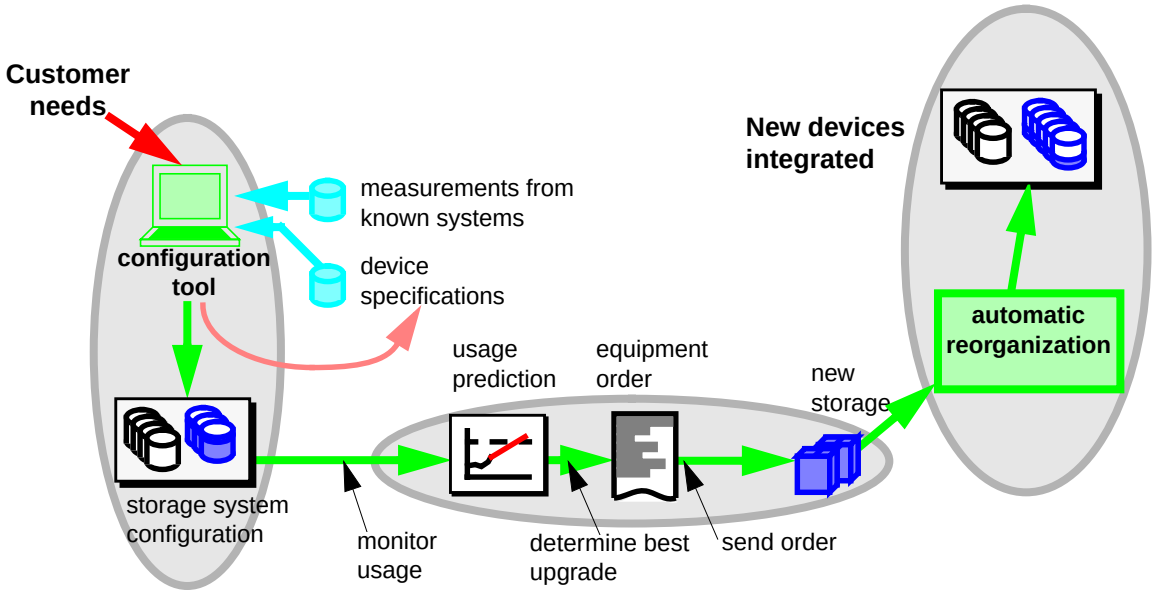
Eliminating storage headaches through self-management

Elizabeth Borowsky, Richard Golding, Arif Merchant, Elizabeth Shriver, Mirjana Spasojevic, and John Wilkes
Storage Systems Program, Hewlett-Packard Labs



The life-cycle of the self-configuring, self-managing storage system starts with a model of the system workload, based on which the initial configuration is built. The system monitors changes in the workload, not just to order more devices, but also to guide load balancing and rearrangement along the way.

The complexity of managing storage is growing rapidly as the storage attached to distributed computing systems gets larger. This has reached the point where the cost of managing a device is several times the purchase cost, and the planning for a medium-to-large installation can require many months. Many system managers will try just about anything to avoid adding new devices to their system because of the reconfiguration effort.

We expect these trends to get worse. The introduction of Fibre Channel back-end storage networks and network-attached storage devices (NASDs) turns individual storage units into shared resources. An ever-increasing number of data-hungry applications introduce new demands, such as guaranteed quality of service, on the allocation and balancing of existing resources.

We believe that the solution is a self-managing system model, in which objects (such as files, database tables, or file systems) are stored in a shared storage utility and the details of low-level object-to-device assignment are managed invisibly by the system.

Each object placed in the store is given a set of capacity, performance, reliability, and cost attributes. Likewise, each device has a set of capability attributes. The system users may also specify goals, such as how to balance system performance against total system cost. The system uses the attributes to build the assignment of objects to devices so that the goals are met, rather than having a user determine that object X will be striped across disk array Y on server Z. Once the system is running, it monitors the actual uses of the objects for changes in behavior, and reorganizes the assignment from time to time to maintain good performance.

We model the assignment of workload to devices as a multiple-knapsack, multiple-constraint optimization problem. Treating this as a general optimization problem, rather than focusing specifically on the object-assignment task, has generated great flexibility in the prototype: for example, we are investigating the value of system goals such as minimizing the expected reorganization cost as new objects of a particular kind are added. Our approach allows this to be intermixed with simpler goals like minimizing system cost.

Our prototype assignment engine, Forum, is capable of assigning several thousand objects to hundreds of devices in a few minutes, and often the result appears significantly better than hand-generated assignments for the same workload. We are currently validating these results using database and Web server benchmarks. This involves specifying benchmark workload requirements and real device capabilities, generating the assignment of objects to devices using the Forum engine, configuring a real system on which to run the benchmark, and finally running the benchmark to ensure that the results were what the Forum engine predicted.We are continuing to extend the prototype by modeling:

- interactions between individual application access streams and the data objects they access;
- incremental expansion of the workload and object migration cost;
- complex customer preferences between cost, performance, and system flexibility; and
- complex devices to represent disk arrays as well as single disks with caches and request queues.

For more information: http://www.hpl.hp.com/SSP/