Data Center Workload Characterization

Christina Delimitrou

Outline

- Introduction
- Related Work
- Implementation
- Proposal
- Future Work
- Suggestions

Outline

- Introduction
- Related Work
- Implementation
- Proposal
- Future Work
- Suggestions

Introduction

- No representative DC workloads publicly available
  - If available, only represent snapshot of the system
- We need to decouple the workload from the underlying system
  - What are the real characteristics of the workload?
  - How can we use it to improve performance/efficiency of the system?

Methodology	Platform Decoupled	System (hw/sw) evaluation	App characterization
Traces	No	(sort of)	No
Models	**Yes**	**Yes**	**Yes**

- Models offer more capabilities to **understand** and **use** a data center workload

Goal

Create a compressed but representative model that captures the features of:

- Storage system
- 3-tier system
- A complete large scale application

… without being as dependent to the underlying system as traces

Outline

- Introduction
- Related Work
- Implementation
- Proposal
- Future Work
- Suggestions

Related Work

- **Storage Characterization for Unstructured Data in Online Services Applications.** Sriram Sankar,Kushagra Vaid (Microsoft)
  - *Mantra*: Derive a **probabilistic model** for disk accesses in large scale applications
  - Start from traces for 3 popular 3-tier apps (hotmail, maps, user-client)
  - Collect stats on:
    - Number of I/Os (app load intensity-block size,type(rd/wr), randomness)
    - Spatial distribution of I/Os (disk blocks)
    - Inter-arrival times between I/Os
    - Outstanding I/O queues
  - Divide disk space in block ranges - assign one state per block range
  - Represent probabilities of changing states between I/Os w/ transitions

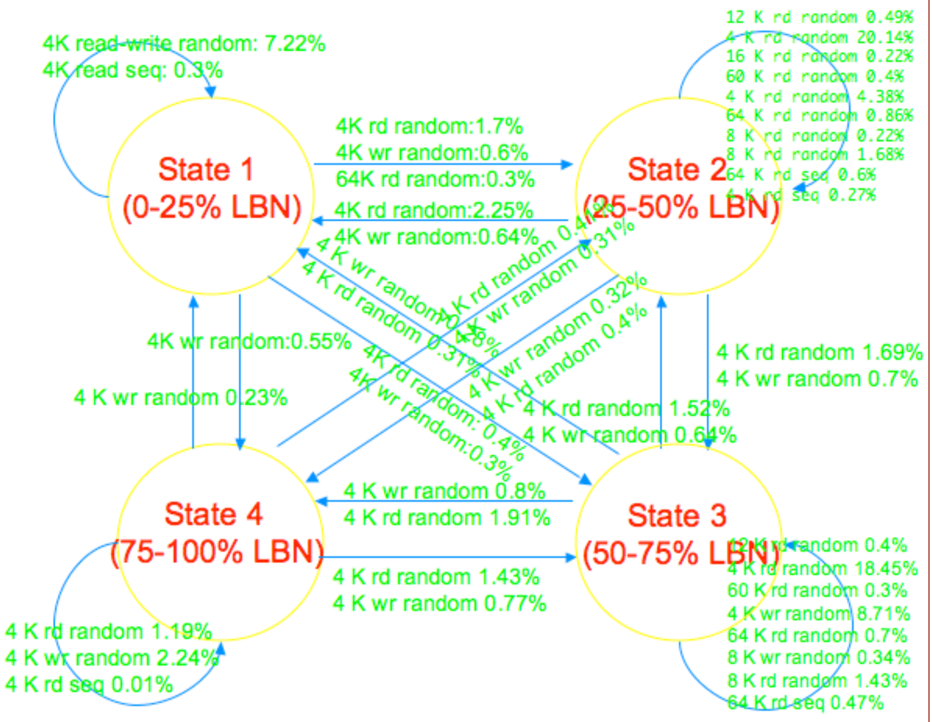
Outline

- Introduction
- Related Work
- Implementation
- Proposal
- Future Work
- Suggestions

Impl : Storage Accesses via State Diagrams

STANFORD ELECTRICAL ENGINEERING

- Model: probabilistic state transition diagrams
  - **State**: Block range
  - **Transition**: Probability of changing block range
  - **Stats**: rd/wr, rd/seq, block size
- Single or Multiple Levels
  - Coarse grained representation
  - Hierarchical representation
  - User Defined level of granularity



This state diagram from a multicore MS server that hosts **hotmail**

Impl : Storage Accesses via State Diagrams (contd)

1. Scripts to read the state diagram (of one of more levels) and define number of worker threads

2. Generator: IOmeter++
   - ✓ Modified open source storage workload generator
   - ✓ Can replay workload on one or more servers (user defined parameter)
   - ✓ Generates a workload that resembles the original in:
     - Probabilities of transitioning between states
     - Inter-arrival rates
     - Block sizes
     - Read/Write and Random/Sequential characteristics
   - ✓ Remains to validate on the response time of requests…

Outline

- Introduction
- Related Work
- Implementation
- Proposal
- Future Work
- Suggestions

Main Milestones

- Workload modeling
  - Use stats techniques to model behavior DC workloads
  - Capture all tiers, whole system, interactions
    - Net, CPU, memory, storage, etc
  - Discussion topics: tracing & modeling techniques, level of detail, target workloads, platform (in)dependence, metrics to capture, …
  - Will try to do that with help from REU/CURIS in the summer…
- Workload generators
  - Create generators that can replay workloads on other sites and other systems
  - Share the generators and models with the community

Datacenter Workload Modeling

STANFORD ELECTRICAL ENGINEERING

- First step: Setup some workloads locally
  - Analytics => Hadoop + (GridMix and PigMix)
  - Virtualized computation => Xen
  - SPECWeb 2009
  - Media streaming & mining => biocomp data
- Workload classes
  - Analytics (mapreduce, hadoop, …)
  - HADI (large scale graph algorithm)
  - 3-tier (mail, maps, apps, …)
  - Latency sensitive apps (search, facebook)
  - Virtualized computation (EC2) and storage (S3)
  - Streaming media (youtube, …)
  - What else??

The Work Proposed

- Traces - 2 - State Diagrams
  - Create the probabilistic model from traces of real workloads (extract probabilities of states and transitions)
  - Make the representation hierarchical and modular
    - Configurable sublevels of hierarchy
    - Inter-arrival time distributions
  - Make the tool for creating the state diagrams publicly available (online library??)
- State Diagrams - 2 - Synthetic Workloads
  - Use the modified workload generator to read the state diagrams and create the synthetic workload
  - Validate the accuracy of workloads against real Microsoft applications
  - Use the synthetic workload to evaluate hardware/software options for efficient servers
  - Make the workload generator publicly available

In Progress

STANFORD ELECTRICAL ENGINEERING

- ✓ Modified Iometer to read the state diagram and create the synthetic workload (represents: states / probabilities / inter-arrival rates)
- ✓ Extended to create a workload from a hierarchical representation

- ✗ Waiting for the scripts that create the state diagrams
- ✗ Currently, validating the resemblance between original and synthetic workloads

Benefits & Limitations

- ✓ High portability
- ✓ Not as “perishable” as traces - not as coupled to hardware
- ✓ If validation succeeds :) they will offer a compressed version of a highly scalable application
- ✓ Opens opportunities for detailed characterization (pattern analysis, learning techniques) for DC workloads
- ✓ Can be used to evaluate and propose efficient solutions for DC design

---

- ✗ Some things are considered fixed:
  - ✗ App code
  - ✗ OS/software in general

  } Model assumptions
- ✗ If these change, the model is no longer representative of the application

Outline

STANFORD ELECTRICAL ENGINEERING

The Fork…

- **Hybrid System Proposal**
  - Use the synthetic workload to evaluate different software - hardware options for efficiency and performance
  - Would use of mobile computing components make sense?
    - Mobile RAM, Flash, SSD
  - Would powering-off disks make sense?
  - Propose a hybrid storage system that improves efficiency
- **Expand a similar methodology to other aspects of the system**
  - **CPU** Utilization
  - **Network** Traffic
  - **Memory** Utilization

    } **Correlation???**
  - What metrics would be interesting/useful to consider?
  - Again, validate with original workloads and propose system options for efficiency
    - Low power processors (Atom, Nvidia, ARM)
    - Network topologies
    - Memory technology

Outline

- Introduction
- Related Work
- Implementation
- Proposal
- Future Work
- Suggestions

Suggestions

STANFORD ELECTRICAL ENGINEERING

- Get workloads - traces through internships
- Use the models to create our synthetic workloads
- Use them to:
  - Evaluate systems - propose energy efficient hybrid systems for data center design
  - Expand this study to other aspects of the system
    - Is it efficient?
    - What overheads will it issue??
    - **Feedback needed**
    - What pattern analysis techniques??
  - Energy efficient memories
  - Network issues
  - Scale down issues

$\left.\begin{array}{l}\text{Energy efficient memories}\\ \text{Network issues}\\ \text{Scale down issues}\end{array}\right\}$ All of interest to EPIC

Suggestions

- What workloads are of interest to each team/would like to see synthetic versions of?
- In what projects could this be used?
- What is the time table for your workloads (initial stage, ready, finished?)
- Will it make a difference in the systems you are designing?
- What other tools/profiling/workload generation would be useful to you?

Questions??

STANFORD ELECTRICAL ENGINEERING

Thank you