SAND2013-4501C

Behavior-Based Simulation of Storage Devices

Matthew L. Curry*[1], Carlos Maltzahn[2], Adam Crume[2], H. Lee Ward[1], Ron Oldfield[1] and Patrick Widener[1]

[1]Sandia National Laboratories†, Albuquerque, NM
[2]University of California, Santa Cruz

Whole system simulation of future supercomputers allows exploration of design decisions prior to building a physical system. This is a fundamental component of codesign: Collaboratively altering system architectures and software to achieve maximum benefit for both parts of a system. To achieve maximum fidelity, simulations have to include the storage system, including buses, RAID controllers, and disk drives. Simulating these components can lend insight into design of file systems, application I/O strategies, storage system architectures, and fault tolerance mechanisms.

Current state-of-the-art disk simulation software packages use discrete event simulation (DES). When there is no hidden knowledge about a device, DES can provide very good accuracy. However, modern hard disk manufacturers do not divulge the internal specifications of a disk drive beyond a few parameters, including revolutions per minute, bus type, cache size, etc. There are a large number of hidden design decisions for each disk drive model, including cache algorithms, command queue depth, read and write ordering algorithms, tracks per sector, track layout, etc. These cannot be directly obtained from a disk, and instead must be imperfectly inferred from a disk using a utility like DIG [4] or DiXtrac [6]. The result is an inaccurate model that requires a large effort to improve via manual parameter tweaking. Because DiskSim is intended to simulate a large variety of disks, there are over two hundred parameters available to set. As disks become more complex, refining these models will become more difficult.

Instead of having a large model that is designed to accomodate all disks, we propose to use machine learning techniques to extract the salient features of the performance profile. These are obtained by running real or synthetic workloads against the target disk to gather performance data. The performance data is synthesized by the machine learning algorithm into a light-weight model that captures the salient features of the performance profile for the disk. The result is a small, reasonably accurate, and computationally inexpensive simulator that is quickly custom-generated for the target disk by an automated process.

One of the largest benefits of this approach is that the generation of a simulator will not explicitly discover or know about internal implementation details of a disk. This is an important future-proofing concept, as new generations of disks continually introduce new technolgoies that would require parameterization in a DES. The machine learning approach would instead infer the performance implications of the new features when constructing the performance model and simulator.

The question of which machine learning algorithm is most appropriate is still an open question. Our initial investigations have shown genetic algorithms to be a potentially promising path. Other researchers have investigated regression trees [7, 8]. Deep learning also appears highly promising. Deep learning also appears highly promising, as it is useful for problems that involve a hierarchy of abstractions [2]. For disk drives, such abstractions can include sector, track, serpentine, zone, and so on. Deep learning has already shown to be useful for image classification, giving record results for the MNIST handwritten digit classification problem [5]

*Primary Contact, mlcurry@sandia.gov

†Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.

Related Work

The defacto software package used for simulating disk drives today is DiskSim [3]. DiskSim is a general discrete event simulator with a large set of parameters, which allows it to simulate a large variety of disks with high accuracy. The disadvantage of DES is the large amount of effort required to gather and tune the parameters for the disk model.

Several projects attempt to characterize a disk's potential performance by extracting features of the disks themselves, such as sectors per track, rotational latency, track switch latency, etc. These parameters can then be used to create a model in DiskSim. Worthington et al. showed disk feature extraction of via SCSI-specific query commands and microbenchmarks [9]. DIXtrac measures or infers over 100 parameters, including some that are not related to disk geometry, by running microbenchmarks against a drive [6]. DIG focuses on extracting geometry-specific parameters [4]. These approaches not only rely on prior knowledge of the set of parameters necessary to characterize a storage device, but also on prior knowledge of how to extract values for these parameters from disks. Our proposed approach does not require either.

Wang et al. explored using classification and regression trees (CART) to create disk performance models [7, 8]. This work showed that when training sets and testing sets are very similar, the accuracy is acceptable. However, there was less fidelity when training and testing workloads were not similar.

Project Overview

Challenges Addressed

This approach allows a disk simulator to be easily constructed in a short amount of time with no human intervention or tweaking. Initial work will target modeling individual disks. However, once success has been attained, we believe this approach can be used to create systems that can characterize other higher level components with black box implementations, including flash disks and RAID controllers.

Maturity

An initial investigation using regression trees found success when training workloads closely matched test workloads [8]. We performed an initial feasability study using genetic programming to predict response times. We generated forests of expression trees which could be self-referential, allowing for explicit representation of state. We found that this approach achieved runtime predictions within 10-15% of actual runtimes for simple workloads like sequential access. We were able to account for congestion due to bursty arrivals.

Uniqueness

While this approach is highly useful for exascale systems, it is not unique. However, the modeling and codesign communities seem to be the primary audience that would be interested in creating a model of existing disk drives. Other communities seem more interested in design space exploration of disks using parameterized simulation [1].

Novelty

All currently available simulators are discrete event simulators. We propose a machine learning approach to quickly generate models that closely approximate real disk performance. While there have been some initial investigations into creating tools based on machine learning [8], none are generally available.

Applicability

This approach is applicable to a wide variety of storage devices with hidden implementations, including appliances and other aggregations of heterogeneous storage devices. Implicit in this model is the ability to simulate multiple layers of a storage hierarchy, including hidden caches.

Effort

We estimate that we can effectively explore the above outlined approach through a three year project that uses approximately 90 person-months in total among seven people. This total includes two Ph.D. students at the University of California, Santa Cruz. We intend to spend then first year investigating modeling single disks, with the following two years extending to modeling collections of disks attached to RAID controllers.

References

[1] Nitin Agrawal, Vijayan Prabhakaran, Ted Wobber, John D. Davis, Mark Manasse, and Rina Panigrahy. Design tradeoffs for SSD performance. In *USENIX 2008 Annual Technical Conference on Annual Technical Conference*, ATC'08, pages 57–70, Berkeley, CA, USA, 2008. USENIX Association.

[2] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards ai. *Large-Scale Kernel Machines*, 34, 2007.

[3] John S. Bucy et al. The DiskSim simulation environment version 4.0 reference manual. Technical Report CMU-PDL-08-101, Carnegie Mellon University Parallel Data Lab, May 2008.

[4] Jongmin Gim, Youjip Won, Jaehyeok Chang, Junseok Shim, and Youngseon Park. DIG: Rapid characterization of modern hard disk drive and its performance implication. In *Storage Network Architecture and Parallel I/Os, 2008. SNAPI '08. Fifth IEEE International Workshop on*, pages 74–83, 2008.

[5] Ruslan Salakhutdinov and Geoffrey E Hinton. Deep boltzmann machines. In *Proceedings of the international conference on artificial intelligence and statistics*, volume 5, pages 448–455. MIT Press Cambridge, MA, 2009.

[6] Jiri Schindler and Gregory R. Ganger. Automated disk drive characterization (poster session). In *Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '00, pages 112–113, New York, NY, USA, 2000. ACM.

[7] Mengzhi Wang. *Performance Modeling of Storage Devices using Machine Learning*. PhD thesis, Carnegie Mellon University, January 2006.

[8] Mengzhi Wang, Kinman Au, Anastassia Ailamaki, Anthony Brockwell, Christos Faloutsos, and Gregory R. Ganger. Storage device performance prediction with CART models. In *Proceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, MASCOTS '04, pages 588–595, Washington, DC, USA, 2004. IEEE Computer Society.

[9] Bruce L. Worthington, Gregory R. Ganger, Yale N. Patt, and John Wilkes. On-line extraction of SCSI disk drive parameters. In *Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, SIGMETRICS '95/PERFORMANCE '95, pages 146–156, New York, NY, USA, 1995. ACM.