Heuristics Considered Harmful: Using Mathematical Optimization for Resource Management in Distributed Systems

Emin Gün Sirer
Dept. of Computer Science, Cornell University, Ithaca, NY 14853
egs@cs.cornell.edu

ABSTRACT

Distributed systems often pose difficult to resolve resource management problems. These problems typically involve the partitioning of a critical resource, such as bandwidth, storage, or computational elements, between competing tasks. Traditionally, such problems are resolved using custom, domain-specific heuristics. Yet heuristics are neither robust to fluctuations in load characteristics nor do they enable the system designer to reason definitively about the emergent properties of the system after deployment.

In this paper, we argue for a more principled approach to resource management in distributed systems. Namely, we propose that resource allocation problems are ideally suited for mathematical optimization. We outline a general approach based on analytical modeling, optimization, and practical implementation. We describe how we have applied this technique to several diverse domains, to yield qualitative improvements in performance and achieve strong guarantees.

1. INTRODUCTION

At the core of many distributed systems lies a difficult resource management problem. These problems typically involve the partitioning of a critical resource, such as bandwidth, storage, or computational elements, between competing tasks. Such resource tradeoffs are encountered in content distribution networks, resource management in the GRID [4], cache management in the domain name system, data distributions in large-scale storage systems, high-performance publish-subscribe systems, as well as many other infrastructure services where performance is a function of resources and resources are limited.

Distributed system designers often resort to *ad hoc heuristics* to address difficult resource allocation problems. A particularly common technique is to use locally managed, independent resource managers that follow simple strategies at each node with no coordination. For instance, a content distribution network may use an independent least recently-used cache manager in each node. These heuristics are typically validated using limited traces collected from the field. While some heuristics might fit some traces well, heuristic techniques are neither robust to fluctuations in load characteristics nor do they enable the system designer to reason definitively about the emergent properties of the system after deployment. Heuristics may achieve significant performance gains on a given trace – after all, they are effectively a way of fitting a function to a given workload, and there is no reason, besides the designer's internal motivation to keep the system simple, why the fit cannot incorporate the entire workload and thus be perfect. The question to ask of heuristics-based approaches, then, is not how well they perform, but for how wide-ranging a set of workloads they perform well. And this characterization is often difficult to formalize; heuristics may work, but there is often no telling when they will stop working.

In this paper, we argue for a different, more principled approach to resource management in distributed systems. The main thesis of this paper is that resource allocation problems are ideally suited for mathematical optimization. Unlike the common use of the term *optimization* in computer science to refer to incremental program transformations designed to improve performance, we are referring to *mathematical optimization*, a process whose goal is to find the true optimal point in a given function, subject to optional constraints.

We outline a general approach to resolving resource allocation problems in distributed systems through mathematical optimization. The pillars of this approach are analytical modeling to capture the core tradeoff, analytical and numerical techniques for determining the optimal solution, and limited runtime aggregation for estimating parameters in the solution. This technique is quite general and we have applied it to diverse problems ranging from optimal failure detection to the design of high-performance, scalable infrastructure services such as content distribution networks, publish-subscribe systems, and a safety net and replacement for the domain name service. We describe how we have applied this technique to several diverse domains, to yield qualitative improvements in the performance of distributed systems and to achieve strong average-case guarantees in the presence of dynamic changes in workloads.

2. APPROACH

The approach we advocate consists of four steps:

Capture the tradeoff. The first step in finding the optimal allotment of resources to competing tasks is to analytically capture the relationship between the amount of resources awarded and the performance achieved as a result. This process requires an articulation of the performance metrics of interest, and a formulation of the metrics as a function of the resources. We call this the performance equation.

Express the constraints. The second step in determining the optimal resource allocation is to capture the constraints on the critical resources. There are typically two kinds of constraints: *resource constraints* and *performance targets*. Resource constraints arise naturally whenever a finite resource is being partitioned. An example of a resource constraint is that the sum of all bandwidths allocated to competing processes in a CDN must not exceed line speed. Resource constraints force the system to achieve the best possible performance while remaining within an upper bound on resource consumption. In contrast, performance targets pose a lower bound on the performance equation that the system must achieve. For instance, a publish-subscribe system might want to ensure that the average time to propagate changes is below a particular threshold.

Such performance targets force the system to achieve the desired performance level while minimizing resource consumption.

Solve. Having expressed the performance equation and the constraints, the system can now be solved. A technique we have used successfully that has not yet been widely adopted in distributed systems is the use of Lagrange multipliers. A system with performance equation f and constraint equation g can be solved by introducing Lagrange multiplier λ, and solving for $\nabla f = \lambda \nabla g$. Often, this solution will require differentiation with respect to independent resource allocations x_i. The system will typically be analytically tractable if the system of equations is independent in x_i. Analytical solutions are desirable because they lead to formulas that can be evaluated efficiently. In cases where analytical solutions are not tractable, numerical techniques can be used to solve for λ and x_i^*, the optimal resource allotment for each competing task.

Implement. Translating the optimal solution achieved in the previous step into a concrete implementation is often non-trivial. The solutions, whether analytical or numerical, require values for parameters to be determined so the system of equations can be solved. For instance, optimal bandwidth allocation for object replication in a content distribution network will typically require a relative ranking of objects by popularity. Determining this order is difficult; done naively, it requires global information. At this stage, various domain-specific design considerations may be used in order to reduce the amount of communication, and to replace global computations with limited, local data aggregation over existing channels. For instance, the structure provided by a distributed hash table can be used to simplify the task of propagating such aggregate information on the relative popularity of objects.

3. APPLICATIONS

We have applied the preceding approach to the construction of three infrastructure services.

CoDoNS: CoDoNS [6,5] is a high-performance, failure-resilient, and scalable name service for the Internet. It serves as both an alternative and a safety-net for the legacy Domain Name System (DNS). The use of mathematical optimization enables CoDoNS to provide strong optimality guarantees; specifically, the system can achieve O(1) lookups on top of a O(log N) peer-to-peer overlay. The result is surprising as heavy-tailed distributions, which occur frequently in distributed systems such as DNS [2], web [1], and RSS [3], were long-thought to be difficult to address [2, 1, 8]. The mathematical optimization framework driving CoDoNs automatically adjusts the system respond to sudden changes in object popularity, as in the so called "slashdot effect."

CobWeb: CobWeb is an open-access content distribution network that can delivers web pages quickly and efficiently. CobWeb operates as a ring of cooperative proxy servers, each of which is capable of serving any HTTP request. When web objects are requested, they are fetched from their origin servers and inserted into the system. Through an analysis of web object popularity, size, and update rate, CobWeb then computes an optimal replication strategy for each object to provide low lookup latency while minimizing overhead.

Corona: CorONA is a high-performance publish-subscribe system for quick and efficient dissemination of web micronews. It is a replacement for, and is backwards compatible with, RSS. The core optimization Corona performs differs from CoDoNS and CobWeb in that the constraints it addresses are not flat constants but vary with the client node; specifically, the system places no more load on the system as what plain RSS would place if it were used instead. The solution then uses a mathematical optimization framework for trading off bandwidth for performance, to provide low-latency news dissemination while ensuring that the load placed on news providers does not exceed a desired limit. This avoids the load and sticky-traffic problems that RSS faces.

Overall, the use of a principled resource allocation framework in these systems provides a strong level of confidence of the robustness of the system. And, as we have shown with Beehive [5], formally capturing and optimizing for the central resource tradeoff enables qualitative improvements in system performance.

4. SUMMARY

The principled approach advocated in this paper can be applied to many other systems problems based on tradeoffs in the presence of constraints. For instance, even ubiquitously deployed, simple basic building block services, such as failure detectors, can be improved significantly through this approach. A failure detector is simply a bandwidth allocator whose goal is to minimize failure detection time without exceeding a given bandwidth budget. Early simulations based on data from PlanetLab indicate that mathematical optimization can improve failure detection latencies by a factor of two without increasing bandwidth consumption. Similar improvements are possible for energy consumption in sensor networks, bandwidth consumption in software distribution, and even for processing overhead in a secure operating system through the judicious selection of optimal chunk size in data transfers [7].

The use of mathematical optimization in system design enables strong performance guarantees and provides assurance under a wide, well-characterized set of workloads. We call upon system designers to abolish unreliable heuristics in favor of a more principled approach to resolving difficult resource management problems.

Acknowledgments

I would like to thank Venugopalan Ramasubramanian, Yee Jiun Song, Ryan Peterson, Bernard Wong, Kelvin So, Dan Williams and Alan Shieh for designing, building and deploying systems based on the approach described in this paper. This work was supported in part by National Science Foundation Grants 0430161 and CCF-0424422 (TRUST).

5. REFERENCES

[1] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proc. of IEEE International Conference on Computer Communications*, New York, NY, Mar. 1999.

[2] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and Effectiveness of Caching. In *Proc. of SIGCOMM IMW*, San Francisco, CA, Nov. 2001.

[3] H. Liu and E. G. Sirer. A Measurement Study of RSS, A Publish-Subscribe System for Web Micronews. In *Proc. of the IMC*, Berkeley, CA, Oct. 2005.

[4] J. Nabrzyski, J. M. Schopf, and J. Weglarz, editors. *Grid Resource Management: State of the Art and Future Trends*. Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[5] V. Ramasubramanian and E. G. Sirer. Beehive: Exploiting Power Law Query Distributions for 0(1) Lookup Performance in Peer-to-Peer Overlays. In *Proc. of NSDI*, San Francisco, CA, Mar. 2004.

[6] V. Ramasubramanian and E. G. Sirer. The Design and Implementation of a Next Generation Name Service for the Internet. In *Proc. of SIGCOMM*, Portland, OR, Aug. 2004.

[7] D. Williams and E. G. Sirer. Optimal Parameter Selection for Efficient Memory Integrity Verification Using Merkle Hash Trees. In *Proc. of NCA Trusted Network Computing Workshop*, Boston, MA, Aug. 2004.

[8] A. Wolman, G. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. Levy. On the Scale and Performance of Cooperative Web Proxy Caching. In *Proc. of SOSP*, Kiawah Island, CA, Dec. 1999.