. IEEE Computer Society, Washington, DC, USA (May 2012)
5. Greenberg, A., Hamilton, J., Maltz, D.A., Patel, P.: The Cost of a Cloud: Research Problems in Data Center Networks. SIGCOMM Comput. Commun. Rev. 39(1), 68–73 (Dec 2008)
6. Hermenier, F., Demassey, S., Lorca, X.: Bin repacking scheduling in virtualized datacenters. In: CP '11: Proceedings of the 17th international conference on Principles and practice of constraint programming. pp. 27–41. Springer, Berlin/Heidelberg, Germany (2011)
7. Hermenier, F., Lawall, J., Muller, G.: BtrPlace: A Flexible Consolidation Manager for Highly Available Applications. IEEE Transactions on Dependable and Secure Computing 99(PrePrints) (2013)
8. Marzolla, M., Babaoglu, O., Panzieri, F.: Server consolidation in Clouds through gossiping. In: WoWMoM '11: Proceedings of the 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks. pp. 1–6. IEEE Computer Society, Washington, DC, USA (Jun 2011)
9. Mastroianni, C., Meo, M., Papuzzo, G.: Self-economy in cloud data centers: statistical assignment and migration of virtual machines. In: Euro-Par '11: Proceedings of the 17th international conference on Parallel processing. vol. 1. Springer, Berlin/Heidelberg, Germany (2011)
10. Quesnel, F., Lèbre, A., Südholt, M.: Cooperative and Reactive Scheduling in Large-Scale Virtualized Platforms with DVMS. Concurrency and Computation: Practice and Experience 25(12), 1643–1655 (Aug 2013)
11. Rouzaud Cornabas, J.: A Distributed and Collaborative Dynamic Load Balancer for Virtual Machine. In: Euro-Par 2010 Parallel Processing Workshops, Lecture Notes in Computer Science, vol. 6586, pp. 641–648. Springer, Berlin/Heidelberg, Germany (Aug 2011)

12. Rowstron, A.I.T., Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware). pp. 329–350. Heidelberg, Germany (Nov 2001)
13. Yazir, Y.O., Matthews, C., Farahbod, R., Neville, S., Guitouni, A., Ganti, S., Coady, Y.: Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis. In: Cloud '10: IEEE 3rd International Conference on Cloud Computing. pp. 91–98. IEEE Computer Society, Los Alamitos, CA, USA (Jul 2010)