

Optimizing Memory System Performance for Data Center Applications via Parameter Value Prediction

Shih-wei Liao
Google Inc.
Mountain View, California
sliao@google.com

Donald Nguyen
University of Texas at Austin
Austin, Texas
ddn@cs.utexas.edu

Tzu-Han Hung
Princeton University
Princeton, New Jersey
thung@cs.princeton.edu

Chinyen Chou
National Taiwan University
Taipei, Taiwan
r96079@csie.ntu.edu.tw

Hucheng Zhou
Google Inc.
Mountain View, California
hucheng@google.com

Chiaheng Tu
National Taiwan University
Taipei, Taiwan
d94944008@ntu.edu.tw

ABSTRACT

A typical data center application requires the processor cycles of thousands of machines. Even a single-digit performance improvement can significantly reduce the cost and power consumption of a data center. Unfortunately, achieving sustained improvement, even if modest, is difficult. Data centers are dynamic environments where applications are frequently released and servers are continually upgraded. For maintainability and fault tolerance, the physical capabilities and configuration of the servers are abstracted from the application programmer.

We study application performance under different processor prefetch configurations. These configurations are largely transparent to the programmer, yet we observe a wide range of performance when comparing the worst and best configurations, with relative performance improvement ranging from 1.4% to 75.1%. Alarming, one application that consumes the vast majority of cycles on our data center has a 23.6% improvement.

Default prefetch configurations favor aggressively prefetching memory, which benefits most applications, but some data center applications have highly tuned memory behavior and aggressive prefetching severely decreases performance. We develop a tuning framework which attempts to predict the optimal configuration based on hardware performance counters. It applies to a large number of performance-critical data center applications without modifying the source code or binaries. The framework achieves performance within 1% of the best performance of a suite of important data center applications.

Categories and Subject Descriptors

C.4 [Performance of Systems]: Design Studies

General Terms

Performance

Keywords

Data center applications, machine learning

Copyright is held by the author/owner(s).
ICS'09, June 8–12, 2009, Yorktown Heights, New York, USA.
ACM 978-1-60558-498-0/09/06.

1. INTRODUCTION

Data center operators leverage economies of scale to increase performance per watt. The large scale of data center applications encourages operators to be meticulous about application performance. A single-digit improvement can obviate thousands of machines. However, optimizing data center applications is a delicate task. To increase machine utilization, operators host several applications on the same physical machine, but this may reduce aggregate performance by increasing contention on shared hardware resources. Cloud Computing introduces third-party applications into the data center. This adds new dimensions to the configuration space and adds external dependencies into the optimization problem. Consolidation is driven by application scale, the need for high utilization and emerging applications. Consolidation dually multiplies the impact of application optimizations and increases its difficulty.

We propose a system to optimize applications according to data center constraints. We model optimization as a parameter search problem and use machine learning techniques to predict the best parameter values. This framework is designed for seamless integration into existing data center practices and requires minimal operators' intervention.

In this work, we focus on optimizing memory system performance by configuring memory prefetchers. As hardware moves towards integrated memory controllers and higher bandwidth, configuring prefetchers properly will play an important rule in maximizing system performance. Currently, manufacturers tune prefetchers for aggressive prefetching. This produces superior performance for applications that are not limited by memory bandwidth, but aggressive prefetching produces pathological behavior for applications that have finely tuned memory behavior. Due to the impact of scale, programmers of data center applications commonly apply memory optimizations that increase the effective utilization of bandwidth like data structure compaction and alignment, and software prefetching. Aggressive prefetching can destroy programmers' careful orchestration of memory.

We have found few papers that address the impact of prefetcher configuration from the perspective of the application programmer or the data center operator, even though proper configuration has a significant performance impact. Instead, most papers address the concerns of hardware designers. Choosing the right configuration, however, has a

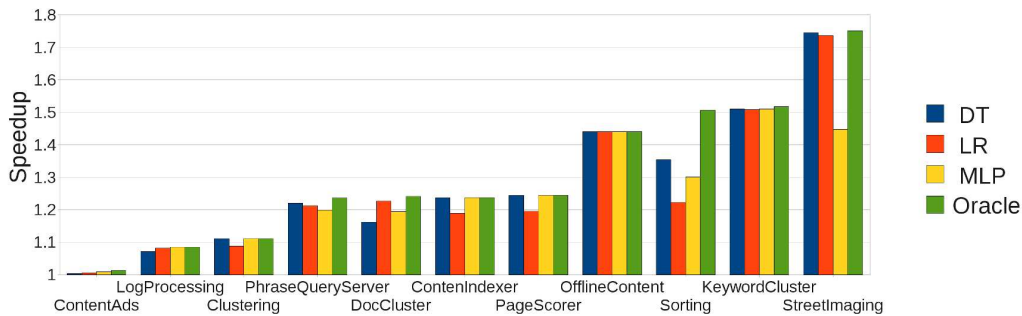


Figure 1: Performance improvements of data center applications relative to worst prefetch configuration by using various machine learning algorithms. *Oracle* denotes the best performing configuration for an application found through exhaustive enumeration.

large impact on performance. On a collection of 11 data center applications, we find a range of improvement between 1.4% to 75.1% comparing the worst to best performing configurations (see Figure 1).

We apply machine learning to predict the best configuration using data from hardware performance counters, and we show a methodology that predicts configurations that achieve performance within 1% of the best performance of a suite of data center applications.

We only show results for isolated application executions. To address the context of data centers, we must also consider co-hosted applications, dynamic application mix and large-scale deployments. We believe that our general tuning framework can be extended to address these issues, but we will consider them in future work.

Our contributions are the following:

1. A system that significantly improves application performance transparently to the application programmer and with minimal overhead to data center operators.
2. A study of the performance of real data center applications under varying hardware prefetcher configurations.
3. An instantiation of machine learning, including selecting the training dataset and problem formulation, that predicts near-optimal prefetch configurations.

2. RESULTS

Our performance tuning framework models application optimization as a parameter search problem using machine learning to predict the best parameter values. We specifically consider the problem of selecting the best memory prefetcher configuration for a machine. For the Intel Core 2 architecture, each processor core has 4 independent prefetchers, each of which can be turned on and off. In total, there are 16 different configurations for each core.

We evaluate several different machine learning algorithms and problem formulations. Figure 1 summarizes our results. DT is a decision tree algorithm. LR is a linear regression model. MLP is a multilevel perceptron. There are many different ways to formulate our prefetcher selection problem as a machine learning problem. The formulation that we report here is as follows. Our feature vector is a list of hardware event counters related to memory performance (e.g., L2 lines

in and out, branch prediction misses, cache lines prefetched) and prefetcher configuration. The learning problem is to predict the best performing configuration given values for the hardware event counters.

For some applications, performance is relatively insensitive to prefetcher configuration. We exclude these applications from our training set. For many applications, several configurations result in performance within a small fraction of the best performing configuration. To avoid overfitting, we consider all near-best configurations as equivalent.

Since we are concerned with data center optimization, our true performance metric is overall system performance. We assign a weight to each application in our benchmark suite which represents the relative number of processor cycles consumed by each application (not shown here) in the data center. When examining the weighted performance of the DT algorithm, we find that we achieve a weighted performance within 1% of the oracle.

3. CONCLUSIONS

In this work, we highlighted some of the difficulties of optimization within the data center environment. We proposed a framework for parameter optimization based on machine learning. We performed several experiments to understand the sensitivity of the framework to several influential components: machine learning algorithms, training datasets and problem formulation methods. For the specific problem of choosing hardware prefetcher configurations, we show that our framework can achieve performance within 1% of the best configuration. We have shown that judicious exploitation of the machine learning-based framework is a promising approach to conquer the parameter value optimization problem.