Evaluating Workload Management Techniques for Cloud Computing

Rizwan Mian, Feb 10, 2012 ©

Problem Statement

The Grid Computing (grid) refers to technologies that allow consumers to obtain computing on demand, analogous in form and utility to electrical grid [1]. Grids and related application technologies are enabling scientists and engineers to build more and more complex applications for managing and processing large data sets, and for executing scientific experiments on distributed heterogeneous resources [2]. Cloud Computing (cloud) aims for the same dream of using computing as a utility [3]. The fundamental vision and concepts are the same. The vision of a global grid has not yet been realised but it might be fair to say that cloud builds on the lessons learnt from building a grid.

The key difference between cloud and grid is that cloud resources are managed and used by different parties. Typically cloud vendors happen to be large establishments like Amazon and Google. In contrast, cloud users vary in scale. This allows separation of concerns. Most existing grid middleware, which enables access to grid resources, is widely perceived as being too heavyweight. The heavyweight nature of middleware, especially on the client-side, has severely restricted the uptake of grids by users [4]. On the other hand, cloud providers – perhaps learning from grid experience – manage complexity on their side and offer interfaces of varying abstraction to their users. From a hardware point of view, cloud offers the illusion of infinite computing resources to users on demand [3].

As a consequence there are many new application opportunities offered by cloud e.g. analytics, batch processing [3]. Different applications may result in different types of workloads. Workloads can vary from unrelated and independent jobs to related and structured workflows. These workloads need to be managed on a cloud. The cloud provider may provide general workload management policies offering a higher level of abstraction to users. However, the policy may be oblivious to a user's internal workload utility properties such as priority or ordering. In such a case, the user must manage workload.

Possible Research

Since cloud emerged in last few years, it is hoped that there are many low hanging fruits. Workload management techniques from grid and general distributed computing may be evaluated in cloud for their "effectiveness". The end goals are (a) to develop a platform to evaluate different workload management techniques (policies), and (b) to evaluate policies.

The cloud platform used would be a simulator called CloudSim (csim) [5]. It is argued that a simulator facilitates researchers to investigate specific aspects of a system, without getting involved in low-level details [5]. Simulation has been used for studying policies. For example, [6] have used simulation for studying rescheduling policy of workflows on grids.

Using bottom-up approach, existing vanilla policies – such as first-come-first-server (fcfs), deadline, budget – would be used to construct this platform. To start with, jobs can have random duration. To introduce realism, cloud resources would be made similar to Amazon EC2 instances. Simulated workload would be based on use cases derived from the Map-Reduce case study [7] or Web-Services of Amazon[1]. Alternately, jobs can be generated whose duration distribution is heavy-tailed – as [8] observes distribution of lifetimes of UNIX processes. Similarly, the size of simulated cloud can be based on the cluster required for Google services [7, 9].

Once evaluated in simulation, the promising policies can be further evaluated on a real cloud.

The platform once developed can be a fundamental base for other interesting studies. The core models for decision making in the policies can either be hard (e.g. analytic, algorithmic etc.) like [10] or soft (e.g. fuzzy controlled, neural networked etc.) like [11]. A comparison of these models can be made for their suitability on the cloud.

Another interesting study is the recently suggested framework of autonomic workflow execution [12]. Pairs of (cost models and optimization algorithms) can be empirically evaluated for their effectiveness – effectiveness being minimal execution cost and time in this case.

[1] Amazon Web Services, (31.1.10), http://aws.amazon.com/solutions/case-studies/

Finally, it is hoped that this project would provide basis to develop a generic platform to evaluate different workload management techniques.

Proposed Research

For the purpose of this project, existing vanilla policies would be evaluated using csim, similar to evaluation of policies in a grid simulator, GridSim [13].

Project Timeline

Below are the major objectives of the project. Note, these objectives may overlap, may be pursued out of order, and may need to be revised in the experience of achieved objectives.

#	**Objectives (O)**	**dates**
1	Understanding csim [5]	Done
2	Playing with csim examples	Done
3	Reproducing experiments [5]	(partially) done
4	Extending csim to include data transfer time	Start: 7th Mar, 10 Deadline: 28th Feb, 10
5	Extending csim cost model to include: • Execution time of vm • Transfer time of data	Start: 7th Mar, 10 Deadline: 28th Feb, 10
6	Simulating MapReduce scenario [7]	Week of 7th Feb, 10
7	Implementing Policies: • fcfs • Deadline • Budget	Start: 28th Mar, 10 Deadline: 14th Mar, 10
8	Extending broker: • Queuing jobs • Handling uncertainty	Start: 20th Feb, 10 Deadline: 6th Mar, 10
9	Introducing uncertainty: random delays and failures	Start:6th Mar, 10 Deadline: 13th Mar, 10
10	Experimentation	Start: 6th Mar, 10 Deadline: 31st Mar, 10
11	Measurements	Continuous
12	Writing up	Deadline: 16th Apr, 10

Concerns

It has been observed the current implementation of csim is missing some critical features that exist in a cloud e.g. O4, O5 and O8. It is hoped that these missing features are relatively easy to implement, and further effort can be invested in proposed research.

References

1. Foster, I., et al. *Cloud Computing and Grid Computing 360-Degree Compared*. in *Grid Computing Environments Workshop, 2008. GCE '08*. 2008.
2. Yu, J. and R. Buyya, *A Taxonomy of Workflow Management Systems for Grid Computing.* eprint arXiv:cs/0503025, 2005.
3. Armbrust, M., et al., *Above the Clouds: A Berkeley View of Cloud Computing*, in *Technical Report No. UCB/EECS-2009-28*. 2009, University of California at Berkeley.
4. Coveney, P.V., et al., *The application hosting environment: Lightweight middleware for grid-based computational science.* Computer Physics Communications, 2007. **176**(6): p. 406.
5. Buyya, R., R. Ranjan, and R.N. Calheiros. *Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities*. 2009. Leipzig, Germany: IEEE Computer Society.

6. Sakellariou, R. and H. Zhao, *A low-cost rescheduling policy for efficient mapping of workflows on grid systems.* Scientific Programming AxGrids 2004, 2004. **12**(4): p. 253-262.
7. Dean, J. and S. Ghemawat, *MapReduce: simplified data processing on large clusters.* Communications of the ACM, 2008. **51**(1): p. 107.
8. Harchol-Balter, M. and A.B. Downey, *Exploiting process lifetime distributions for dynamic load balancing.* ACM Transactions on Computer Systems, 1997. **15**(3): p. 253.
9. Barroso, L.A., J. Dean, and U. Holzle, *Web search for a planet: The Google cluster architecture.* IEEE Micro, 2003. **23**(2): p. 22-8.
10. Wolski, R., *Dynamically forecasting network performance using the Network Weather Service.* Cluster Computing, 1998. **1**(1): p. 119-32.
11. Yu, K.M., et al., *A fuzzy neural network based scheduling algorithm for job assignment on computational grids*, in *Network-Based Information Systems, Proceedings*. 2007, Springer-Verlag Berlin: Berlin. p. 533-542.
12. Paton, N.W., et al., *Optimizing Utility in Cloud Computing through Autonomic Workload Execution.* IEEE Data Engineering Bulletin, 2009. **32**(1): p. 51-58.
13. Buyya, R. and M. Murshed, *GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing.* Concurrency and Computation: Practice and Experience, 2002. **14**(13-15): p. 1175-1220.

Ref: gridsim,

X Models: hard – analytical, algorithmic; soft – fuzzy control system, neural network (nn)

X Framework & architecture based study vs workload management techniques evaluation

X Evaluation: depends on the requirement

Realism:
X - resource type: vm (similar to ec2)
X - reproducing amazon case studies
- trying different worload management techniques with amazon case studies

Policies
- fcfs
- deadline
- budget
- budge-cost

X a technique for will be empirically verified.

X Def. of cloud and grid
X Differences identifying one
X Possible applications

The Cloud offers a different paradigm for application execution

X One of the key administrative difference is that cloud resources are provided and used by different.There are many

X Many research opportunities in cloud. Testing out techniques in cloud.

,of autonomic workflow management. Using bottom-up approach, existing techniques

Def. of workflow management

X Usage of simulator
X Testing out autonomic workflow execution