Chapter 6

Conclusions and Future Work

6.1. Conclusions

In this thesis, I proposed a new approach to I/O performance evaluation—self-scaling benchmarks and predicted performance. This thesis' self-scaling benchmark yields information not only about performance but also about what workloads are appropriate for the system. The self-scaling benchmark scales automatically to current and future machines by adapting the workload to the system being tested. Further, it gives insight into the system's performance characteristics by revealing the performance dependencies for each of five workload parameters. I used the self-scaling benchmark on nine systems, ranging from a single-disk workstation to a striped-disk mini-supercomputer, and gained insight about appropriate blocksizes, file cache sizes, read/write behavior of the caches, and the benefits of sequentiality.

My first attempt to make a self-scaling benchmark showed that scaling all workload parameters was too extreme. However, scaling some parameters created a highly useful evaluation tool, as the modified self-scaling benchmark showed.

In the process of using the self-scaling benchmark, I found that many systems are not tailored to handle I/O-intensive workloads. Two systems that were designed specifically to be file servers (a Solbourne and an Auspex file server) crashed or deadlocked while running the self-scaling benchmark. Clearly, much more needs to be done to make systems more robust since more I/O-intensive workloads are forthcoming.

Predicted performance restores the ability to compare two machines on the same workload lost in the self-scaling benchmark. Further, it extends this ability to workloads that have not been measured by estimating performance based on the graphs from the self-scaling benchmark. I have shown that this prediction is far more accurate over a wide range of workloads than any single-point benchmark, both in absolute performance and relative performance of two systems. In addition, prediction using the *product of single parameter functions* model gives much more accurate performance estimates than does interpolation using even twice as many measured workload points.

The accuracy of my method of predicting performance shows that it is possible to characterize the performance of a wide range of workloads accurately with a few tens of workloads points. I hope that the added utility of being able to apply the performance graphs of the self-scaling benchmark to such a wide range of workloads will overcome the industry's tendency to focus on the performance of a single workload.

Self-scaling benchmarks and predicted performance could fundamentally affect how manufacturers and users view I/O evaluation in four critical ways. First, it condenses performance over a wide range of workloads into a few graphs. If manufacturers or standard performance groups (such as SPEC or computer magazines) were to publish such graphs, users could

use predicted performance to estimate, *without further measurements*, the I/O performance of their specific workloads. Ideally, manufacturers would go further and publish results from the self-scaling benchmark for many different system configurations, such as for different numbers of disks and main memory sizes, along with the cost of each configuration. With this information, users could calculate the price/performance ratios for each configuration *on their specific workloads* and make much more informed purchasing decisions.

Second, by taking advantage of self-scaling benchmark's ease of use, manufacturers could easily evaluate many I/O configurations. Instead of merely reporting performance for each I/O configuration on a few workloads, the self-scaling benchmark would report both the performance for many workloads and the I/O workloads that perform well under this configuration. Hence, manufacturers could better identify each product's target application area. Because the price of each configuration can easily be calculated, the price/performance of systems that match the users needs can also easily be calculated. This can help buyers make choices such as whether to purchase many small disks or a few large large disks, more memory and a larger file cache or faster disks, and so on.

Third, system developers could benefit by using the self-scaling benchmark to understand the effects of any hardware and software changes. Unlike traditional, single-point benchmarks, these effects would be shown over a wide range of workloads.

Lastly, the self-scaling benchmark introduces a new approach to scaling—scaling aspects of the workload to the changing performance characteristics of systems. This approach has the potential to successfully scale with tomorrow's faster and larger I/O systems.

6.2. Future Work

More work needs to be done in the general area of self-scaling benchmarks. In the future, I/O systems will support a wider range of workloads and stretch the limits of predicting performance. Perhaps more focal points will be needed to characterize performance over this

increased range of workloads. It would also be interesting to use the self-scaling benchmark on a wider variety of systems. I see four dimensions that would be fruitful to explore with self-scaling benchmarks. First, more classes of systems need to be measured. PCs, mainframes, and multiprocessors could each add insights to the usefulness of self-scaling benchmarks as well as validity to the prediction model described in Chapter 5. Second, more network-based client-server systems should be measured, because these types of file servers are becoming more prevalent. Third, the self-scaling benchmark could be used on a wider range of I/O systems, including solid-state and optical disks, magnetic tapes, and systems with file migration. Fourth, non-Unix operating systems would be interesting to measure. Each of these dimensions of different types of systems would demonstrate the utility of self-scaling benchmarks or would serve to further refine the self-scaling ideas. Self-scaling benchmarks could also be applied to non-I/O areas such as processor performance. I suspect this will prove much harder since synthetic workloads for processors are notoriously bad at accurately representing real applications.

In the area of predicted performance, researching ways to enhance prediction accuracy or to better characterize the workload areas that yield high error would be beneficial. One can also imagine a completely different approach to performance prediction. Instead of the current scheme, which separates the self-scaling benchmark's measurement phase from the performance prediction phase, I could introduce feedback between the two phases. For example, the self-scaling benchmark could choose random workloads in the workload space and make sure the current set of focal points accurately predicts performance. When a workload is found for which the current set of focal points does not accurately predict performance, a new focal point could be added to predict that workload. Although this would lead to longer running times for the benchmark, it may enhance prediction.

This thesis has brought up the need for more realistic, flexible benchmarks, that is, benchmarks that can be modified to create many different workloads (such as Willy) but still retain a

link to real-world applications. One possible candidate is a *flexible trace*, where a trace of a real application is modified to have different characteristics (such as changing the number of processes or the average request size).

In conclusion, computer performance evaluation must continue to develop novel ways to stress our rapidly evolving new systems. Self-scaling benchmarks and predicted performance can play a key role in this process because they have the potential to break the weary habit of using obsolete benchmarks by automatically keeping benchmarks up-to-date with new generations of systems.