Campus Grids

Derek Weitzel
Computer Science and Engineering
University of Nebraska–Lincoln
Lincoln, NE 66588-0115
dweitzel@cse.unl.edu

November 3, 2010

Abstract

At many universities, each department maintains a independent cluster. Many of these clusters run under utilized. In this paper we will describe a framework and technologies that where used to link departmental clusters such that submission at one cluster could lead to execution on another. This framework is then further expanded to operating on a production national grid.

1 Introduction

The problem we have solved

- Department clusters waste power by being under utilized significant portions of time.
- Researchers have peaks in usage (think paper writing) and need overflow capacity.
- Overflow capacity for their local cluster.

Why the problem is not already solved or other solutions are ineffective in one or more important ways

- A common solution to linking clusters is condor flocking. Flocking requires every cluster to run condor daemons on their nodes.
- Another solution is single vendor/software solution. This again requires buy-in from all departments on campus.
- Placing globus gatekeepers on each cluster would allow jobs to be submitted to each cluster without modifying the underlying batch system. But, this would require a higher layer of abstraction over the globus gatekeepers to optimally balance load between clusters. Also, it does not provide a method for transparent execution on other clusters.

Why our solution is worth considering and why is it effective in some way that others are not

- The framework described in this paper is designed as a modular framework that will allow clusters overflow onto each other and to the grid. This framework requires running condor on only one node in the cluster. Each job that comes from another cluster will go through the default scheduler, whether it's PBS, SGE, or LSF.

How the rest of the paper is structured

- The short statement below is often all you need, but you should change it when your paper has a different structure, or when more information is *required* to describe what a given section contains. If it isn't *required* then you don't want to say it here.

The rest of this paper first discusses related work in Section 2, and then describes our implementation in Section 3. Section 4 describes how we evaluated our system and presents the results. Section 5 presents our conclusions and describes future work.

2 Related Work

Other efforts that exist to solve this problem and why are they less effective than our method

- Globus is a translation layer between the globus specific Resource Specification Language, and the local resource manager. It has been very successful in that is has a large install base. Globus also has deep integration with standard grid credentials.
- Condor Flocking
- Diagrid
- GlideinWMS
- Panda

Other efforts that exist to solve related problems that are relevant, how are they relevant, and why are they less effective than our solution for this problem

- Many times no one has solved your exact problem before, but others have solved closely related problems or problems with aspects that are strongly analogous to aspects of your problem

3 Implementation

What we (will do | did): *Our Solution*

- Another way to look at this section is as a paper, within a paper, describing your implementation. That viewpoint makes this the introduction to the subordinate paper, which should describe the overall structure of your implementation and how it is designed to address the problem effectively.
- Then, describe the structure of the rest of this section, and what each subsection describes.

How our solution (will | does) work

- This is the body of the subordinate paper describing your solution. It may be divided into several subsections as required by the nature of your implementation.
- The level of detail about how the solution works is determined by what is appropriate to the type of paper (conference, journal, technical report)
- This section can be fairly short for conference papers, fairly long for journal papers, or *quite* long in technical reports. It all depends on the purpose of the paper and the target audience

- Proposals are necessarily a good deal more vague in this section since you have to convince someone you know enough to have a good chance of building a solution, but that you have not *already* done so.

4 Evaluation

How we tested our solution

- Performance metrics
- Performance parameters
- Experimental design

How our solution performed, how its performance compared to that of other solutions mentioned in related work, and how these results show that our solution is effective

- Presentation and Interpretation
- Why, how, and to what degree our solution is better
- Why the reader should be impressed with our solution
- Comments

Context and limitations of our solution as required for summation

- What the results *do* and *do not* say

5 Conclusions and Future Work

The problem we have solved

- The most succinct statement of the problem in the paper. Ideally one sentence. More realistically two or three. Remember that you simply state it without argument. If you have written a good paper you are simply reminding the reader of what they now believe and of how much they agree with you.

Our solution to the problem

- Again, the succinct statement that you have presented a solution
- Sometimes it works well to leave it at that and not even describe your solution here. If you do, then again state your solution in one or two sentences taking the rhetorical stance that this is all obvious. If you have a good solution and have written an effective paper, then the reader already agrees with you.

Why our solution is worthwhile in some significant way

- Again, a succinct restatement in just a few sentences of why your solution is worthwhile assuming the reader already agrees with you

Why the reader should be impressed and/or pleased to have read the paper

- A few sentences about why your solution is valuable, and thus why the reader should be glad to have read the paper and why they should be glad you did this work.

What we will (or could) do next

- Improve our solution
- Apply our solution to harder or more realistic versions of this problem
- Apply our solution or a related solution to a related problem

References

[1] Anderson, J., Ramamurthy, S., Jeffay, K., "Real-Time Computing with Lock-Free Shared Objects", *Proceedings of the 16th IEEE Real-Time Systems Symposium*, IEEE Computer Society Press, December 1995, pp. 28-37.

[2] Baruah, S., Howell, R., Rosier, L., "Algorithms and Complexity Concerning the Preemptively Scheduling of Periodic, Real-Time Tasks on One Processor," *Real-Time Systems Journal*, Vol. 2, 1990, pp. 301-324.

[3] Goddard, S., Jeffay, K. "Analyzing the Real-Time Properties of a Dataflow Execution Paradigm using a Synthetic Aperture Radar Application," *Proc. IEEE Real-Time Technology and Applications Symposium*, June 1997, pp. 60-71.