Modeling and Simulating Flash based Solid-State Disks for Operating Systems

Kaoutar El Maghraoui
Gokul Kandiraju
Joefon Jann
Pratap Pattnaik

IBM T. J. Watson Research Center

Outline

- Solid-state Disks vs. Hard Disks
- Related Work
- Internal Architecture of an SSD
- Linear model for Throughput
- Microbenchmarks for Parameter Extraction
- Flash Simulator
- Experimental setup and Results
- Conclusion

Solid-State Disks vs. Hard Disks

Hard Disks

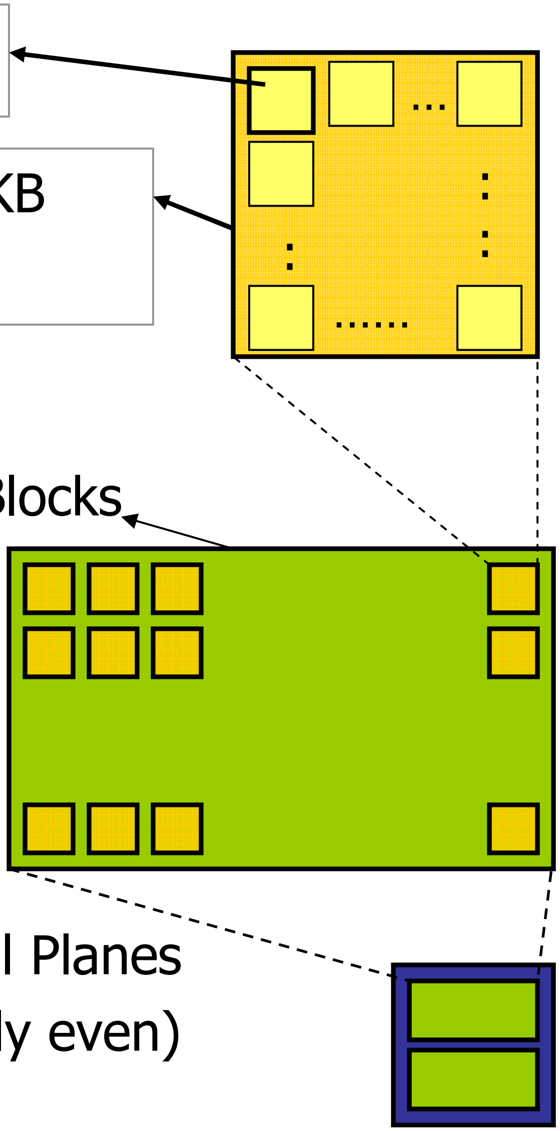
- Most widely used storage devices (introduced in 1956)
- Mechanical Nature
  - Data stored on platters
  - Data access requires head movement: seektime, rotation time, transfer time.
- Performance strongly dependent on access pattern
- High power consumption
- Lower resistance to shocks

Solid State Disks

- Semiconductor Device built on Flash Memory
  - NAND-based Flash memory is a solid-state memory that allows the storage of persistent data
- No mechanical components
  - No seek time. Therefore can provide a much faster, and a more uniform random access speed compared to HDD
- Low access latency, Low power consumption
- Higher resistance to shocks, Small size / Light weight
- Making huge strides into Enterprise Storage
- Consumer Electronics will drive the use of SSDs in future

Solid-State Disk: Typical Organization & Behavior

- 1 Page = 4KB
- 1 Block = 256KB (64 pages)
- 1 Plane = 2K Blocks = 512MB
- 1 Chip = Several Planes (usually even)



- Types of Flash Memory
  - Single Level Cells (SLC)
  - Multi Level Cells (MLC)

- Operations supported
  - Read
    - Page Granularity
  - Erase
    - Block Granularity
  - Program
    - Page Granularity

- Can Wear-out
  - SLC : 100K erase limit
  - MLC: 10K erase limit
  - Wear-leveling is provided for Endurance

Modeling/Simulating SSDs

- SSDs are Expensive

- Modeling/Simulation has always been an important tool for systems research as it gives insight into the system behavior, narrows down the design space, and reduces implementation efforts

- Previous work on building NAND-Flash based SSD simulators
  - Microsoft Research: A NAND-Flash simulator based on the DiskSim simulation environment from the CMU
    - Simulates SSD latencies, multiple request queues, logical block maps, block erasure, and wear-leveling, the page-based FTL scheme
  - U. Seoul: CPS-SIM is another Flash SSD simulator that is limited to a single FTL scheme
  - PennState: Flash-Sim from PennState supports simulating multiple FTL schemes using workload traces. Flash-Sim uses also DiskSim to simulate queuing effects

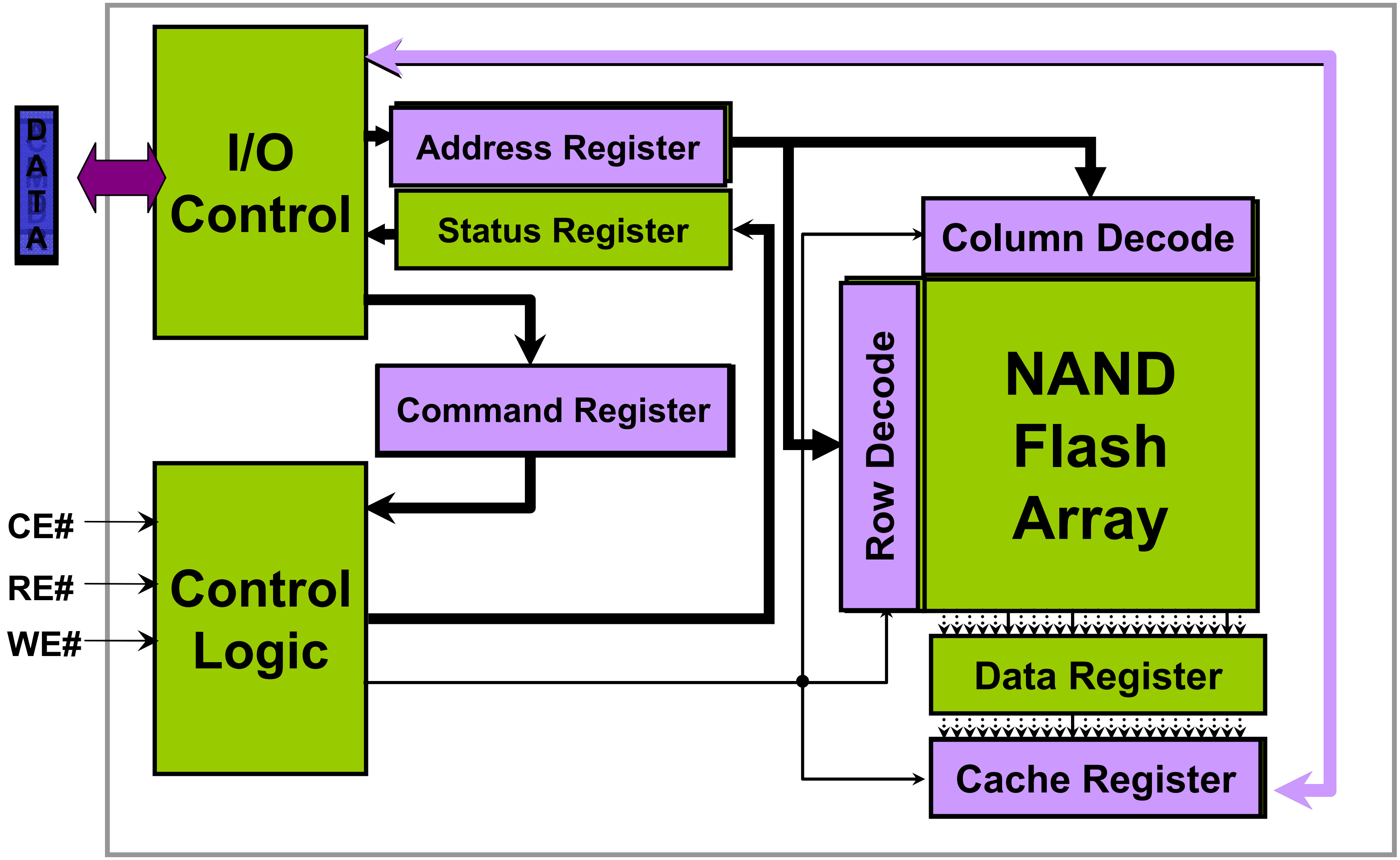
- Modeling an SSD has received far less attention:
  - In fact, to our knowledge, there is no prior work that develops models for Flash devices (work in this direction has been more focused on developing circuit level models)

Our Contributions

- In this paper, we
  - Propose a Model for SSDs and model their throughput based on the internal Flash architecture
  - Use this Model to develop a simulator for SSDs
    - Compared to the other simulators, our simulator is capable of simulating any Flash SSD devices because its parameters can be extracted from any SSD
    - No Traces
      - A kernel extension that can be configured on a running OS as a paging device. Hence, it does not require any traces as input
      - A good choice to test (i) Impact of Flash on scientific/commercial applications and (ii) Novel OS policy changes for Flash devices
  - Propose Benchmarks to extract the Model Parameters
    - We extract the parameters from a Real SSD and feed into our Simulator
  - Validate the Simulator using Commercial Applications/Microbenchmarks.

Internal Operation of a Flash Device

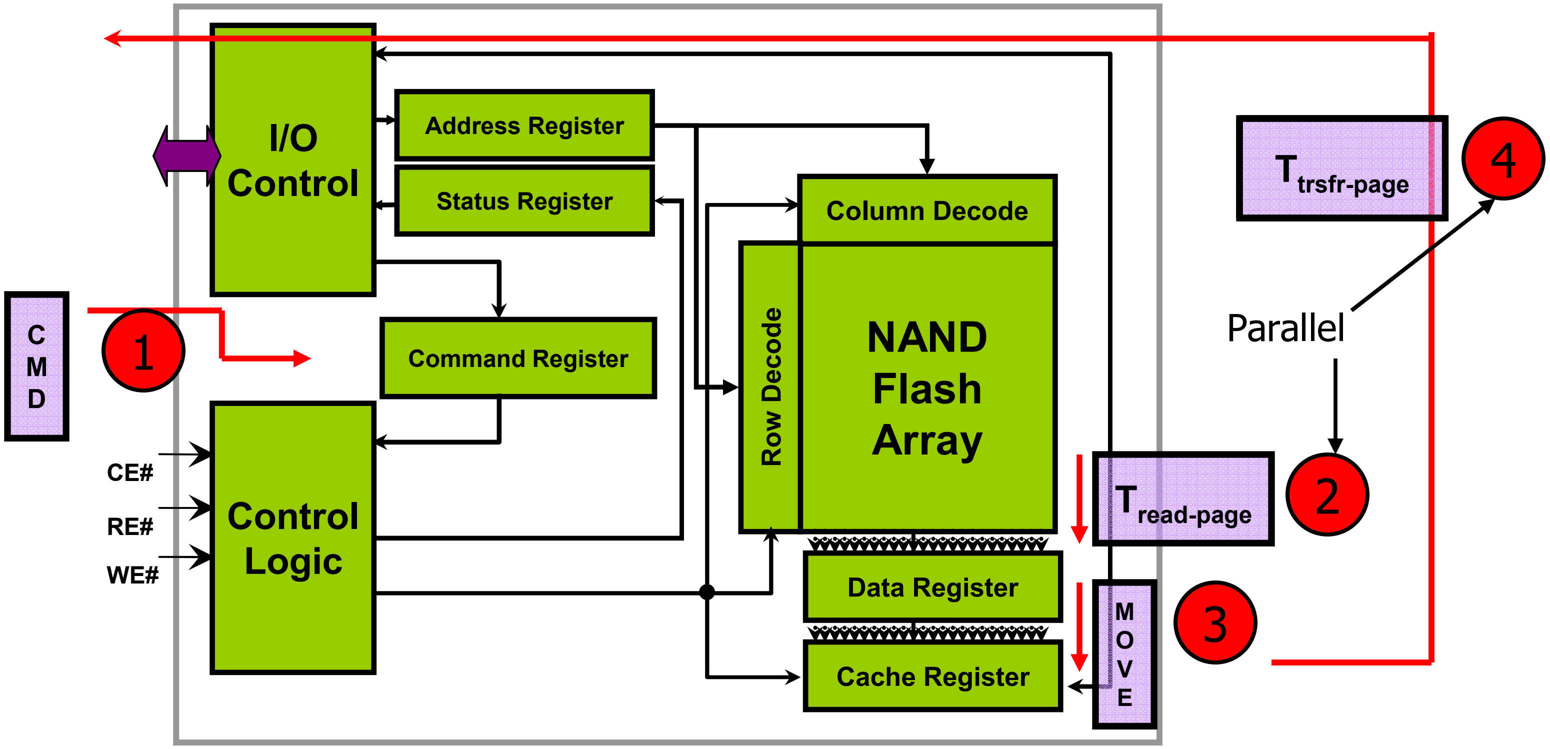
FlashChip



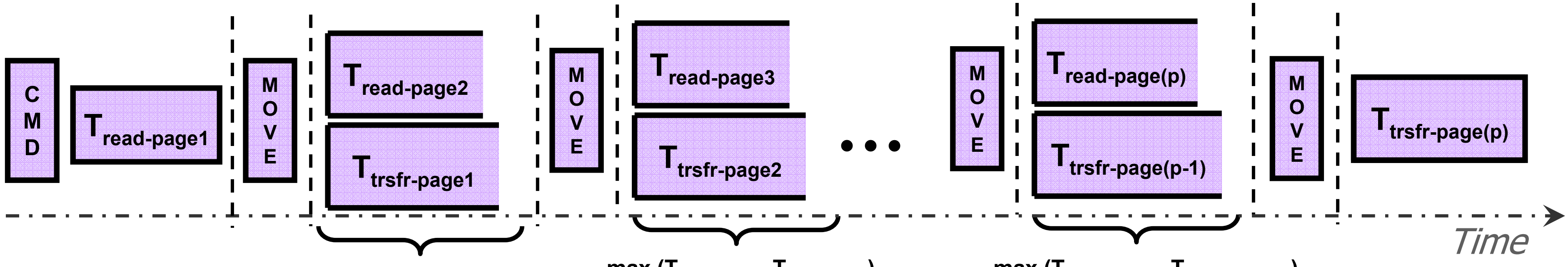
CE: Chip Enable
RE : Read Enable
WE: Write Enable

Source : Micron NAND Flash Memory Technical Document, MT29F4G08AAA, MT29F8G08BAA, MT29F8G08DAA, MT29F16G08FAA.

Data Read Operation Timing – *Cached Mode*



Total Read Time:



max ($T_{read-page2}$, $T_{trsfr-page1}$)

max ($T_{read-page3}$, $T_{trsfr-page2}$)

max ($T_{read-page(p)}$, $T_{trsfr-page(p-1)}$)

Linear Model for Timing/Throughput

- Read Time (p Pages)

$$= T_{cmd} + T_{read} + p * T_{move}$$

$$+ \ (p - 1) \ \mathit{max}[T_{transfer}, T_{read}] + T_{transfer}$$

- Therefore,

Read Time (n Bytes) $= A + n * B$

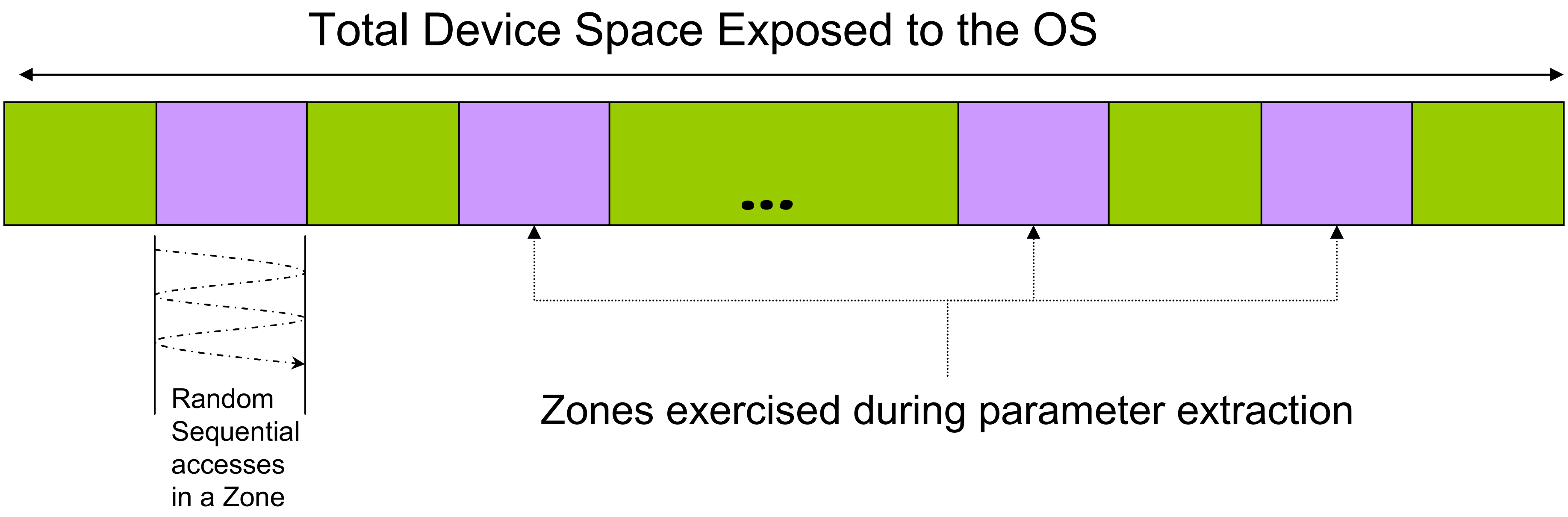
A and B are
- constants that depend on the Chip characteristics
- Will capture OS overheads and Flash controller variations when determined through experimentation

Hence,

Sequential Read Throughput(n): $\dfrac{n}{A_{sr} + n * B_{sr}}$

Similar parameters for other patterns: $\{A_{sw}, B_{sw}, A_{rr}, B_{rr}, A_{rw}, B_{rw}\}$

Extracting Model Parameters

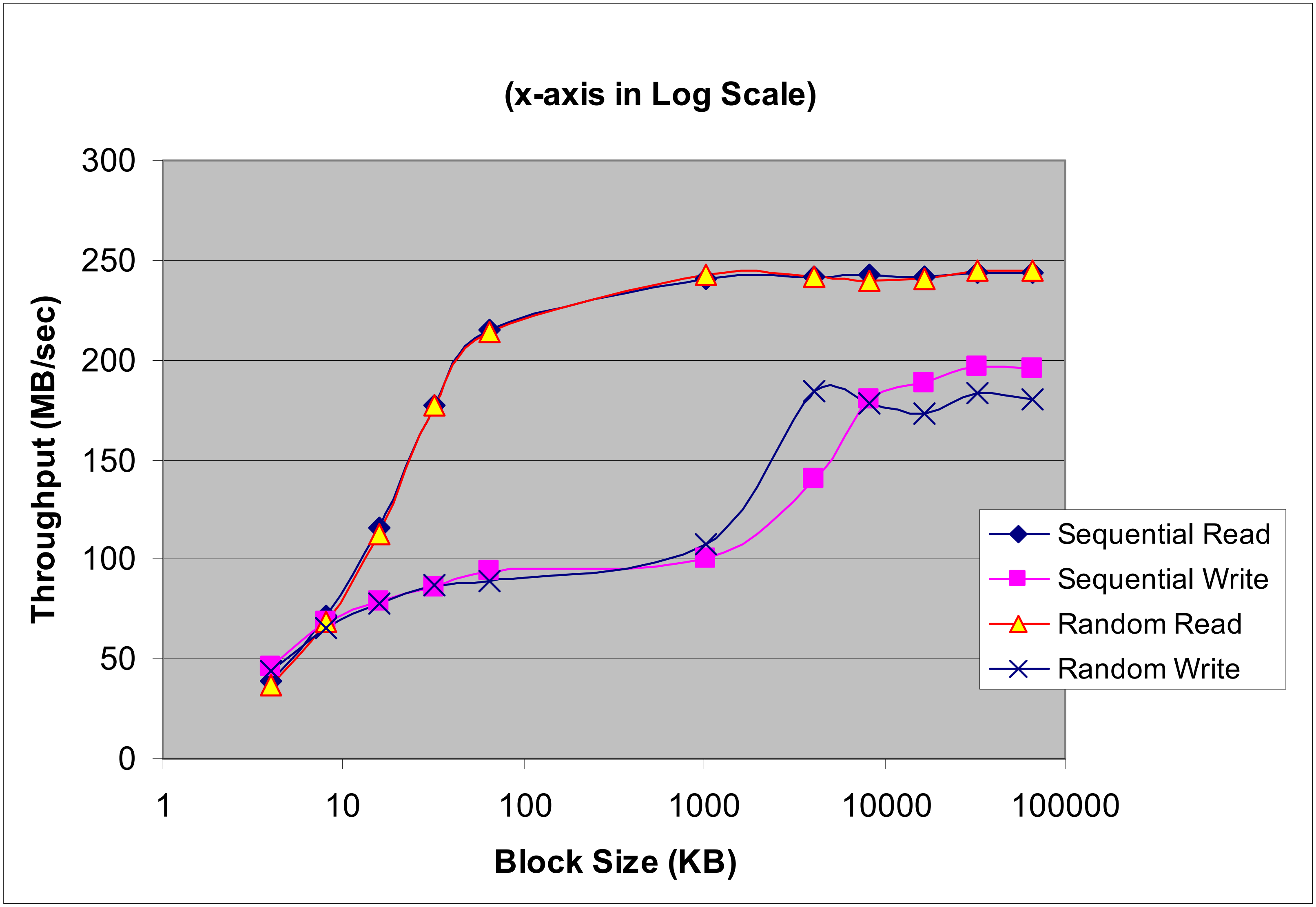


Total Device Space Exposed to the OS
...
Random Sequential accesses in a Zone
Zones exercised during parameter extraction

- Device accessed in ‘Raw’ mode
- Odd no. of zones to avoid any correlations
- Generate Seq/Rand accesses for various block sizes (request sizes)

Throughput for a pattern (SR,SW,RR,RW) for each block-size
= Avg. Throughput across all requests for all zones
for that block-size

Throughput of the Zeus Flash Disk



Linear Regression Fit Results

	Seq Rd	*Rand Rd*	*Seq Wr*	*Rand Wr*
A(μs) *(Intercepts)*	127.5 (A_{sr})	230.6 (A_{rr})	2167 (A_{sw})	770 (A_{rw})
B(μs/KB) *(Slope Coefficients)*	4.005 (B_{sr})	3.987 (B_{rr})	4.96 (B_{wr})	5.382 (B_{rw})
r^2	0.999997	0.999981	0.99932	0.999722
P-Value	5.21 x 10^{-26}	1.39 x 10^{-22}	1.45 x 10^{-15}	2.54 x 10^{-17}

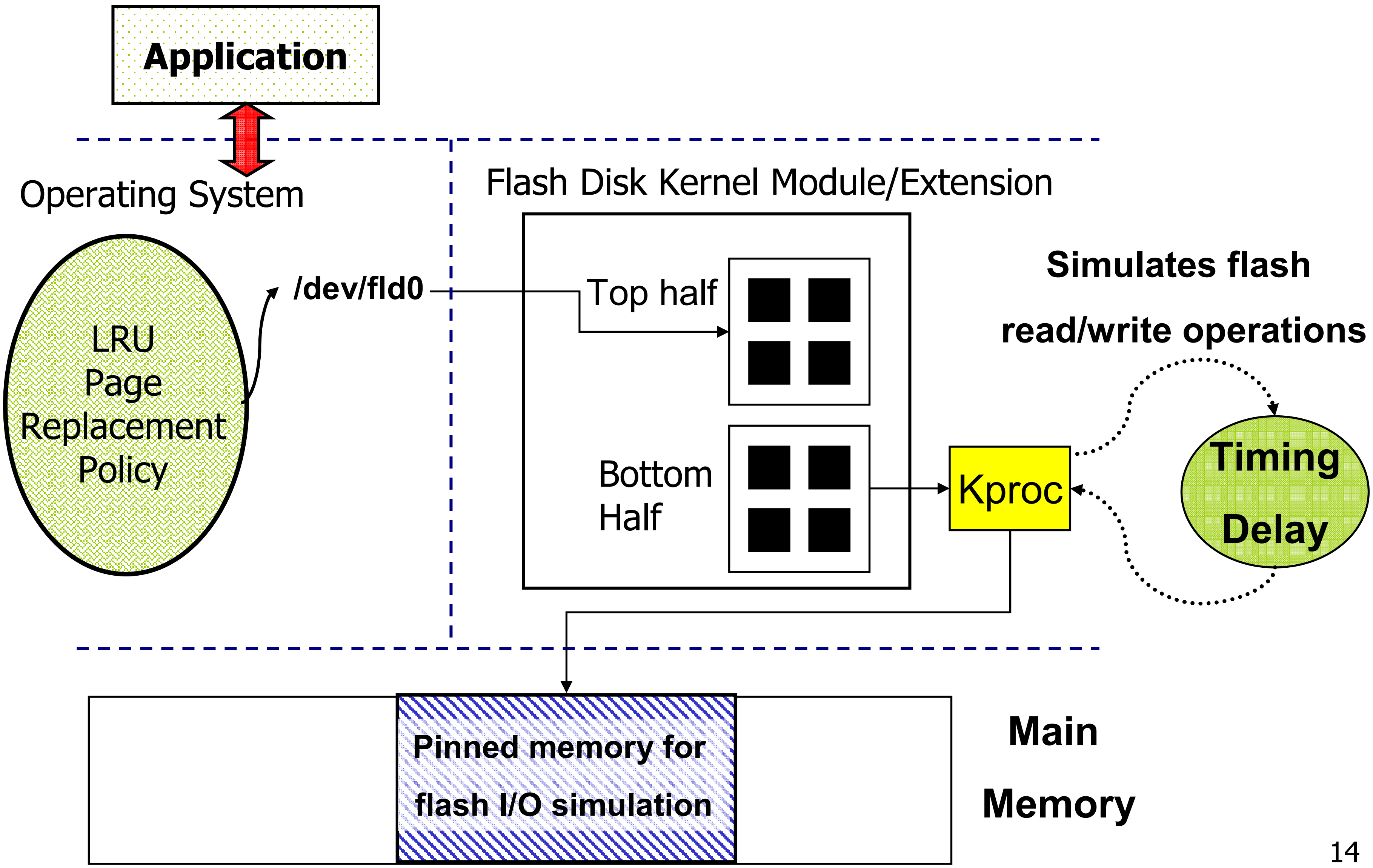
Throughput Parameters Extracted for the STEC Zeus Flash Disk

Flash Simulator on AIX

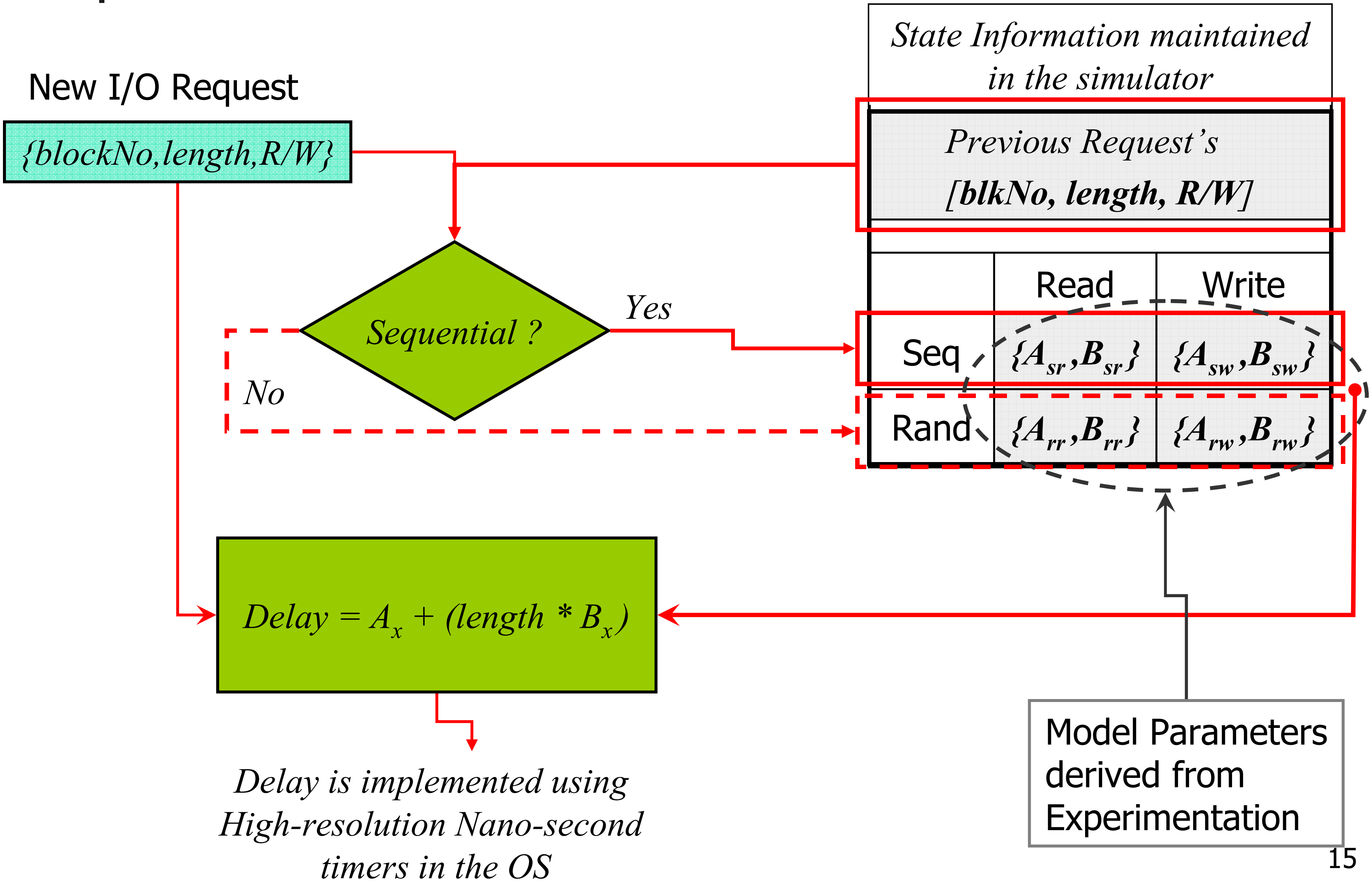
- A Kernel Module/Extension
  - A Flash disk appears once the module is loaded
- Flash Disk is simulated using pinned memory
- The parameters to simulate the delay are the Extracted Model Parameters
- No need of Traces
- Easily configurable on AIX with a few commands

- Timing delays for read/write operations
- Simulating the Delay without building the internal Flash layout in the simulator. Delay based on:
  - Sequentiality
  - Size of the Requests
  - Read/Write
- Minimal State information
  - Previous Request's [*BlockNo, Size, Read/Write*]
  - Model Parameters

Flash Simulator on AIX



Simulating the Delays

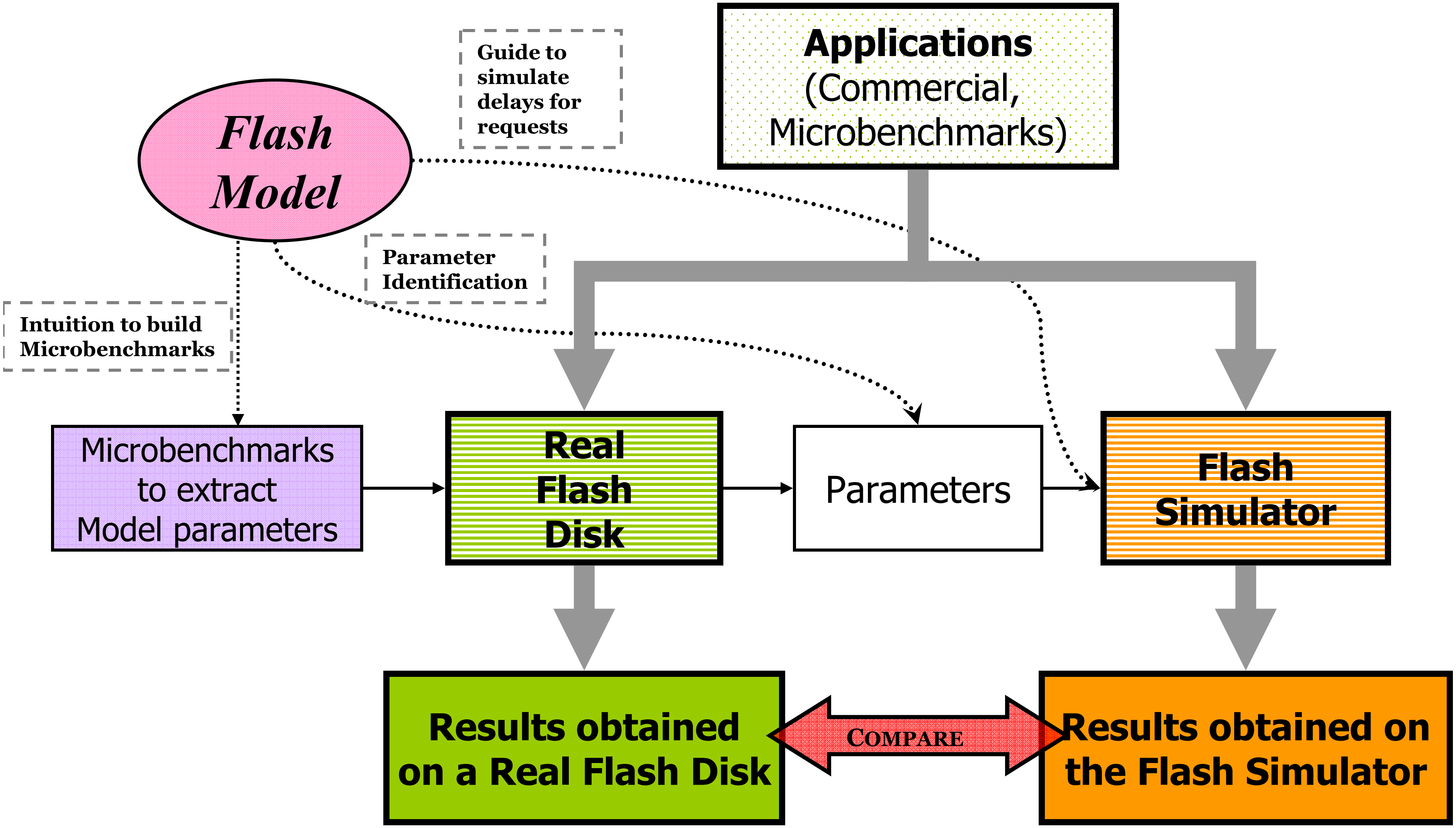

New I/O Request
{blockNo,length,R/W}
Sequential ?
Yes
No
State Information maintained in the simulator
Previous Request's
[blkNo, length, R/W]
Delay = A_x + (length * B_x)
Model Parameters derived from Experimentation

	Read	Write
Seq	$\{A_{sr}, B_{sr}\}$	$\{A_{sw}, B_{sw}\}$
Rand	$\{A_{rr}, B_{rr}\}$	$\{A_{rw}, B_{rw}\}$

$$Delay = A_x + (length * B_x)$$

Delay is implemented using High-resolution Nano-second timers in the OS

Overview of the Approach

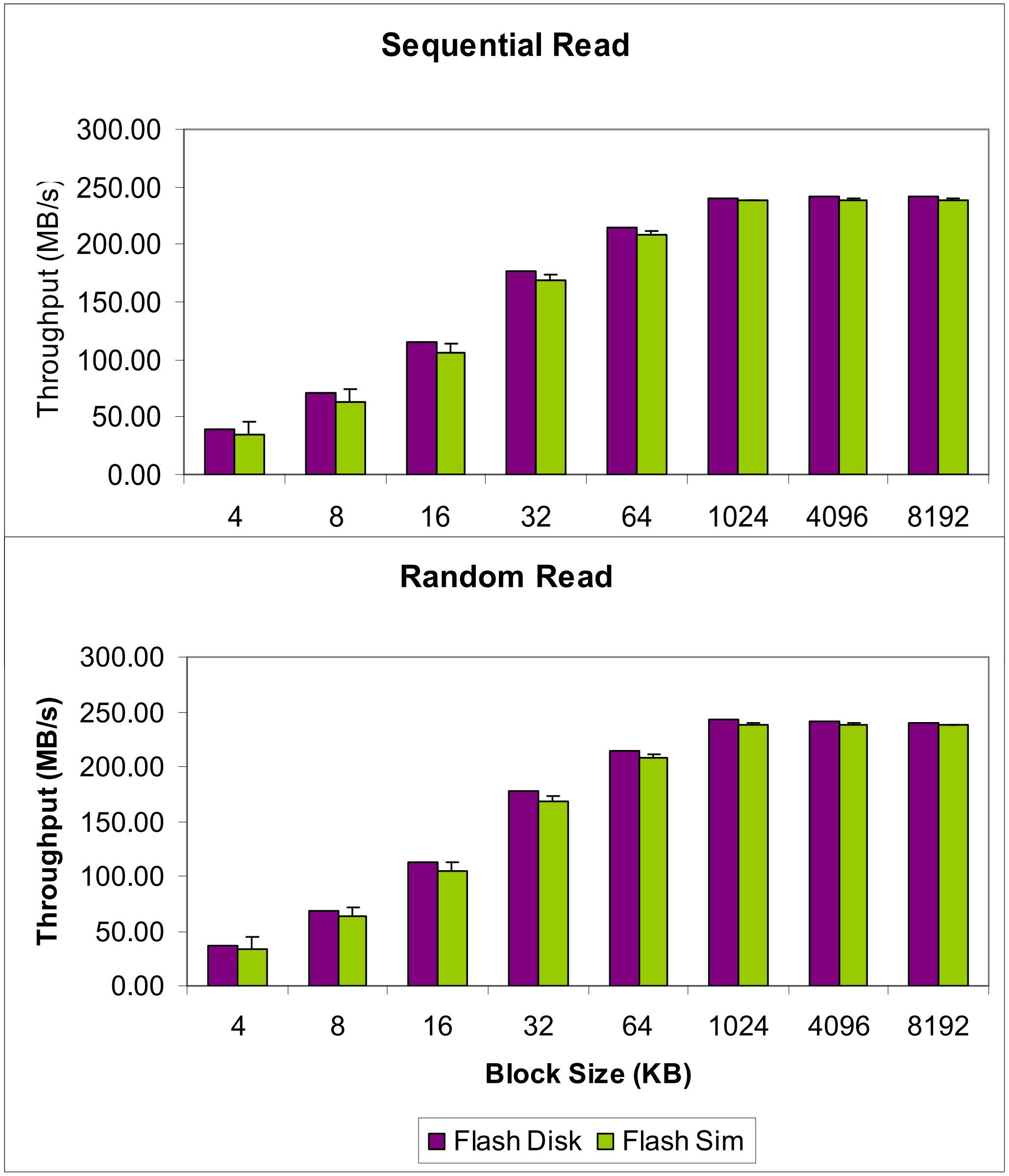

Applications
(Commercial,
Microbenchmarks)
Flash
Model
Guide to simulate delays for requests
Parameter Identification
Intuition to build Microbenchmarks
Microbenchmarks to extract Model parameters
Real Flash Disk
Parameters
Flash Simulator
Results obtained on a Real Flash Disk
COMPARE
Results obtained on the Flash Simulator

Experimental Setup

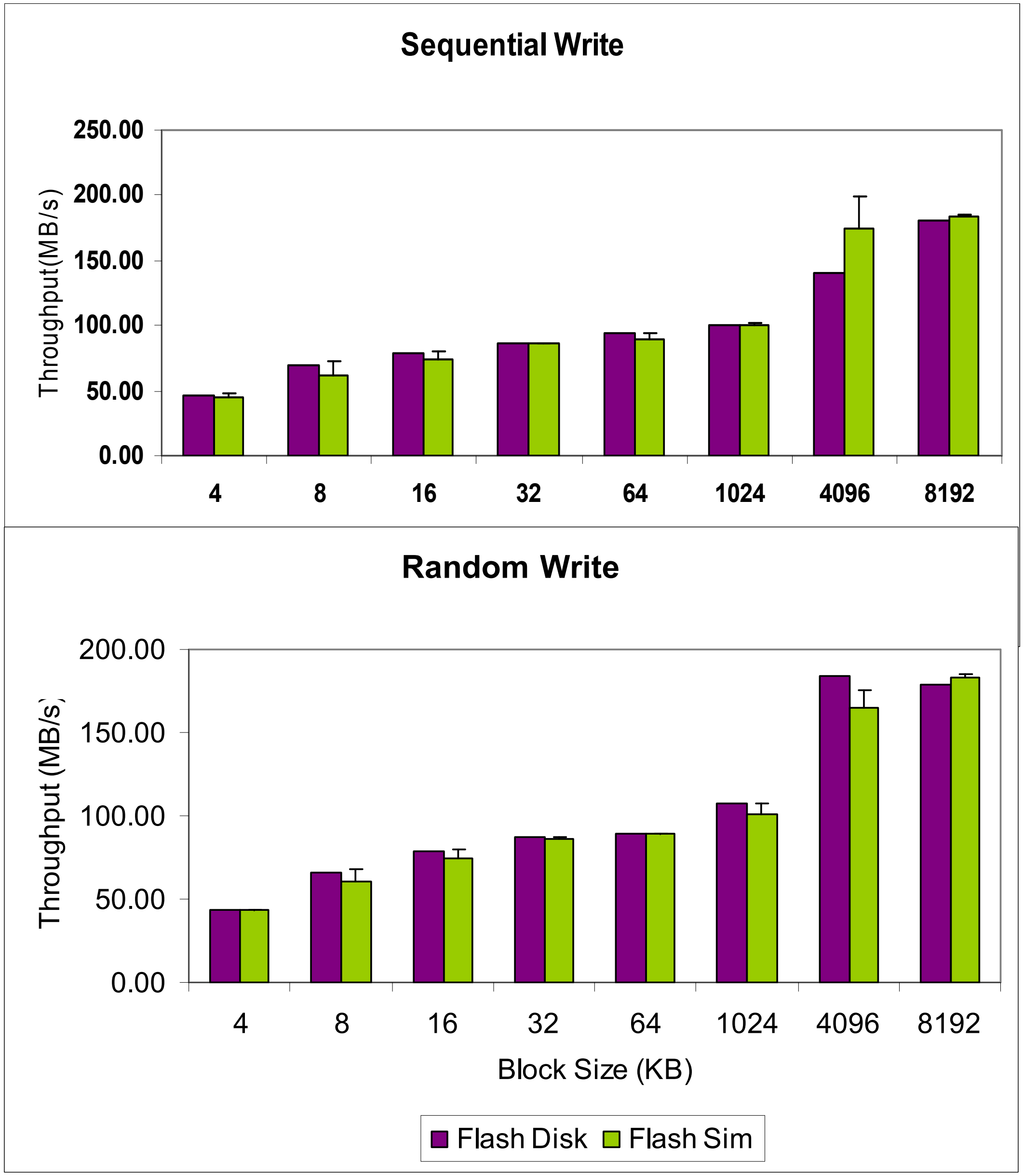
- All the experiments were conducted on
  - IBM AIX Version 6 Operating System
  - POWER6 processor
    - 4.7 GHz clock speed
    - 2 physical CPUs and 2 hardware threads (SMTs) per CPU
  - 2 STEC Zeus 70GB Solid-State drives (for paging)
  - IBM 300GB SAS Hard Disk drive (for paging)

- Applications
  - Microbenchmarks (Raw I/O)
    - Sequential Read, Random Read, Sequential Write and Random Write
  - SPEC Jbb 2005 Benchmark
  - Day Trader Benchmark

Raw I/O benchmarks Results: Read



Raw I/O benchmarks Results: Write



Percentage Error for Raw I/O Workloads

I/O Operation	Average Percentage Error (%)
Random Read	**4.6**
Sequential Read	**5.31**
Random Write	**4.10**
Sequential Write	**6.57**

Average % Error of the Raw I/O Throughput of Read and Write ops between the Flash Simulator and the Zeus Flash Disk.

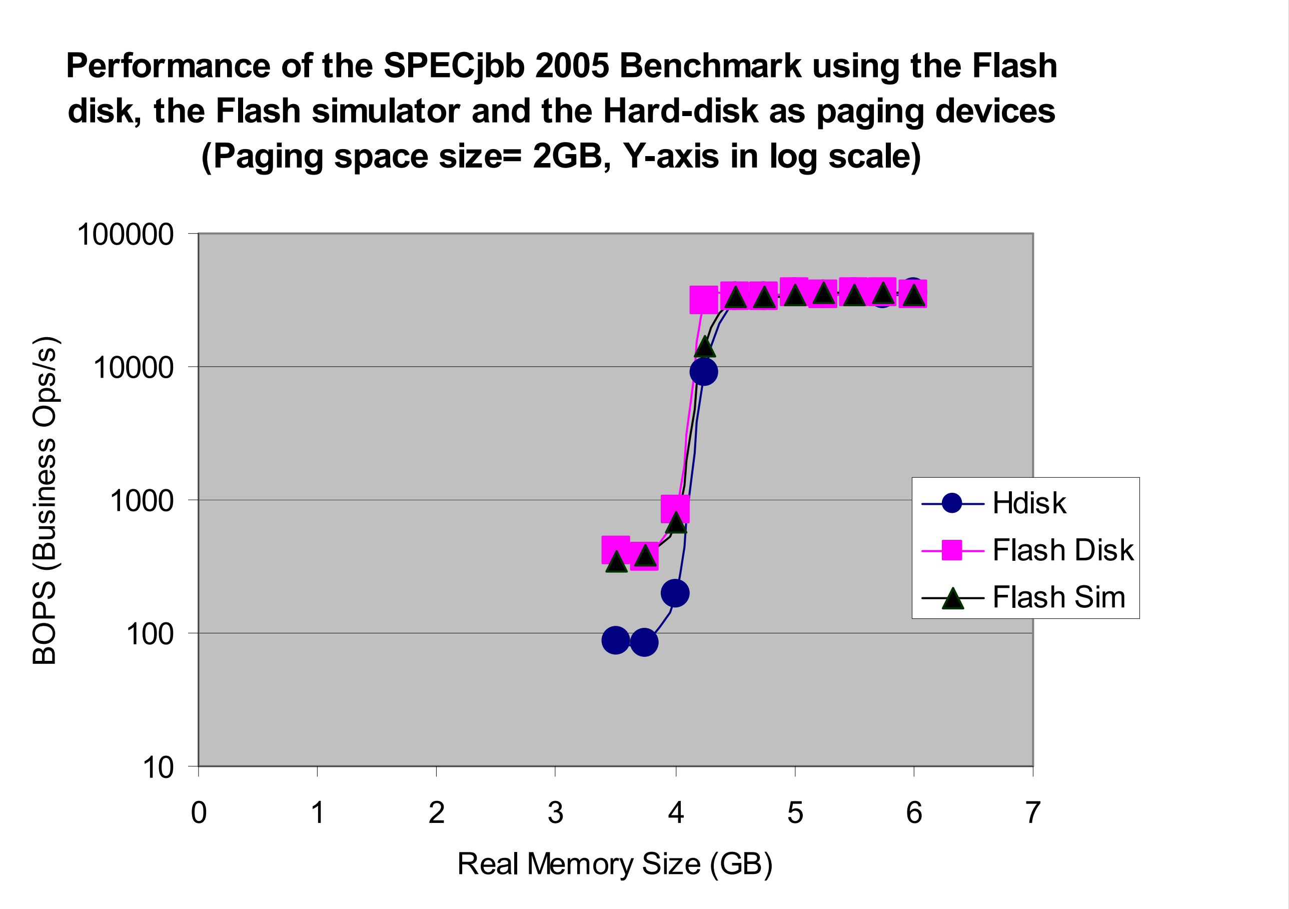
SpecJbb 2005 Results

SPECJbb 2005 benchmark from the Standard Performance Evaluation Corporation (SPEC) is based on the TPC-C benchmark specifications. It emulates a 3-tier system in a JVM with emphasis on the middle tier.

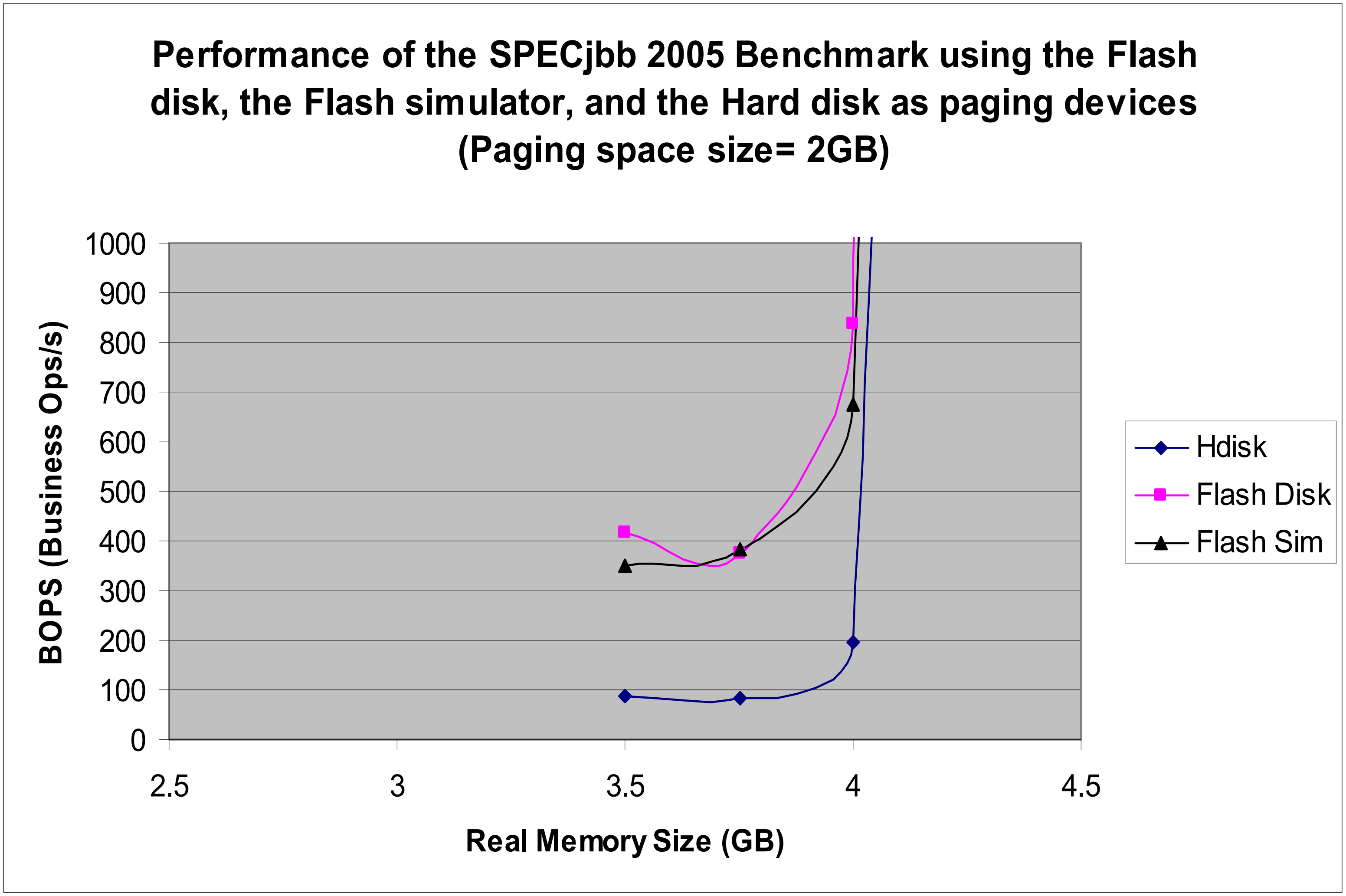
SPECJbb was executed with 10 warehouses

Each experiment was run for 240 seconds

The main metric of measurement was BOPS (Business Operations Per Second)


Performance of the SPECjbb 2005 Benchmark using the Flash disk, the Flash simulator and the Hard-disk as paging devices (Paging space size= 2GB, Y-axis in log scale)

SpecJbb 2005 Results



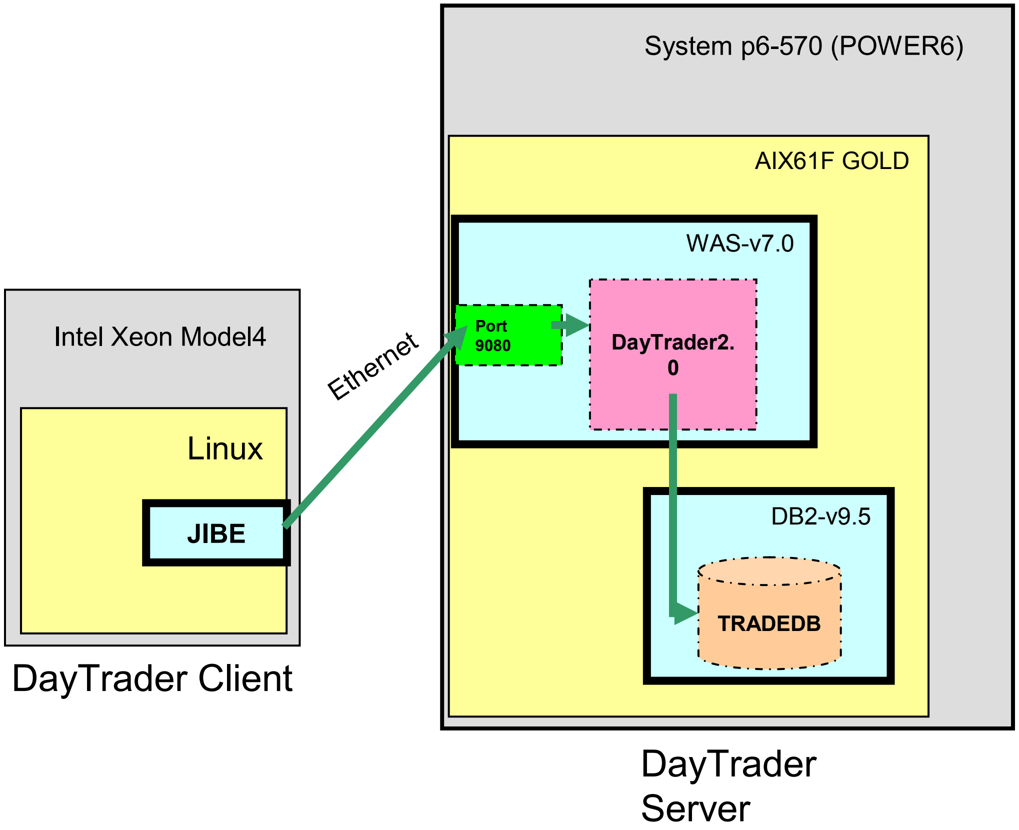
Day Trader Application Setup

DayTrader is a Websphere and DB2 benchmark application that emulates an online Stock trading system.

- Allows users to perform typical trading operations such as login, viewing portfolios, looking up stock quotes, and buying or selling stock shares.
- Several Web-based load drivers provide realistic workload scenarios that drive the application.

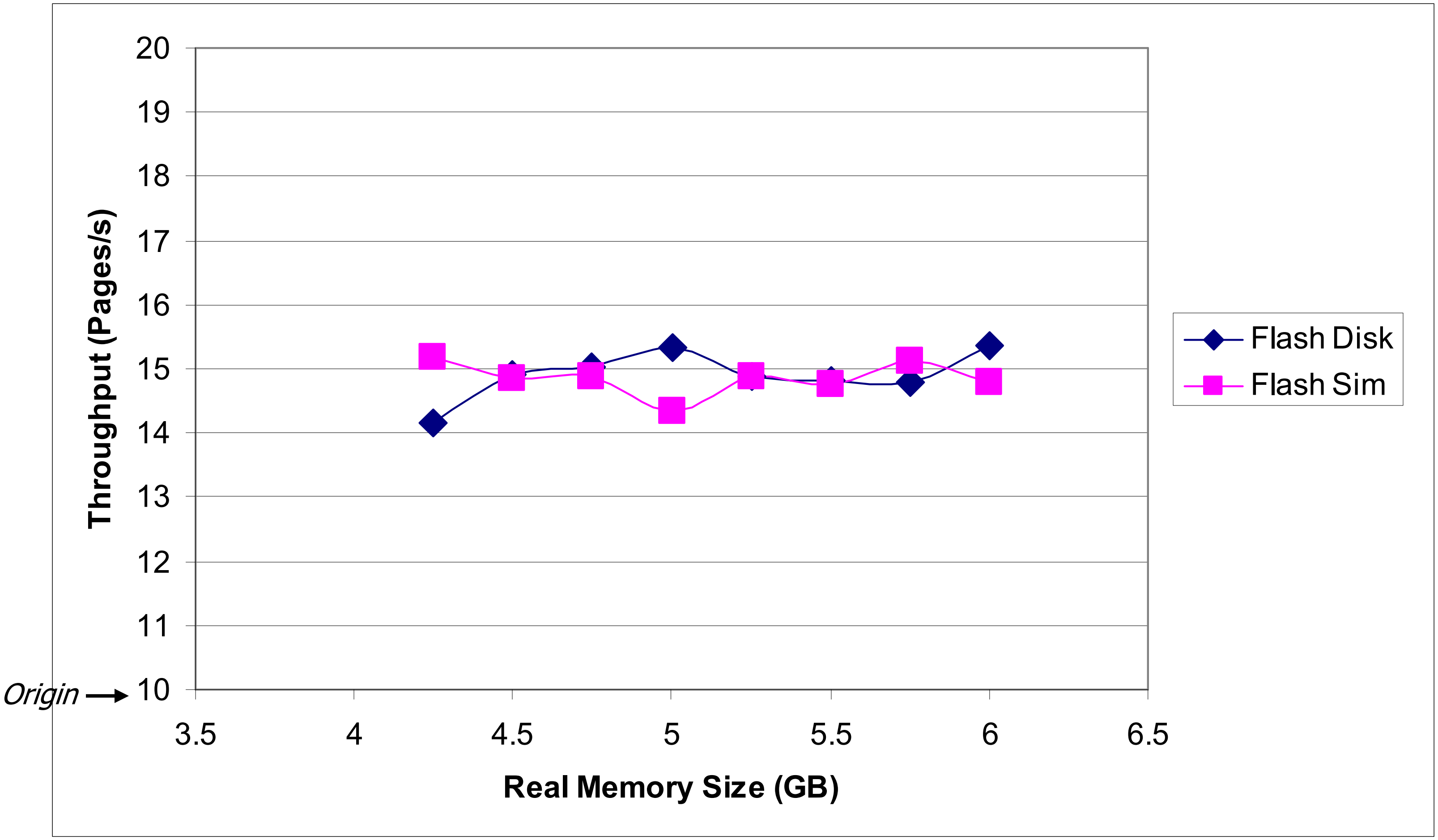
Setup:

- Stressed memory environment to trigger paging to the Flash disk or the Flash simulator.
- Simulated 500 clients
- Each client sent various types of requests (login, query account, update account; get portfolio, quote; buy, sell, etc.) to the server.



Performance Metrics:
- *Throughput* : Number of web pages serviced per second
- *Response time* (seen by the end user for each request).

Day Trader Results – Throughput



Throughput of the DayTrader Application While Paging to Flash Disk and Flash Simulator (Paging Space Size = 2GB).

Contributions

- A model for Throughput (and read/write transfer time) of a Flash device for sequential and random access patterns
  - To our knowledge, this is the first model being proposed for internal workings of a Flash device
  - The model is validated on STEC Zeus Solid-state Disk and model parameters for the Zeus SSD are extracted
  - A method to extract model parameters using microbenchmarks is presented

- A technique to build an efficient simulator using the model and extracted parameters
  - A key feature of this technique is that it maintains minimal state information and simulates delays using the {*block size; sequentiality; R/W*} properties of an I/O request

- A simulator is built as a kernel extension in the AIX operating system
  - This simulator can be configured in a matter of minutes and commercial/scientific applications can be run without any change or trace-collection

- The simulator is validated using raw-I/O and commercial workloads.

Conclusions

- Flash devices can be modeled using Linear Models
  - Unlike hard disks, Flash devices do not have rotational delays

- Flash simulators can be built using such models without simulating every minor detail and internal organization of a Flash device.
  - A throughput-based simulation of applications is within 7% error range compared to a real Flash disk
  - We suggest conducting the parameter extraction process a few times during the lifetime of the Flash disk

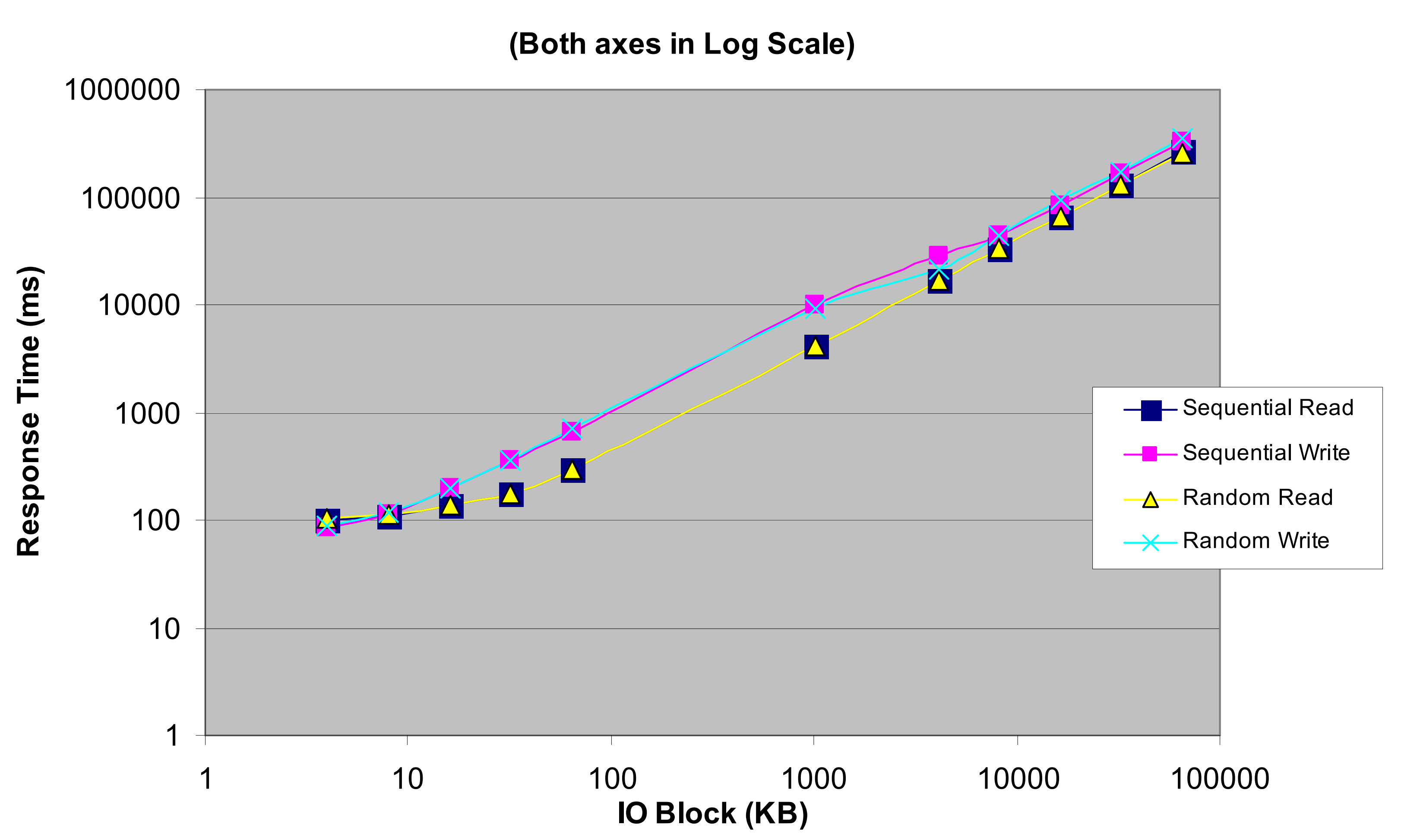
- Work is underway to improve this model
  - To incorporate layout and endurance related statistics and metrics and to derive them from a real Flash disk
  - Further validate this using a much wider set of benchmarks

THANK YOU!

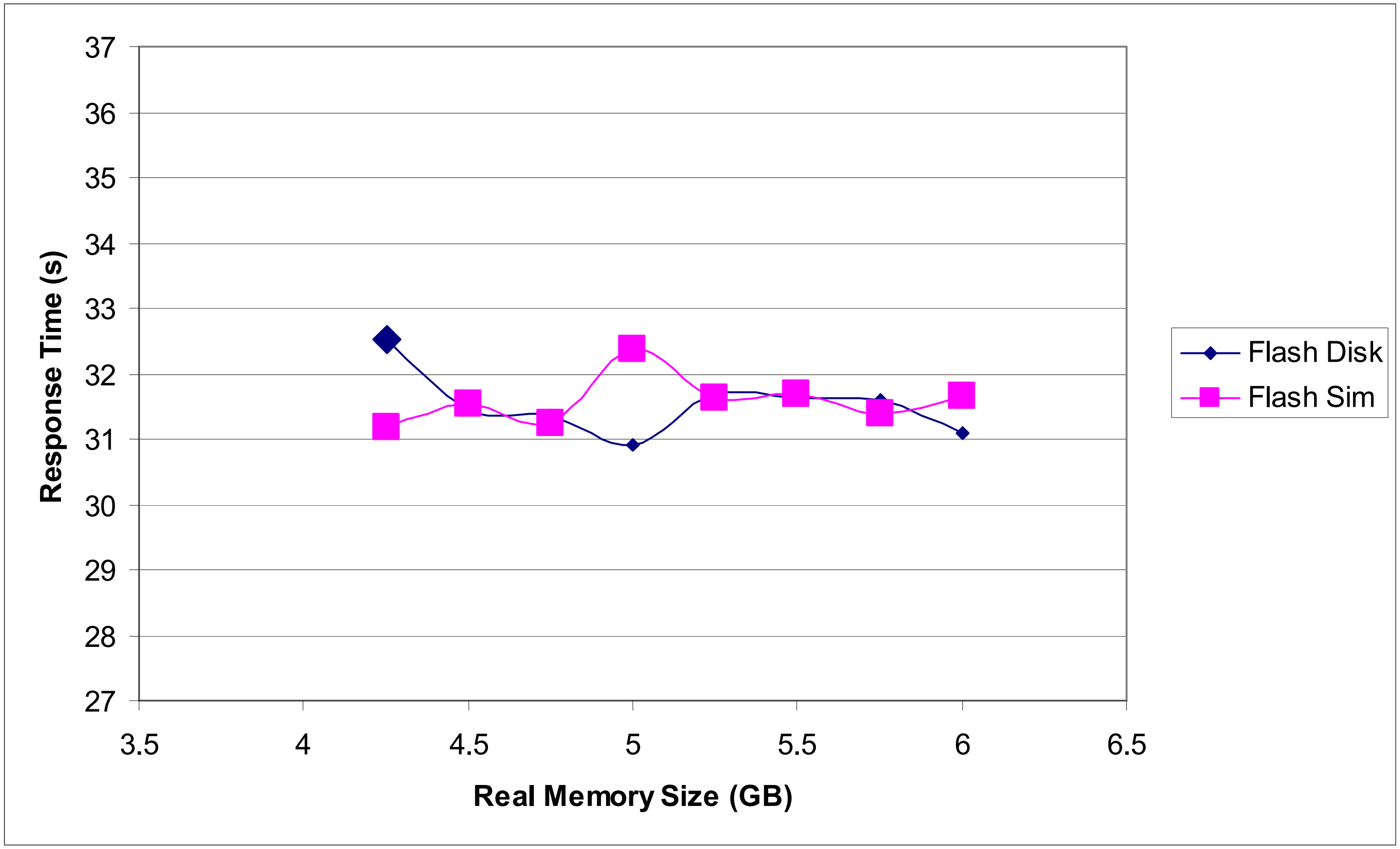
QUESTIONS?

BACKUP

Response Time of the Zeus Flash Disk



Day Trader Results – Response Time



Response Time of the DayTrader Application While Paging to Flash disk and the Flash Simulator (Paging Space Size = 2GB).