Using attribute-managed storage to achieve QoS

E. Borowsky, R. Golding, A. Merchant, L. Schreier,
E. Shriver[*]*, M. Spasojevic*[**]*, and J. Wilkes*
Hewlett-Packard Laboratories

Abstract

Specification of storage systems by means of user-oriented Quality-of-Service attributes is the key to ease of use and efficient resource utilization. Attribute-managed storage systems hide details of the underlying storage systems through *virtual store* abstractions—units of storage with quality of service guarantees. The mapping of virtual stores onto physical storage devices can be optimized to achieve high level goals such as balancing system performance against total system cost. We demonstrate the feasibility of this approach with a prototype matching engine called Forum.

Keywords

storage systems, attribute-managed storage, network-attached storage, quality of service

1 INTRODUCTION

The configuration and management of large quantities of on-line storage is a non-trivial task—yet it is central to the functioning of most operating system services. The difficulties inherent in the problem are compounded by sheer scale—tracking the thousands of physical and logical devices required to support a few tens of TB of data.

Consider first the problem of initial configuration of storage devices so that performance and availability goals are met. Now consider configuration with performance guarantees in the face of a workload that is constantly shifting, and a storage pool that is changing as new devices are added and obsolete or defective ones removed. Compound with this the desire to share the storage across multiple computer systems with nearly arbitrary interconnection topologies via storage fabrics like Fibre Channel. Finally, the introduction of network-attached storage devices will only exacerbate the problem.

Things have reached the point where the cost of managing a device is several times the purchase cost. The planning for a medium-scale (few TB) installation can require many months. At the same time, these problems represent a significant opportunity: storage hardware was about a $50b business in 1995, and about 25% of information technology budget expenditures (Network Storage Conference, 1996). Improving the way storage is used and managed could be a huge win: for users, for system managers and administrators, and for computer system vendors.

2 ATTRIBUTE-MANAGED STORAGE

Most existing approaches to storage management operate at too low a level: they require people to allocate and configure disks or disk arrays to particular pieces of work, but provide little help to them in doing so. For example, logical volume managers [Chang90] provide a number of low-level mechanisms to allow multiple disks to be grouped together

[*] Elizabeth Shriver was supported by the NFS grant CCR-9504175.

[**] Contact author: Mirjana Spasojevic, Hewlett-Packard Labs, MS 1U-13, 1501 Page Mill Road, Palo Alto, Ca 94304, USA. E-mail: mirjana@hpl.hp.com.

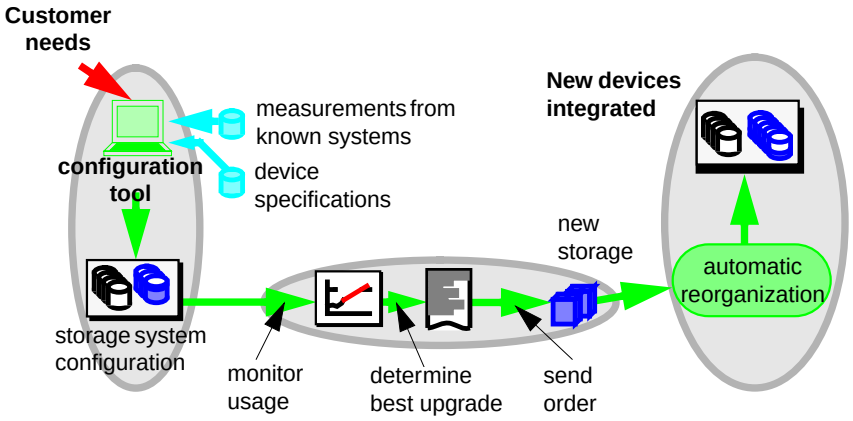

Figure 1. The life-cycle of the self-configuring, self-managing storage system.

in different ways, but provide no additional help in predicting performance effects, or determining the best data layout to use.

Instead, our approach is to abstract the gory details of the underlying storage system by having the storage system present a *virtual store* abstraction. Each virtual store represents a unit of storage and can hold a large-scale data object such as a file system, relational database table, or a single large file. It has a set of Quality-of-Service (QoS) attributes such as size (capacity), performance capabilities, and resilience (a set of availability and reliability metrics) which satisfy requirements of the application workload. In turn, virtual stores are mapped onto physical storage devices, each having a set of capability attributes such as capacity, performance, and availability.

The QoS attributes of virtual stores are high-level storage system goals, or requirements: they specify *what* the storage system is to achieve, *not how* it is to do so. This allows us to automate the mapping while taking into account multiple simultaneous needs like bandwidth, I/O rate, capacity, or system cost.

There are several advantages of the attribute-based approach. For example:

- automated mappings can be made in minutes rather than the weeks needed for manual configuration;
- because the storage system knows the goals of its clients, it can adapt to changes in physical configurations automatically;
- the optimization engine can be directed to meet different goals, such as how to balance system performance against total system cost;
- the same optimization technology that can generate initial assignments of workload to devices can be used to ask “what if?” questions to explore the effects of future changes in configuration or load;
- once the system is running, the QoS goals for the virtual stores can be monitored, and adjustments made in the assignment.

Figure 1 shows several ways in which we envision this technology will be applied.

3 THE FORUM PROJECT

We are actively developing technology to validate the concept of attribute-managed storage. We have developed a prototype assignment engine (or solver) called Forum.

We model the assignment of workload to devices as a multiple-constraint, multiple-knapsack optimization problem. Treating this as a general optimization problem, rather than focusing specifically on the storage-assignment task, has generated great flexibility in the prototype: for example, we can easily add system goals such as minimizing the expected reorganization cost as new workloads are added.

The main Forum solver components are:

- a set of device performance models for utilization, throughput, and response time based on our prior work [Ruemmler94, Wilkes95, Wilkes96];
- a set of QoS constraints that express things like “you cannot put more data on a device than it can hold”, and “the device utilization cannot be greater than 1”;
- sets of device parameters that describe the capabilities of available devices;
- a workload model which captures variability in requirements of an application;
- a search engine, which explores the potential space of assignments.

As the solver runs, it uses the device models to decide whether or not an assignment can be made. Although the optimization problem is NP-hard, we have been able to adopt several heuristics from the literature [Toyoda75]. We have found that remarkably simple ones produce pretty good results: the solver is capable of assigning several thousand objects to hundreds of devices in a few minutes.

In addition to dealing with performance goals, the solver can handle availability requirements [Wilkes91]. An additional component, called Corbel (written for us by Khalil Amiri of CMU), builds and solves Markov models to predict the availability and reliability properties of all the possible configurations of the storage devices of interest.

4 VALIDATION

Although our prototype assignment engine claims to make assignments that support probabilistic QoS guarantees, it is essential to compare our models against the real world to see how they do in practice. To this end we are performing a set of validation experiments. We take both synthetic and more realistic workloads (such as database benchmarks), and compare the predicted performance on the assignment emitted by the solver with the actual behavior of the running workload. Our goal is to be slightly conservative: we want to keep the target system well-utilized, but for its performance to remain inside the bounds predicted by the solver which, in turn, must fall within the bounds required by the workload.

We have operated under the premise that the more information is provided to the optimization engine about a workload, the better solution it can provide. However, we also want the minimal set of attributes required to generate good solutions. Figure 2 shows the results from one of our tests: allowing for closed arrival processes in the internal performance model makes for a significantly more accurate estimate of system performance.

The primary validation test has centered around a decision support database benchmark. We first configured our system using the best expert advice we could find, and ran the benchmark. The performance measurements were then used as requirements for

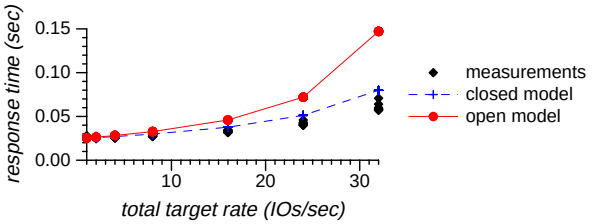

Figure 2. Results of the solver validation for a set of workloads assigned to an HP97560-300 disk.

assigning the same workload to the same system, with the goal of minimizing the number of devices used. The resulting assignment used fewer devices, and system performance matched the requirements–the benchmark execution time was within 2%.

5 CURRENT WORK

Our current work lies in several activities:

- enhancing the workload model to include interactions between individual access streams and the data objects that they access;
- extending the solver with different algorithms;
- continuing to improve the model of device performance, adding models of disk arrays and using a composite model of device performance that accounts for the effects of cache and request queueing policies; and
- validation across a broader range of realistic workloads.

In the future, we plan to embed the Forum smarts in network-attached storage devices (see http://www.hpl.hp.com/SSP/NASD for more information), which will provide us with ways to support performance and security guarantees in shared storage systems. The way in which this is done is going to have a large impact on future operating systems and the way they access their storage.

REFERENCES

[Chang90] A. Chang, M. F. Mergen, R. K. Rader, J. A. Roberts, and S. L. Porter. Evolution of storage facilities in AIX Version 3 for RISC System/6000 processors. IBM Journal of Research and Development, 34(1):105–10, January 1990.

[Clegg86] Frederick W. Clegg, Gary Shiu-Fan Ho, Steven R. Kusmer, and John R. Sontag. The HP-UX operating system on HP Precision Architecture computers. Hewlett-Packard Journal, 37(12):4–22, December 1986.

[Gelb89] J. P. Gelb. System managed storage. IBM Systems Journal, 28(1):77–103, 1989.

[Ruemmler94] Chris Ruemmler and John Wilkes. An introduction to disk drive modeling. IEEE Computer, 27(3):17–28, March 1994.

[Toyoda75] Y. Toyoda. A simplified algorithm for obtaining approximate solutions to zero-one programming problems. Management Science, 21(12):1417–27, August 1975.

[Wilkes91] John Wilkes and Raymie Stata. Specifying data availability in multi-device file systems. Position paper for 4th ACM-SIGOPS European Workshop (Bologna, 3–5 September 1990). Published as Operating Systems Review, 25(1):56–9, January 1991.

[Wilkes95] John Wilkes. The Pantheon storage-system simulator. Technical Report HPL–SSP–95–14. Storage Systems Program, Hewlett-Packard Laboratories, Palo Alto, CA, 29 December 1995.

[Wilkes96] John Wilkes, Richard Golding, Carl Staelin, and Tim Sullivan. The HP AutoRAID hierarchical storage system. ACM Transactions on Computer Systems, 14(1):108–36, February 1996.