MIRAGE: Storage provisioning in large data centers using balanced component utilizations

ABSTRACT

This paper presents MIRAGE, an architecture for data center storage provisioning that takes the approach of maintaining storage services for applications by ensuring well-balanced utilizations in all internal components of the storage infrastructure. We implemented MIRAGE on our local storage infrastructure and observed the sensitivity of the MIRAGE load-balancing algorithm to a combination of performance and heterogeneity skews. We also evaluated MIRAGE by deploying it on a financial data center. We reduced the service times of resource-constrained storage pools by an average of 68%.

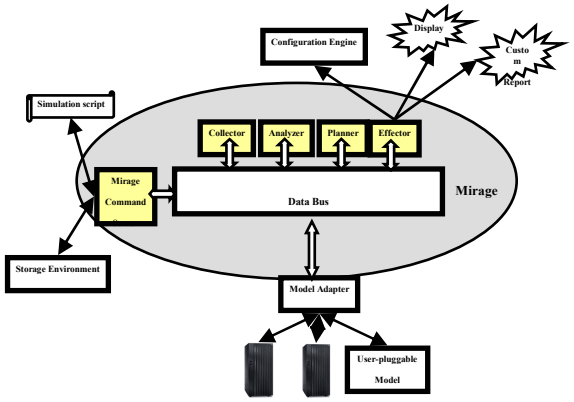
1. Introduction

The promise of storage area networks (SAN) was to separate storage from application servers and to develop storage as a first-class entity that would provide services to applications. However today's storage area networks are very complex because of both scale and heterogeneity of its components. A typical SAN could comprise of thousands of application servers, tens of thousands of storage volumes and a few hundred thousand data paths between the servers and the storage volumes. The growth of storage data is now estimated at around 50% per year [1] and challenges the financial resources of an organization to keep up with the demand. Thus data center administrators are increasingly using storage provisioning techniques of migration and consolidation to optimize capacity allocation instead of over-provisioning. It is not possible for a human or a set of human administrators to take decisions about complex SAN using manual tools, neither is it possible to guarantee that the decisions will achieve storage service requirements.

While it is tempting to argue that well-balanced and low component utilizations do not necessarily guarantee storage service requirements, it is worthwhile to note that the vast majority of storage quality of service requirements are punitive in nature and caused by utilization threshold violations by a storage infrastructure component. Furthermore, today's applications are complex and resemble logic circuit boards and it is difficult to derive non-punitive storage quality of service requirements for these applications. MIRAGE is complementary to traditional approaches to storage service maintenance that observe application service times and dynamically tune resource allocations to meet storage service requirements.

2. Architecture

MIRAGE can be described as a modular analytic engine that gathers data from the storage environment and generates configuration actions, reports, and display actions. The goal of the MIRAGE analytic engine is to provide long-term decision making support for three storage provisioning tasks: allocation, migration and consolidation. The figure below details the component architecture of MIRAGE.



The Collector component gathers configuration and performance data from the data bus on a periodic basis and stores the data in an internal repository. The Analyzer gathers performance traces and predicts the behavior of the traces into a pre-determined time into the future. The Analyzer interacts with the Model Adapter module that provides performance simulation for devices in the storage infrastructure. The Planner aggregates the utilizations and performs a graph-analysis to isolate resource constraints in the storage infrastructure. Following this, the Planner component uses a load-balancing algorithm to reallocate workloads in the storage infrastructure and generates several candidate plans. The Effector component is another plug-in module that allows the user to decide what to do with the candidate plans.

3. Load-balancing Algorithm

The load-balancing algorithm is central to all decision making in MIRAGE. The goal of the algorithm is to ensure that performance utilizations are balanced across all components in the storage infrastructure (SAN). Our paper proposes the use of component utilizations as metrics to extract the best application service times from the storage subsystem. In particular, the minimization objective considered is the sum of the mean and the standard deviation of the performance utilizations of storage components. Prior research [2] has studied the relationship between component utilizations and application service times and shown that the service times are monotonic with the component utilizations.

The output of the load-balancing algorithm is a set of migration tuples where each tuple is of the form: <V, SP, TP, C, B> where V is the volume to be migrated, SP is the source pool, TP is the target pool, C is the cost of migration and B is the benefit in migration.

Steps of Algorithm are:

1. The storage pools are ordered using a *pool ranking mechanism*
2. Select the pool with the highest rank as a candidate source pool.
3. Choose the pool with the lowest rank as a candidate target pool
4. The volumes in the source pool are ranked by *a volume ranking mechanism*

5. A candidate re-allocation plan is created where the highest ranked storage volume is selected for migration from the source storage pool to the target storage pool.
6. We add the candidate re-allocation plan to the target storage pool to the master list of re-allocations.
7. Re-compute the utilizations of the pools based on the re-allocation plan and apply *a stopping criterion.*

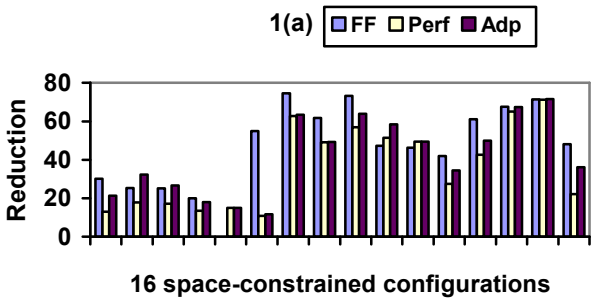
The goal of the pool ranking mechanism is to order the storage pools for consideration in a migration plan. The suitability of a storage pool for a migration plan is determined by two principal factors: the aggregate performance utilization of the storage pool and the space utilization of the storage pool. The performance capabilities of the storage pool are determined by the pool composition, so aggregate performance utilization is a function of *max* and *hierarchical* utilizations of the individual and shared components of the pool. Another important challenge in devising an adaptive scheme is to correctly infer the relative weight between performance and space as an incorrect inference might yield a sub-optimal migration that limits the possible reduction in the aggregate performance utilization of a storage pool. We develop an adaptive mechanism for inferring the relative weight between the aggregate performance and space utilization of a storage pool. In this algorithm, the rank of an individual storage pool p with an aggregate performance utilization U_p and space utilization S_p is determined by the equation:

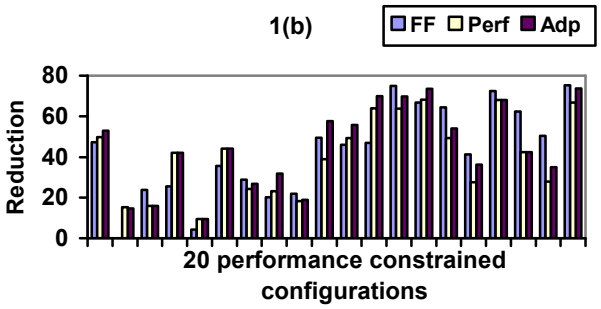
$$kU_p + (1-k)S_p, k = \frac{f_{Perf}(U_p)}{f_{Perf}(U_p) + f_{Space}(S_p)}$$

$$f_{Perf} = N\left(\frac{U_p - \overline{U}}{\partial(U)}\right)$$

$$f_{Space} = \frac{100}{100 - S_P} + N\left(\frac{S_p - \overline{S}}{\partial(S)}\right)$$

Here, the factors $\overline{U}$, $\overline{S}$ and $\partial(U)$, $\partial(S)$ represent the mean and standard deviation of the population of aggregate performance and space utilizations of storage pools in the storage infrastructure, and N is a linear scaling function to remove negative values. Figure 1(a) and (b). Results comparing the behavior of FirstFit(FF), Perf and Adaptive(Adp) algorithms for space and performance constrained configurations (x-axis) in terms of the reduction in the standard deviation of storage pool aggregate performance utilizations (y-axis). FirstFit results are not shown if there is an increase in the standard deviation.



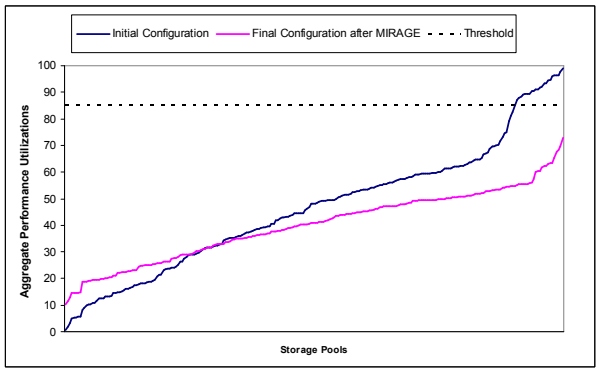


The goal of the volume ranking algorithm is to rank the storage volumes in the source storage pool as candidates for migration. There are two important considerations for selecting a volume: the size of the storage volume and the workload on the storage volume. The goal of the stopping criterion is to determine when the storage pools are balanced in terms of utilization. The stopping criterion is important as the cost of migrating storage volumes between storage pools is high and we need to consider the incremental benefit of a migration decision.

4. Data Center Study

We evaluated MIRAGE in a larger SAN in a financial service firm. We imported a week's worth of configuration and performance data from the storage management tools deployed in the storage infrastructure, and fed the imported data into MIRAGE. The SAN comprised of 6 storage controllers (4 IBM DS8000 and 2 IBM DS6000), 240 storage pools and 3678 storage volumes. The aggregate performance and space utilization of the storage pools had a mean of 50.13% and 51.58% respectively, with a standard deviation of 23.85% and 6.255% respectively. After the application of the migration plan, the service times for bottlenecked storage pools were reduced from 10.919ms to 3.575ms, a reduction of 68%.

Results show the sorted aggregate performance utilization (y-axis) of the 240 storage pools in a financial data center (x-axis) before and after MIRAGE was applied.



5. References

[1] R.Villars, What do I keep and how do I keep it?, IDC Directions Conference, April, 2007

[2] An Analytic performance model of disk arrays. Edward K. Lee and Randy H. Katz. SIGMETRICS Performance Evaluation Review, 1993.