Monitoring, Analyzing, and Visualizing Distributed Computing Clusters

Peter Bui, John Stupka
Computer Science
{buipj,stupkajr}@uwec.edu

1 Project Proposal

In recent times, many scientific disciplines such as Biology, Chemistry, and Physics have grown increasingly reliant on computing for simulation, modeling, analysis, and processing of immense amounts of data. In order to perform these complex and ever-growing computations, it has become necessary to have a distributed computing cluster which allows researchers to harness multiple machines as a single system in order to improve the speed in which experiments and computations can be performed.

At UW-Eau Claire, the Physics and Chemistry departments spearheaded the local push for computational science by purchasing and maintaining their own computational clusters. A few years ago, a group of STEM faculty from multiple disciplines including Biology, Chemistry, Computer Science, Material Science, Math, and Physics proposed a new larger supercomputing system that would unify the existing systems from different departments while introducing a newly procured system. In 2012, this proposal was accepted as a Blugold commitment project and thus the Blugold Supercomputer (BGSC) was born.

Currently, the BGSC is still undergoing testing and configuration, but is generally operational and available to all UWEC faculty and students for research and teaching. One key element missing from the system, however, is a framework for monitoring, analyzing, and visualizing the activity of not only the machines, but also user activity. Because a cluster consists of multiple machines working together, it is always possible that one or more machines may be misbehaving (e.g. hardware failure, software misconfiguration). For a fully operational cluster, it is imperative that a system exists to monitor the hardware and notify the appropriate system administrators. Additionally, because this system is used by a variety of users, it would be prudent to track user activity to ensure consistency and fairness in allocating resources on the system.

Therefore, BGSC system administrators need a software tool that will allow them to ask probing questions such as:

1. **Machine Statistics**: Over the course of a day, what is the typical CPU or memory usage? How fast does disk storage grow? Are any machines under-utilized?

2. **Machine Availability**: Which machines get used the most? Which machines appear to be down or crash the most often?

3. **Machine Performance**: Which machines perform the best (and under which circumstances)?

4. **User Activity**: Which user, group, or course is the most active? Which set of users causes the most problems (and why)?

5. **Troubleshooting**: What are some common problems encountered by the system? Can these be detected and automatically fixed?

A tool that provides insight into these questions would have a profound in fact on the way the system is configured and managed. Being able to answer these questions would allow system administrators to accurately troubleshoot problems and hopefully prevent future ones from occurring.

The goal of this project is to build a framework for monitoring, analyzing, and visualizing distributed computing clusters. In particular, we plan on using this tool on the new BGSC and the faculty mentor's DPL cluster by deploying it and testing it on real cluster activity.

2 Methodology

In order to accomplish this goal, we will implement a variety of tools. The first layer is an application that runs on each machine and collects information such as currently running programs, CPU and memory usage, etc. on that particular machine. This information will be sent to another application that will process this data into a database to allow for efficient querying of information. Another component will be a web portal that will allow for interactive access to the data and for real-time visualization of the system via graphs and diagrams generated on-the-fly. These visualizations will be displayed above the faculty mentor's office on a larger LCD monitor for viewing by the general public.

To implement this project, we will be using the Python [1] programming language along with the Tornado [2] framework. A prototype of the data collection and aggregation component current exists, but needs to be fine-tuned and optimized. What remains is the implementation of a data querying language and the generation of dynamic visualizations.

3 Context and Significance

In terms of the overall concept, Computer Science has been working on various self-healing systems [3] for quite some time. Unfortunately, due to the complexity of the problem, many such systems are tied to their particular hardware configuration and are not generally applicable to different systems. Because of this, there are many different applications for collecting data from multiple machines [4, 6, 5] and visualizing the aggregated information, but very few for actually analyzing the information in-depth. Instead, these systems rely on human operators to pour through large amounts of data to determine the state of the system.

Our system will be different in the way it approaches the aggregation of data and the querying of information. In former, we will use a RESTful [7] architecture to allow for extensibility and interoperability with external tools. This will enable easy development of the web interface and will allow others to directly access our aggregated data to form additional new analytical tools. In the latter portion, we will develop a fast and efficient time-series key-value storage system that will allow us to examine the state of the system at different time intervals. This will be a departure from traditional SQL databases, and more akin to the new trend in NoSQL[8] data storage solutions.

4 History

This project has not been funded by ORSP previously. The faculty member has written a partial prototype of the proposed system while working with an independent study student last Spring. The goal of this project is to complete the system by adding the described components and to put it into real world use.

5 Dissemination

We hope to present the results of our research in the following venues:

1. A paper at the Midwest Computing and Instruction Symposium (MICS).

2. A poster or presentation at the Celebration of Excellence in Research and Creative Activity (CERCA).

If appropriate, we will also submit a poster or paper to other Computer Science venues. Additionally, we plan on open sourcing our software so that others can utilize our tools. Finally, we plan on using this platform on existing computing clusters at UW-Eau Claire.

References

[1] Python. https://www.python.org/.

[2] Tornado. http://www.tornadoweb.org/.

[3] Ghosh, Debanjan, et al. “Self-healing systemssurvey and synthesis.” Decision Support Systems 42.4 (2007): 2164-2185.

[4] Sacerdoti, Federico D., et al. ”Wide area cluster monitoring with ganglia.” Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on. IEEE, 2003.

[5] System statistics collection daemon. https://collectd.org/.

[6] Imamagic, Emir, and Dobrisa Dobrenic. ”Grid infrastructure monitoring system based on Nagios.” Proceedings of the 2007 workshop on Grid monitoring. ACM, 2007.

[7] C Pautasso, O Zimmermann, F Leymann. “Restful web services vs. big’web services: making the right architectural decision”. Proceedings of the 17th international conference on World Wide Web, 805-814.

[8] Stonebraker, Michael. ”SQL databases v. NoSQL databases.” Communications of the ACM 53.4 (2010): 10-11.