

Statement of Research for Taliver Heath

Research on the systems side of Computer Science straddles the line between science and engineering. Both aspects are important, so neither side should be ignored in favor of the other. In my research, I have sought to discover the underlying nature of the mechanisms responsible for a given behavior. Using this understanding, I could create a simple model of the process whose underlying science is hopefully applicable across a wide range of technologies. With this model, I could re-engineer the existing system to optimize its performance, and validate my changes with measurements on real systems when available. I have used this technique in a wide variety of applications, ranging from failure analysis to modeling energy usage of clusters on a per-request basis.

Failures in Clusters

Computers fail. If the computer in question is in a cluster of servers, the failure can result in a lost transaction or sale, which translates directly into lost income for the company in question. In my first research project, we attempted to measure the distribution of these failures in a clustered computing environment.

Without attempting to find a root cause behind the failures of the systems, we found the machines obeyed a certain failure distribution in time, which we could model with a small number of parameters. Using this model, we were then able to propose a strategy that used this behavior to increase the availability of a cluster of machines. The strategy was general enough so that for any failure distribution other than a pure Poisson, it could be deployed and reduce the perceived number of failures from the client side. This work was validated using a series of failure traces collected from a variety of clusters, and in [6] we found that for an approximate 30% increase in cluster capacity, 70% of failures could be masked from the end user.

Disk Power Consumption

With this higher-level system analysis complete, I turned my focus to modeling energy usage in disk devices. In a laptop computer, battery life is very important. To extend the battery life of a computer, either a more powerful battery is needed, or the energy necessary to complete a task must be reduced. In many instances, hard drive usage accounts for a large portion of the energy expended to complete a task. We examined application use of disk drives, and studied a very large range of policies for reducing energy usage of the disk by exploiting idle times.

When a disk is idle, it can be sent into a state that uses less power, which in turn saves energy. There are penalties associated with this, however. Going to a lower state and then back up to an active state takes time, which may cause an increase in latency. Resuming the higher active state often has a peak power greater than the average power of the higher state. It follows that if the disk is not in the lower state for a long enough time, switching states can lead to an increase in required energy.

Since disks need to be in lower states to save energy, we needed to ensure that the idle times in applications would be long enough to enjoy the energy savings possible by changing states. We came up with a simple buffer optimization that increased idle

times, and simultaneously made those idle times of predictable lengths. This idle-time predictability lead to two useful results: the first was that we knew exactly the optimal state to send the device to; the second was we knew when the device should be preemptively activated. This preemptive activation hid the latencies associated with resuming the high energy state.

Using these techniques, we saw up to 89% reductions in energy consumption, with little or no degradation in performance across a variety of applications. The policies explored are detailed in [3]. We were able to use a compiler to automate the buffer transformation, which is detailed in [4] and [5]. Other groups have since proposed using similar techniques in other devices as well[10].

Power in Clusters

While conserving energy in a laptop environment intuitively makes sense, conserving energy in a clustered environment often does not seem as important. However, large computing centers draw tremendous amounts of power not only for running the computing infrastructure, but for cooling the infrastructure as well. This power usage can be up to 25% of the operating cost. Furthermore, computational capacity is often based on peak load expectations, which can cause up to 89% of the resources to be unnecessary the majority of the time [7].

In previous work[9], we had examined how to reduce the energy used by a web server running on a homogeneous cluster. For homogeneous servers, turning off as many servers as possible while still serving the requested load was the most efficient method to save energy. In a heterogeneous clustered environment, this may not be the case.

Real-life server clusters are almost invariably heterogeneous in terms of the performance, capacity, and power consumption of their hardware components. For example, the Teoma/AskJeeves search engine is supported by a highly heterogeneous server cluster with thousands of nodes. In fact, the different services involved in the search engine, such as the indexing and Web services, are themselves supported by heterogeneous nodes. The heterogeneity comes from nodes with different processor and network interface speeds, as well as different numbers of processors and memory sizes [11].

The reason for the heterogeneity of real-life server clusters is simple and at least three-fold: (1) failed or misbehaving components are usually replaced with different (more powerful) ones, as cost/performance ratios for off-the-shelf components keep falling; (2) any necessary increases in performance or capacity, due to expected increases in offered load, are also usually made with more powerful components than those of the existing cluster; and (3) traditional, PC-style nodes are slowly being replaced by collections of single-board “blade” nodes to save physical data center space and ease management. The combination of traditional and blade nodes makes for highly heterogeneous clusters, since some blade systems exploit laptop technology to consume significantly less energy than traditional computers. In essence, server clusters are only homogeneous (if at all) when first installed.

We can take advantage of this heterogeneity due to the fact that our load is heterogeneous in nature as well. Some requests require more disk usage. Some requests need

only small memory accesses. Other requests, for example dynamic HTTP requests, require heavy CPU utilization. We examined a trace composed of dynamic and static requests, and modeled the machines in our cluster to determine power and throughput of proposed configurations. We then used a simulated annealing algorithm to find the optimal configurations for a given request load. Using this generated table of request loads and configurations, we set the system to respond to load changes and measured the savings in energy over a two hour run using an accelerated trace. We found we could save 41% more energy than the heterogeneous technique, with a negligible loss rate.

Current Research

One of the reasons that energy and power are important in server environments is due to the heat that they produce. This heat must be removed using cooling equipment, and if not removed, can quickly lead to failure of the individual servers or the entire cluster. Because of this concern, the cooling requirements for server clusters is often provisioned for maximum utilization of the cluster. If a clustered environment were temperature-aware, the designers may be able to provision for a smaller cooling requirement.

We are currently investigating a per-system approach to thermal management in a clustered environment. A per-component model might work well with the previous energy-usage model I created, but might prove too detailed. For this reason we are also investigating a “heat per request type”-model. This should have the advantage of still fitting well into the current model, while not being too complex to measure and instantiate.

From these models and implementations, I plan on investigating policies which explore the differences between scheduling for power, scheduling for energy, scheduling for performance, and scheduling for temperature. Previous work has shown that policies for energy can lead to different decisions than policies for power [8]. Preliminary results indicate that policies designed to reduce temperature may not necessarily be energy efficient.

Future Directions

In each of prior cases, I looked for a model of behavior and used it to modify an existing system. The models not only provided a deeper understanding of the systems’ characteristics, but also allowed for broad parameter investigations. The modified systems benefited from the models and demonstrate the importance of analytical skills in systems research.

Eventually, I would like to investigate areas involving the video gaming community, which is very important to the computing industry: in 2003 it was responsible for over 7 billion dollars in sales[1]; in 2002, EverQuest would have ranked as the 77th largest economy on the planet, putting it just after Russia[2]; the average age of people that bought games in 2003 was 36; forty-three percent of game players play online one hour or more a week; in 2002, 15% of respondents to a PC magazine survey claimed that Internet gaming is the reason they purchased a broadband connection.

This area has lots of benefits for study. There are generally many people willing to try out game servers, and game servers allow for real time tweaking and monitoring of player interactions as well. Many collaborations are possible involving most areas of systems and applied computer science research, from architecture to wireless networking. These game servers have requirements low latencies, high quality of service, resource accounting, and abuse detections that other fields of research in systems research could benefit from. Since most of the fruits of the research would help in developing networked games, private funding opportunities might be available as well.

By modeling the user interactions with the game systems and the states inside the game server in question, we could better understand where to apply optimizations for improving performance, and also give a measure of user satisfaction with the game server in question. With these models, we could then explore various policies for adjusting quality of service on an individual level, allowing a better assignment of available resources. These policies could be extended beyond gaming systems as well to any system with similar restrictions on bandwidth and latency.

References

- [1] Entertainment Software Association. Top ten industry facts. <http://www.theesa.com/pressroom.html>.
- [2] David Becker. Everquest spins its own economy. *news.com.com*, January 2002.
- [3] T. Heath, E. Pinheiro, and R. Bianchini. Application-Supported Device Management for Energy and Performance. In *Proceedings of the Workshop on Power-Aware Computer Systems*, February 2002.
- [4] T. Heath, E. Pinheiro, J. Hom, U. Kremer, and R. Bianchini. Code transformations for energy-efficient device management. *IEEE Transactions on Computers*, 53(8), August 2004.
- [5] T. Heath, E. Pinheiro, J. Hom, U. Kremer, and R. Bianchini. Application Transformations for Energy and Performance-Aware Device Management. In *Proceedings of the 11th International Conference on Parallel Architectures and Compilation Techniques*, September 2002. Best student paper award.
- [6] Taliver Heath, Richard P. Martin, and Thu D. Nguyen. Improving Cluster Availability Using Workstation Validation. In *Proceedings of the ACM Sigmetrics Conference*, June 2002.
- [7] A. Iyengar, J. Challenger, D. Dias, and P. Dantzig. High performance web site design techniques. *Internet Computing, IEEE*, 4(2):17–26, May/Apr 2000.
- [8] U. Kremer. *Low-Power Electronics Design*. CRC Press, 2004.
- [9] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath. Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems. In *Proceedings of the International Workshop on Compilers and Operating Systems for Low Power*, September 2001.

- [10] Haijin Yan, Rupa Krishnan, Scott A. Watterson, David K. Lowenthal, Kang Li, , and Larry L. Peterson. Client-centered energy and delay analysis for tcp downloads. In *12th IEEE International Workshop on Quality of Service*, June 2004.
- [11] Tao Yang. Personal communication. October 2003.