

Towards A Volunteer Cloud Architecture

Abdulelah Alwabel, Robert Walters and Gary Wills
Electronics and Computer Science School,
University of Southampton
{aala10, rjw1, gbw}@ecs.soton.ac.uk

Abstract. Volunteer cloud computing is a new type of clouds aiming at moving volunteer computing towards the cloud. The new cloud type is motivated by the fact that building a cloud out of non-dedicated resources can be useful for scientific projects which cannot afford the cost of consumption of cloud services provided by cloud service providers such as Amazon. However, Volunteer Clouds are in its infancy level with some challenges and issues that ought to be tackled. This paper presents a new architecture which can facilitate volunteer clouds being a viable cloud solution.

1. Introduction

Volunteer cloud computing (VCC) mixes the concepts of cloud computing (Armbrust et al., 2010) and volunteer computing (Anderson & Fedak, 2006) by offering all or some of cloud services without charging. VCC has some advantages compared to commercial clouds. The First advantage is the cost effectiveness of volunteer clouds since all resources are offered voluntarily which can be very useful for projects that cannot afford commercial clouds' services (Chandra & Weissman, 2009). Secondly, it reduces energy consumption and gas emissions because it utilises computing resources that would otherwise remain idle. Arpaci et al., (1995) show that the average percentage of local resources being idle within an organisation is about 80%. In contrast, commercial clouds set up a huge number of dedicated resources in their data centres, thus, they have a negative impact on the environment since their data centres consume massive amounts of electricity (Gupta & Awasthi, 2009). Finally, commercial clouds are inefficient in terms of data mobility and pay little attention to the location of clients (Weissman et al., 2011). This paper presents and discusses a new architecture which can facilitate volunteer clouds being a viable cloud solution

2. The Architecture

The abstract level of the architecture, Figure 1, divides VCC into three layers: (i) a service layer; (ii) a middleware layer; and (iii) a physical layer. The service layer is concerned with delivering services in a way similar to commercial clouds.

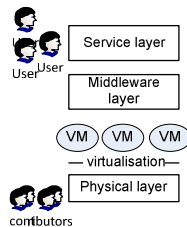


Figure 1: VCC Architecture

2.1. Service Layer

The service layer provides services via an interface to customers based on SOA approach. The business model in VCC is similar to that of commercial clouds. VCC's contributors volunteer their resources to form a VCC for a certain time, and they may be services consumers at the same time if they wish.

2.2. Middleware Layer

The aim of the layer is to provide resources to the service layer as they would be provided by a commercial cloud. The layer, shown in Figure 2, consists of *task management* and *QoS management*. *Task management* works with tasks received from the service layer. It involves *task scheduler*, *load balancing* and *self-automation*. The *task scheduler* organises tasks coming from the service layer by passing them to suitable resources which are offered in the physical layer. *The load balancing* ensures that the load is distributed appropriately, thus minimizing the required time to process a task. *Self-automation*

helps to provide the rapid elasticity in VCCs. It allows users to scale services up or down according to their needs. *QoS management* ensures that a minimum quality level is maintained. The *performance monitor* in *QoS management* ensures that the performance of each task is maintained at an acceptable level which is reported in the *SLA reporting* component. Node volatility is quite high in VCCs, so the *performance monitor* must cooperate with the *resource management* to find reliable nodes among available resources that suite each task. The *fault recovery* component can be vital with regards to improving the performance of the overall VCCs.

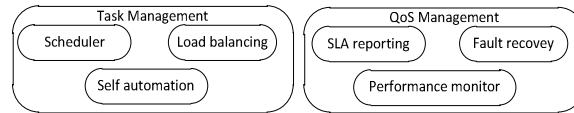


Figure 2: Middleware Layer

2.3. Physical Layer

Resource manager, Figure 3, is responsible for resource aggregation, resource allocation and resource monitoring. It aggregates volunteer nodes denoted by the public. An aggregation mechanism can classify resources according to a number of criteria with the aim of optimising the quality of service. For example, the history of each volunteer node can be useful in terms of recognising which node should be selected by the resource allocator for each task. The *allocator* receives tasks from tasks management and allocates them to the required resources. The *allocator* can decrease the interruption of services by assigning tasks to nodes with higher reliability. The *monitor* component observes allocated resources regularly in case any of them becomes unavailable. In this case, the *monitor* informs the *fault recovery* in the middleware in order to recover the task from a replicated node.

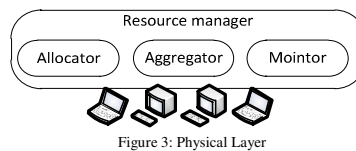


Figure 3: Physical Layer

3. Conclusion

In conclusion, this paper presented an architecture for volunteer clouds which can be used to enhance volunteer clouds. The architecture has three layers: service layer which is an interface between customers and a volunteer cloud; middleware which is responsible of processing tasks while preserving QoS as requested by customers; and physical layer which contains raw resources (typically PCs, laptops ...etc) managed by *resource manager*. Our future work will be about improving the quality level of services provided by VCCs. This involves designing performance metrics tool and employing fault recovery techniques to improve the overall reliability of VCCs.

References

- Anderson, D. P., & Fedak, G. (2006). The Computational and Storage Potential of Volunteer Computing. *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, 73-80. Ieee. doi:10.1109/CCGRID.2006.101
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., et al. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1721672>
- Arpaci, R. H., Dussseau, A. C., Vahdat, A. M., Liu, L. T., Anderson, T. E., & Patterson, D. A. (1995). *The Interaction of Parallel and Sequential Workloads on a Network of Workstations*. *Science* (Vol. 23). ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=223618>
- Chandra, A., & Weissman, J. (2009). Nebulas: Using distributed voluntary resources to build clouds. *Proceedings of the 2009 conference on Hot topics in cloud computing* (pp. 2–2). USENIX Association. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.148.7267>
- Gupta, A., & Awasthi, L. K. (2009). Peer enterprises: A viable alternative to Cloud computing? *Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference on* (Vol. 2, pp. 1–6). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5439456
- Weissman, J. B., Sundarajan, P., Gupta, A., Ryden, M., Nair, R., & Chandra, A. (2011). Early experience with the distributed nebula cloud. *Proceedings of the fourth international workshop on Data-intensive distributed computing* (pp. 17–26). ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1996019>