Leveraging Approximation to Improve Datacenter Utilization

Feng Qi(fq26)
Advisor: Christina Delimitrou
School of Electrical and Computer Engineering

Motivation

Cloud Computing has increased in popularity, but datacenter utilization has remained for the most part low.

Resource utilization over 30 days for a large production cluster at Twitter managed with Mesos[1].

Disadvantage of common approach is resource contention coming along with application sharing hardware and software resources, thus causing unpredictable performance.

Introduction

We present Pliant[2], a cloud runtime system that improves datacenter utilization by co-scheduling interactive services with approximate computing applications.

When the runtime detects quality of service(QoS) violations in the interactive service, it employs approximation to reduce contention, and absorbs the resource reduction as a loss in output accuracy.

Approximate Applications

- Significance of approximate computing
  Those applications absorb enforced resource reduction as a loss in output quality and present an opportunity to improve datacenter efficiency without performance degradation.

- Way to find approximate application
  ACCEPT[3] compiler framework is used to find these regions of codes that can be approximated.

Evaluation

- Configuration:
  We use two servers with 20 physical (40 hyperthreaded) cores each, and 128GB of RAM, one as server and one as client.
  We use memcached as the latency-critical application, and PARSEC, Spark MILib and SPLASH-2 as the approximate computing applications.

- Results:
  (a) Comparison of 99%-tile latency of memcached when co-located with precise and approximate versions of memcached application.
  (b) Performance stats of the precise and approximate versions of memcached application that correlate with the 99%-tile latency of the corresponding memcached runs.

System Design

The overview setup of Pliant. The interactive service shares physical cores with the approximate computing applications. A different client machine is used to drive the load of the interactive service.

- A lightweight performance monitor
  Runs on the client side and continuously samples the throughput and end-to-end latency (average and tail) of the interactive service, and notifies the runtime system in the event of a QoS violation.

- An interference monitor
  Runs on the server and collects performance counter information that identifies the resource(s) suffering from contention.

- A runtime system
  Enforces a degree and method of approximation based on the output of the performance and interference monitors.
  The system uses DynamoRIO[4] to switch between the precise and different approximate versions of the approximate computing applications.

Summary and Future Work

Plaint is able to leverage approximate computing applications to increase datacenter utilization. It co-schedules interactive services with approximate applications and employs incremental and interference-aware approximation, selecting the appropriate type of approximation based on the level and cause of the QoS violation of the interactive application.

Future work:
1. Configure more cloud approximate applications that makes our system practical.
2. Explore the possibility of switching multiple approximate applications to varying levels of approximation.
3. Develop LLVM-based compiler instead of using DynamoRIO to reduce switching overhead.
4. Leverage hardware isolation techniques including containers, thread pinning.

Reference


Acknowledgement

I would like to thank Seungjun Kulkarni and Gyfika Fesole for their excellent effort to this project. And I also would like to sincerely thank Prof. Chris for her remarkable insight and advice. This project would not have been possible without their help.