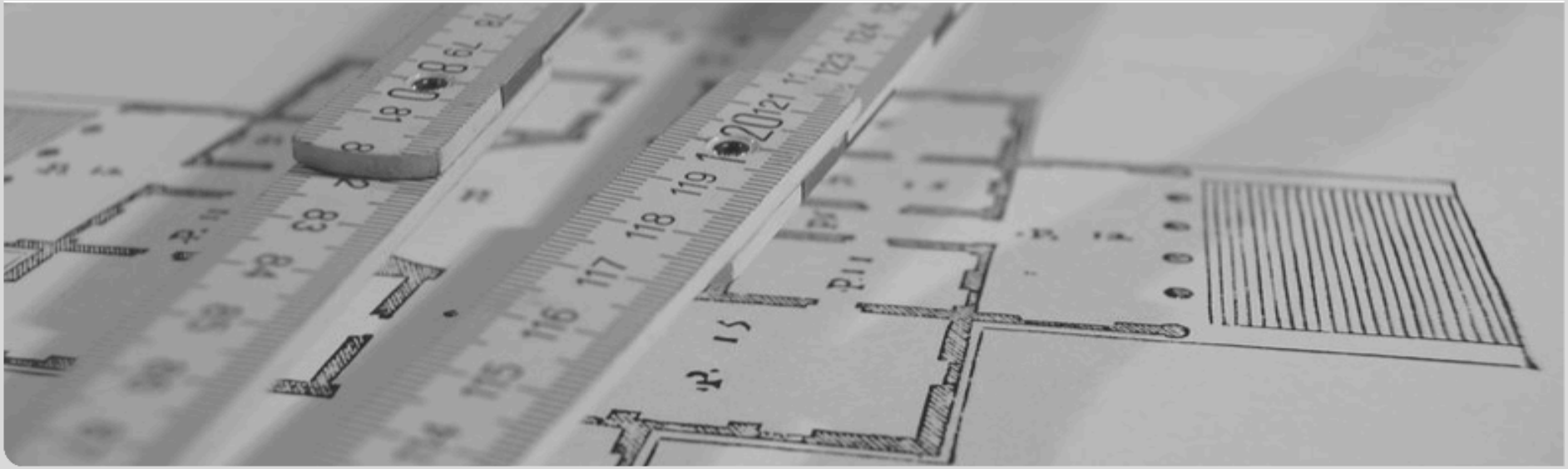
Automated Workload Characterization for I/O Performance Analysis in Virtualized Environments

Axel Busch, Qais Noorshams, Samuel Kounev,
Anne Koziolek, Ralf Reussner, Erich Amrehn

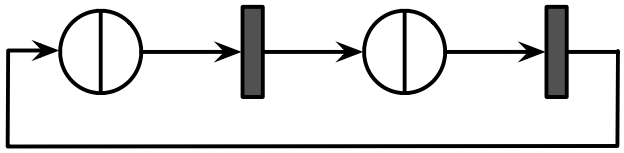
busch@kit.edu

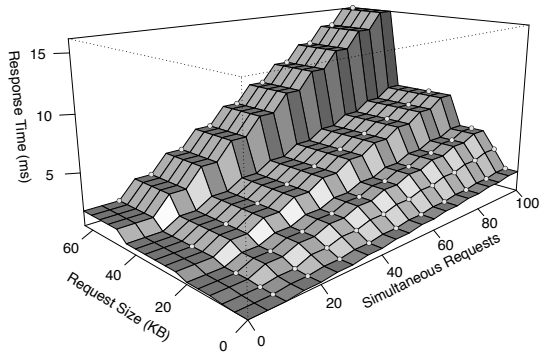
February 4th, 2015
ICPE 2015, Austin, TX

ARCHITECTURE-DRIVEN REQUIREMENTS ENGINEERING GROUP
INSTITUTE FOR PROGRAM STRUCTURES AND DATA ORGANIZATION, FACULTY OF INFORMATICS



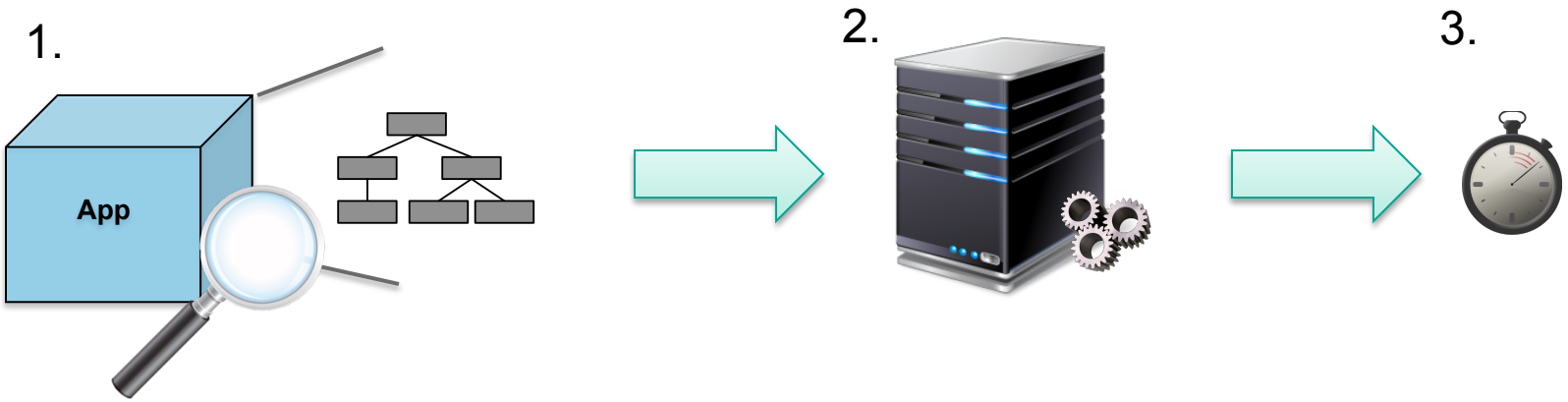
Motivation





- Many measurements to perform
- Invasive instrumentation needed
- Time consuming model development

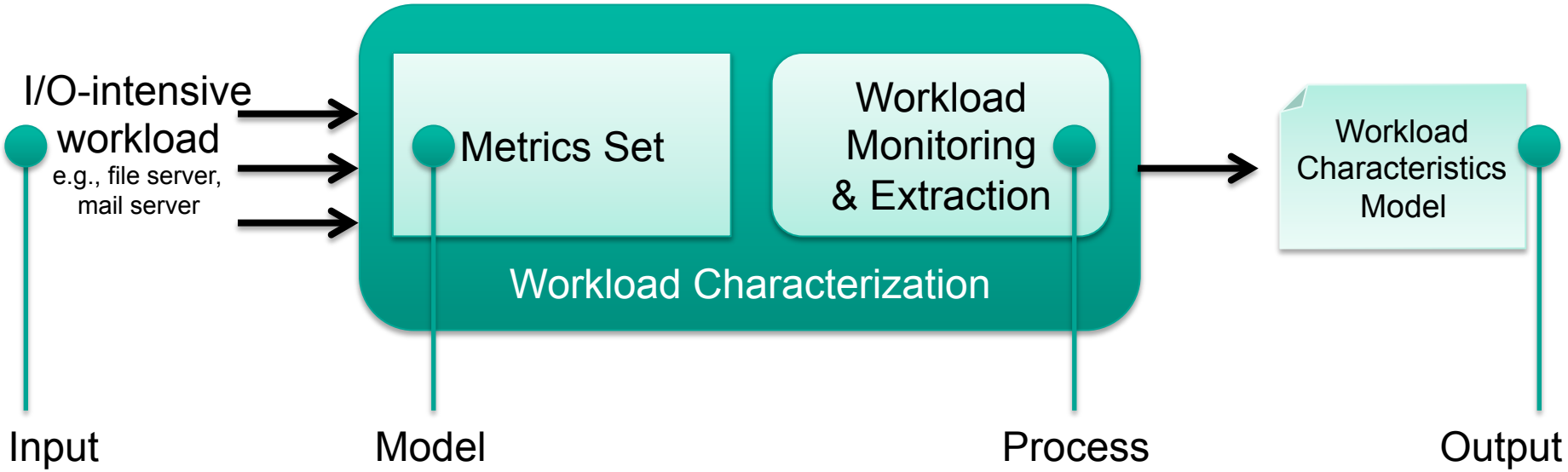
Motivation



- Fully automated workload characterization
- Lightweight approach
  - Non-invasive instrumentation
  - No need to install the full software stack
  - No need to develop any complex models by hand
- Applicable in virtualized environments
- Fast estimation of performance behaviour in typical scenarios

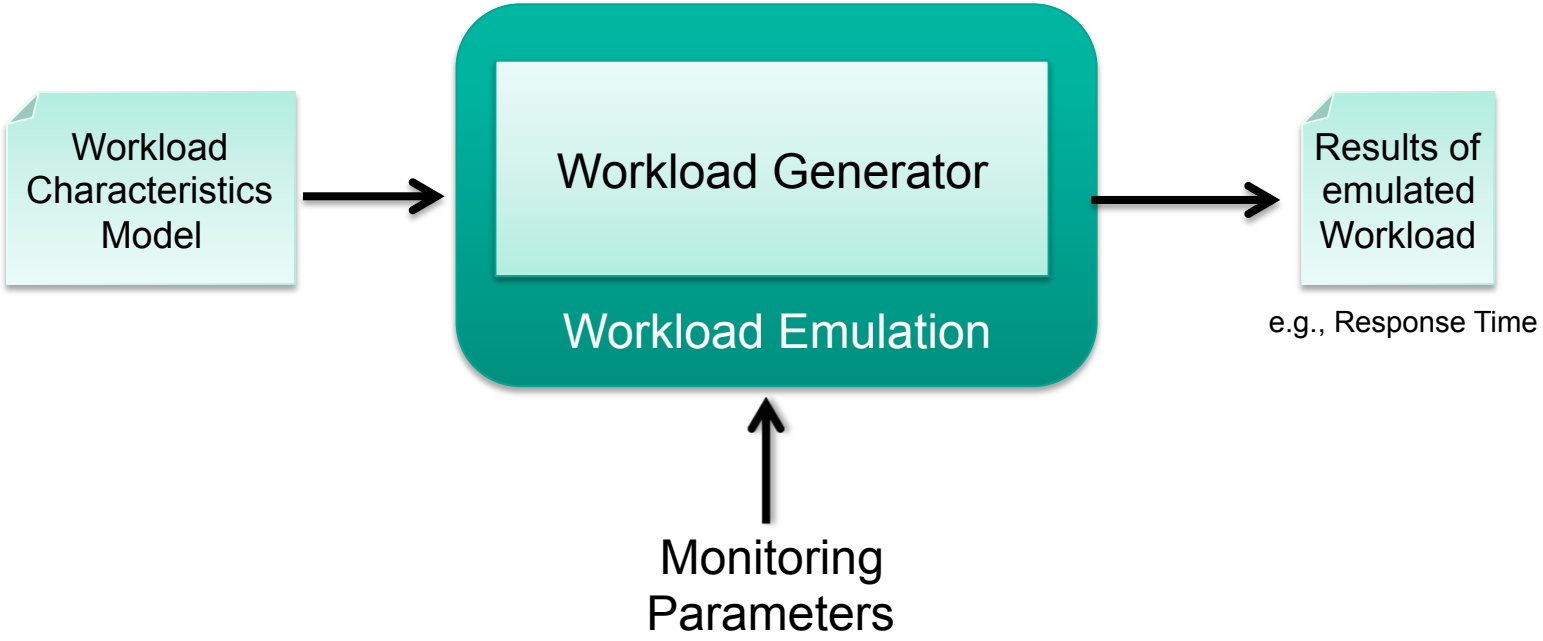
Methodology

1. Workload Characterization

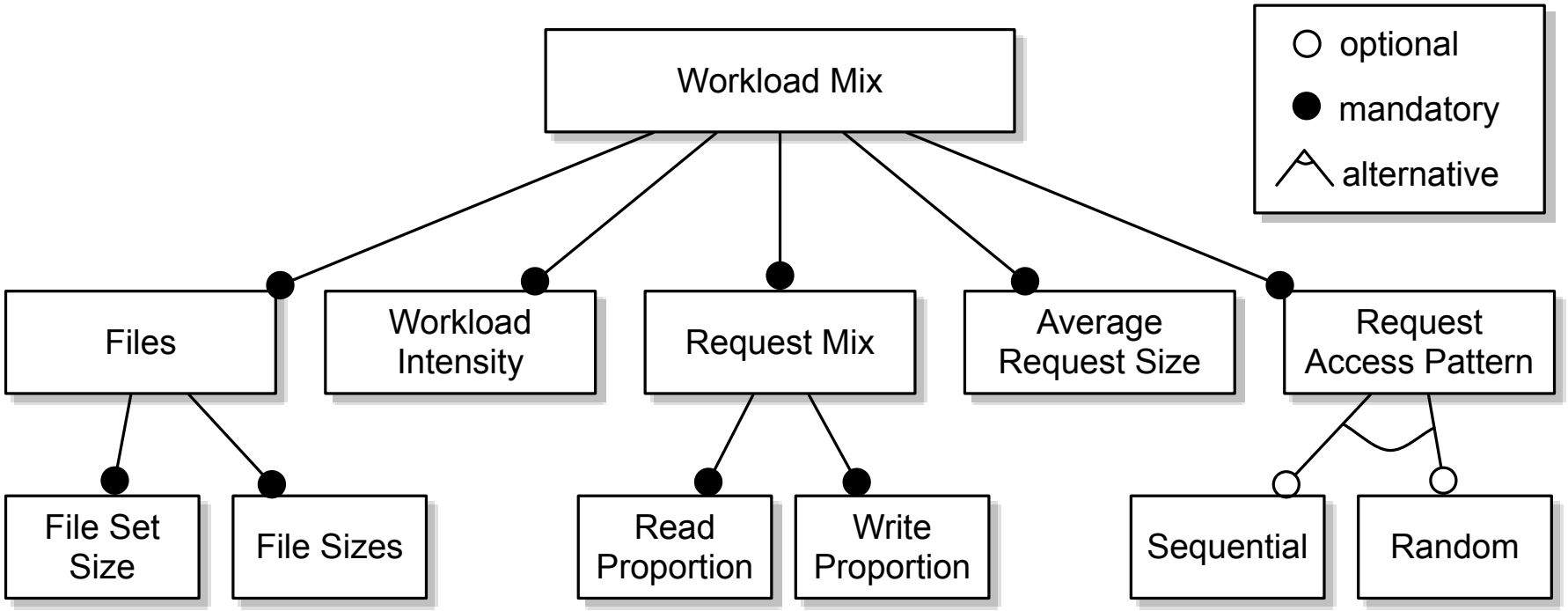


Methodology

2. Workload Emulation



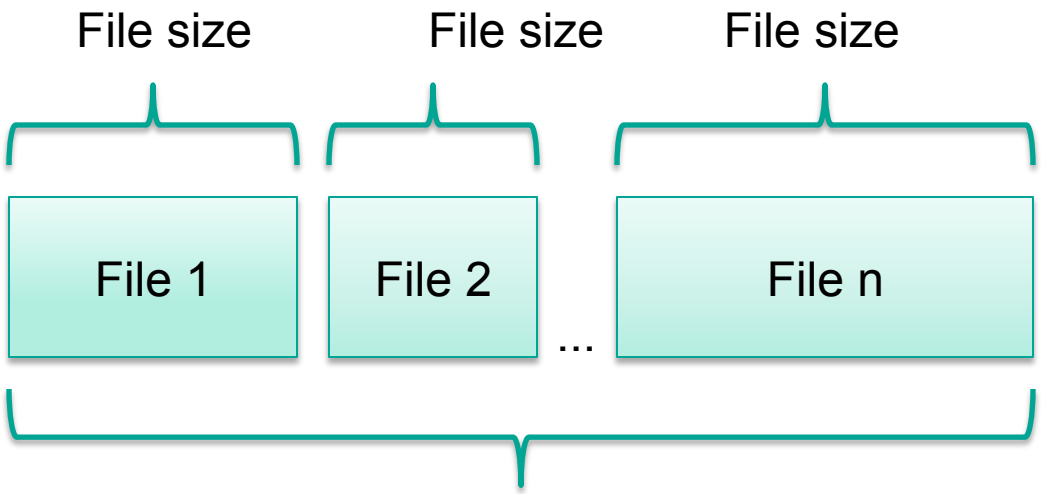
Metrics set

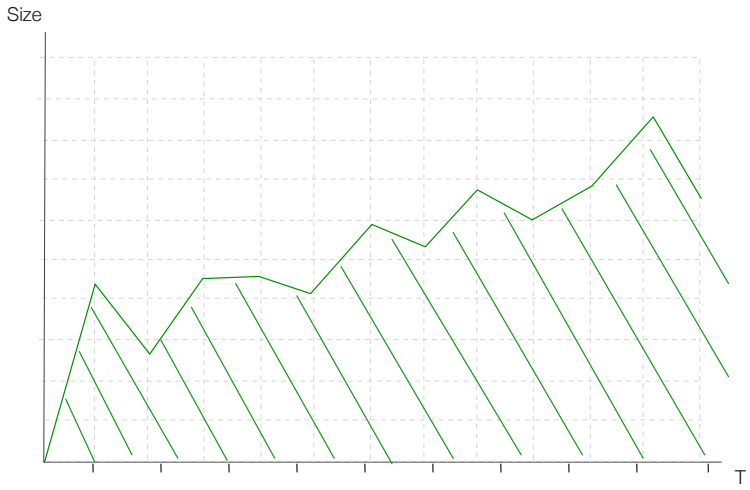


(simplified)

(Experimental Evaluation of the Performance-Influencing Factors of Virtualized Storage Systems. Q. Noorshams, S. Kounev, and R. Reussner. In EPEW '12, volume 7587 of LNCS. Springer, 2012.)

File size & File set size





$$fileSetSize^{avg} = \int_0^T \frac{\sum_{\iota=1}^{n(t)} \phi^{\iota}(t)}{T} dt$$

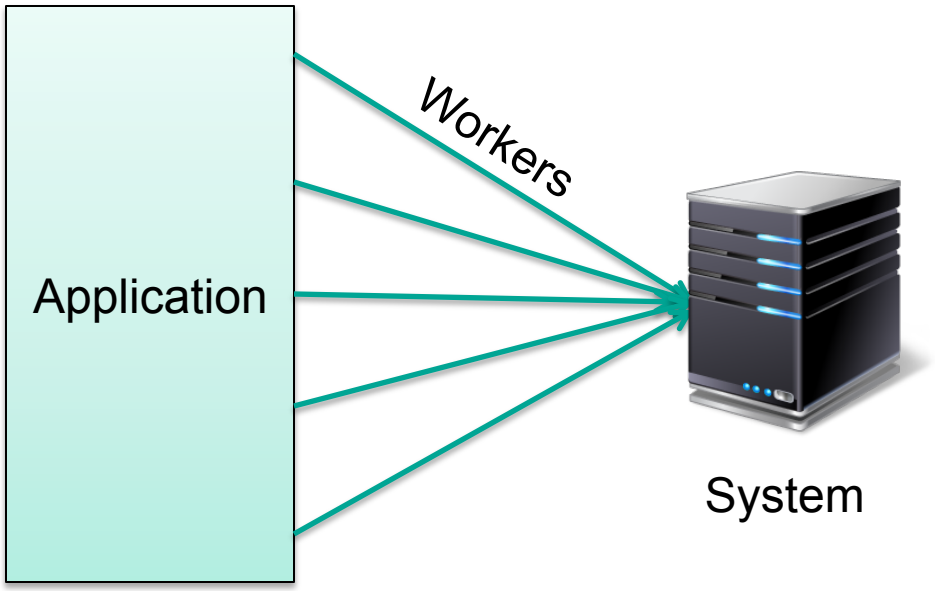
$$fileSize^{avg} = \int_0^T \frac{\sum_{\iota=1}^{n(t)} \phi^{\iota}(t)}{T \cdot n(t)} dt$$

$\phi^{\iota}(t)$: size of the ι-th file at time t

$n(t)$: number of files at time t

$[0, T], T > 0$: observation period

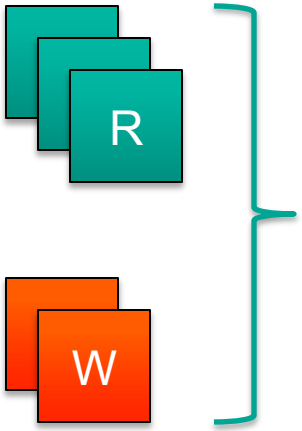
Workload Intensity



$$workloadIntensity^{avg} = \int_0^T \frac{\chi(t)}{T} dt, \chi(t) : \text{workload intensity at time } t$$
$$[0, T], T > 0 : \text{observation period}$$

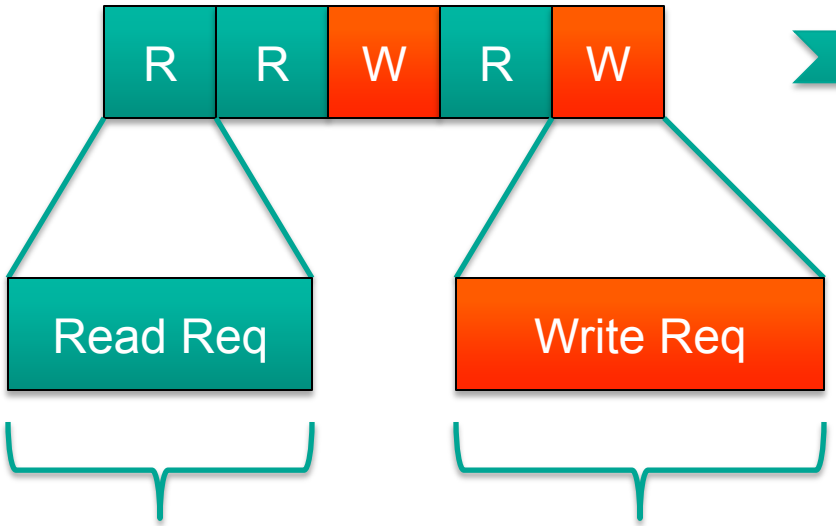
Request Mix

System



$$reqMix = \frac{\#readRequests}{\#readRequests + \#writeRequests}$$

Request Size

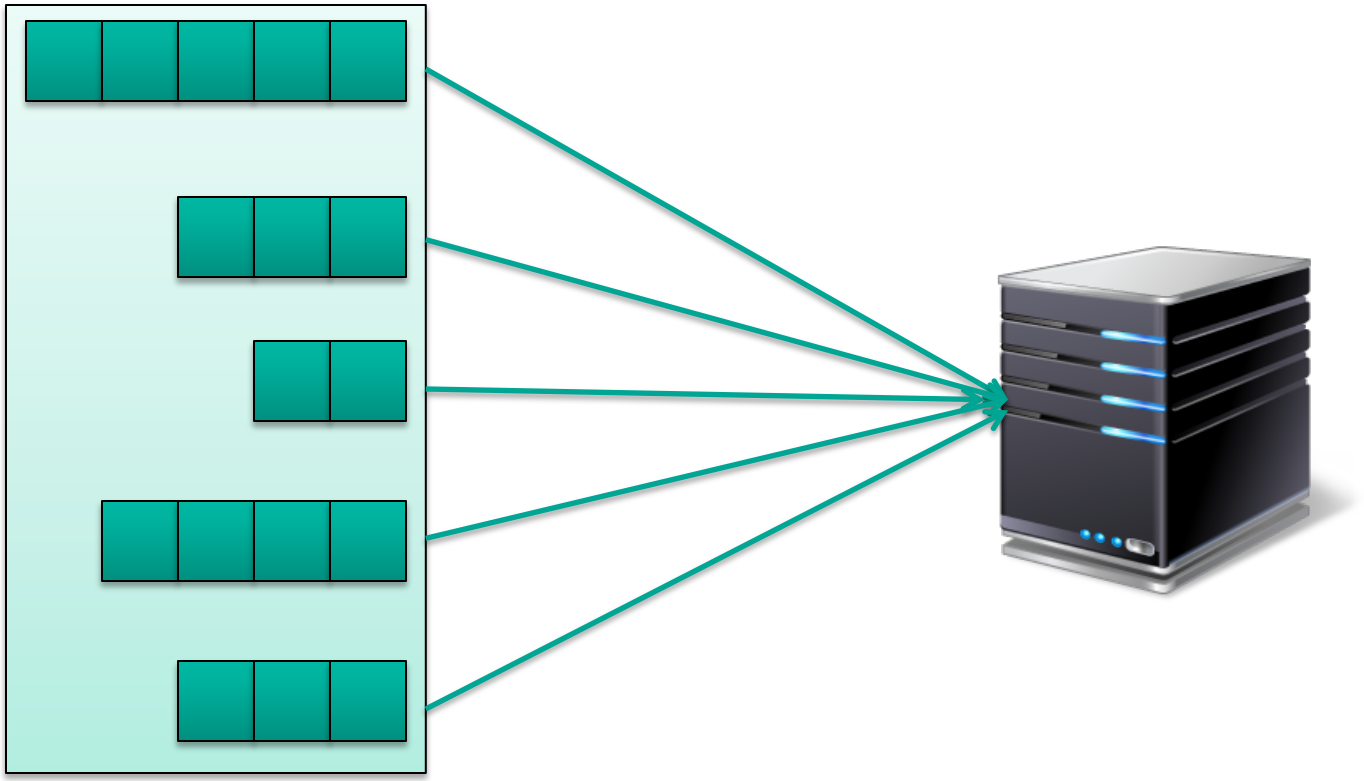


System

Request Size Read Request Size Write

$$requestSize = \frac{\sum_{j=0}^{|\Gamma|-1} \Gamma_j}{|\Gamma|}, \ \Gamma \text{ contains all observed request sizes}$$

Access Pattern



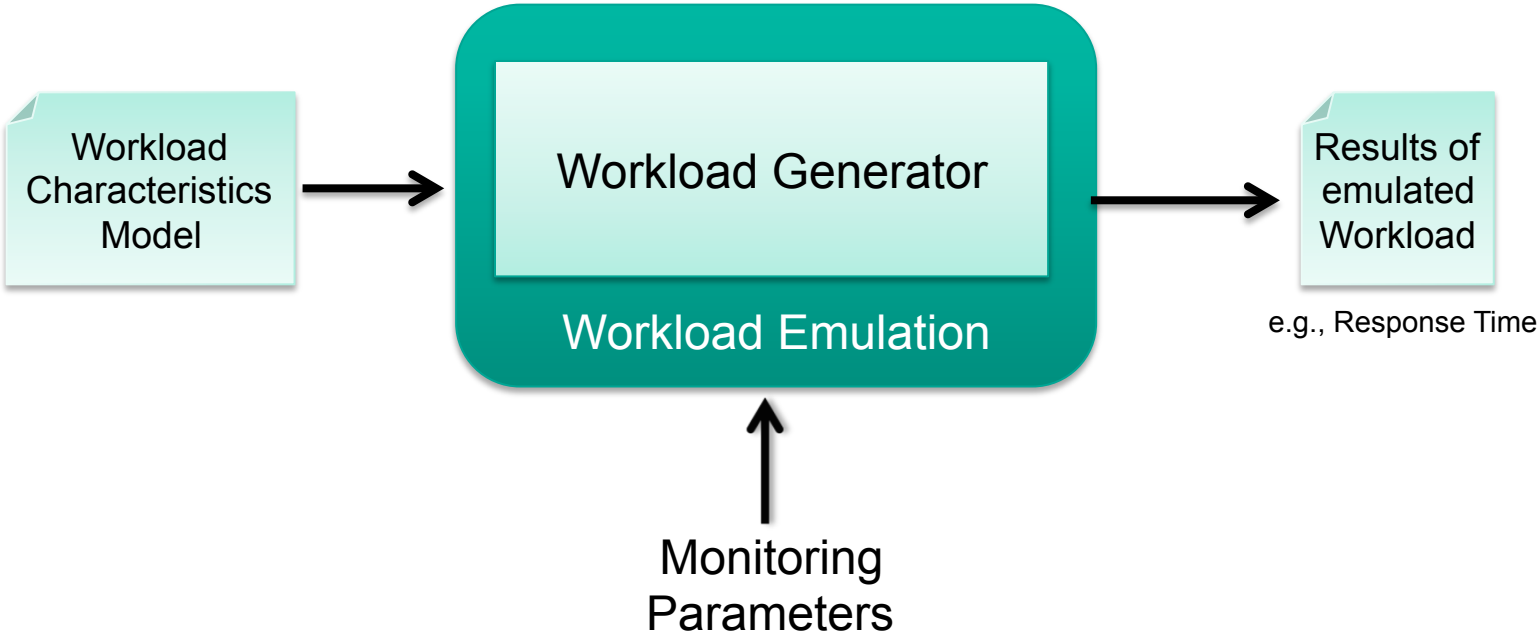
Access Pattern

1. Searching for consecutive block access
2. Counts the number of consecutive blocks
3. Results in number of consecutive block accesses in percent

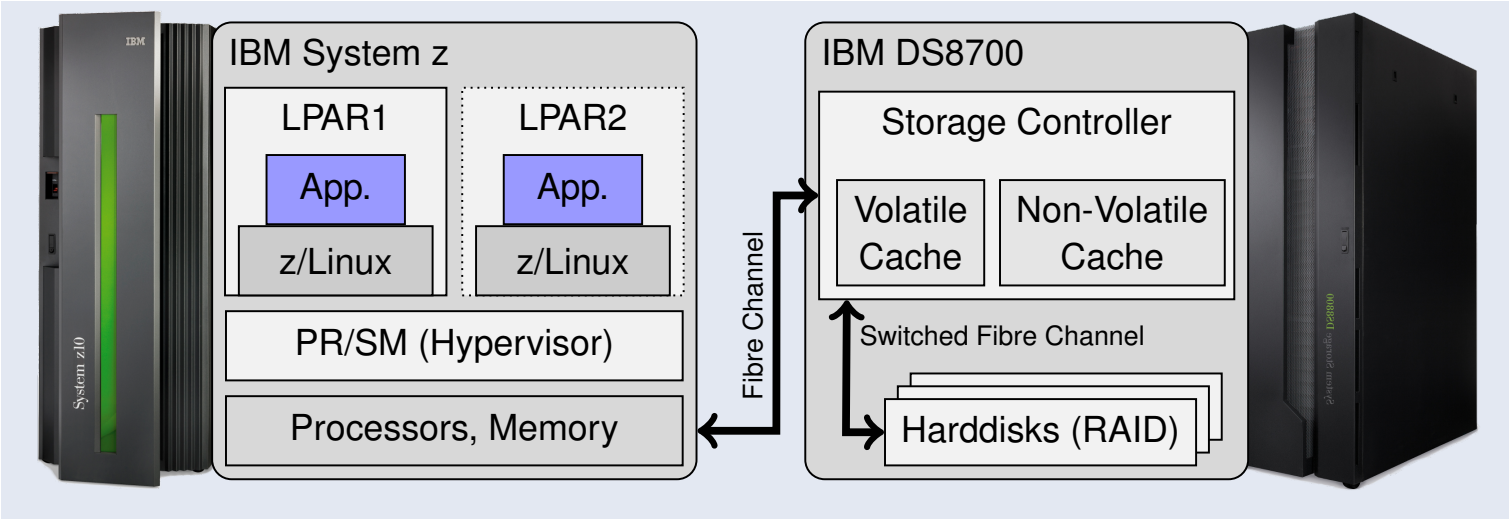
Algorithm 1 Access Pattern Recognition Algorithm

```
while i < req do                          //  Iterate through requests
    for j such that i < j < req do
        block_end = R_{i2}                // End block of request R_i
        block_start = R_{j1}              // Start block of request R_j
        if block_end = block_start then
            req_seq ← req_seq + 2         //  Count both R_i, R_j
            R ← R \ {R_i, R_j}
            continue while;
        end if
    end for
    i ← i + 1
end while
return req_seq / req
```

Application

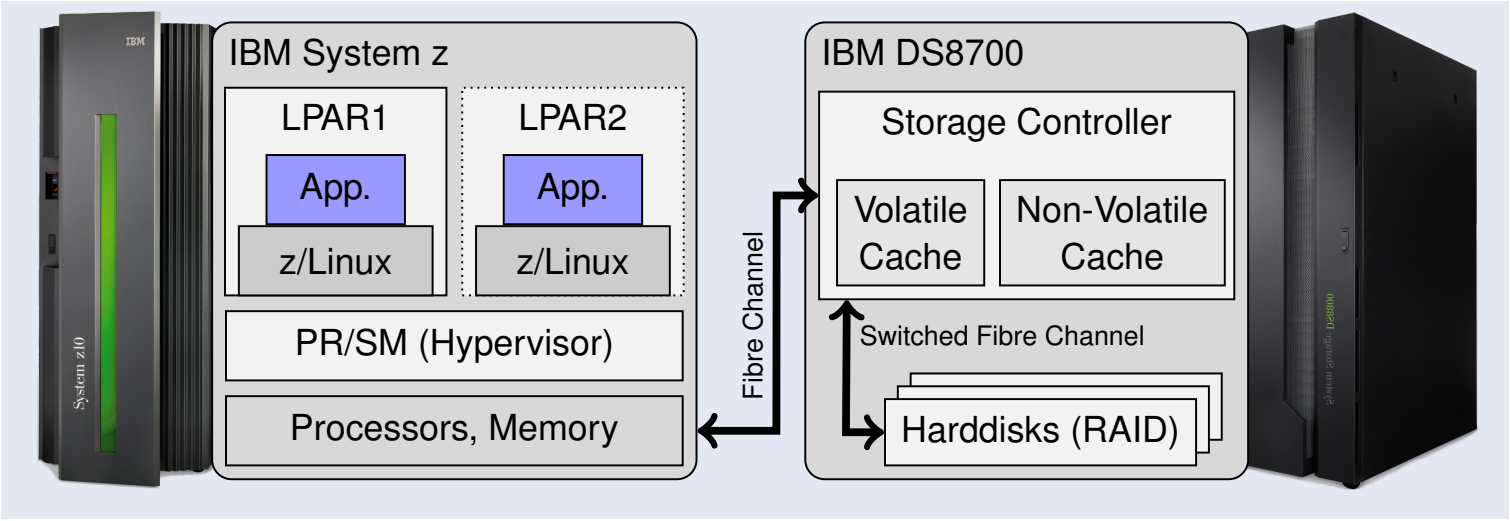


System Setup



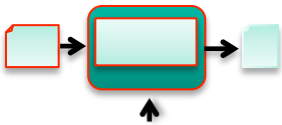
- Workload characterization performed on an IBM System z and DS8700 storage system
- Both systems represent high-end virtualized environments for critical business applications

System Setup



- System z configuration:
  - Debian z/Linux VM (=LPAR)
  - 2 IFLs (cores) ~2760 MIPS
  - 4 GB RAM
- DS8700 configuration:
  - 50 GB volatile cache (VC)
  - 2 GB non-volatile cache (NVC), i.e., battery-backed cache
  - RAID5 array with 7 HDDs (15k r/min) with 1 hot spare disk

Tools



- Filebench as storage system benchmark
- Used for generating workloads to be characterized

https://github.com/Filebench-Revise

- Flexible File System Benchmark as application layer I/O benchmark
- Used for emulating workloads

https://github.com/FFSB-Prime/ffsb

- Storage Performance Analyzer as measurement coordinator

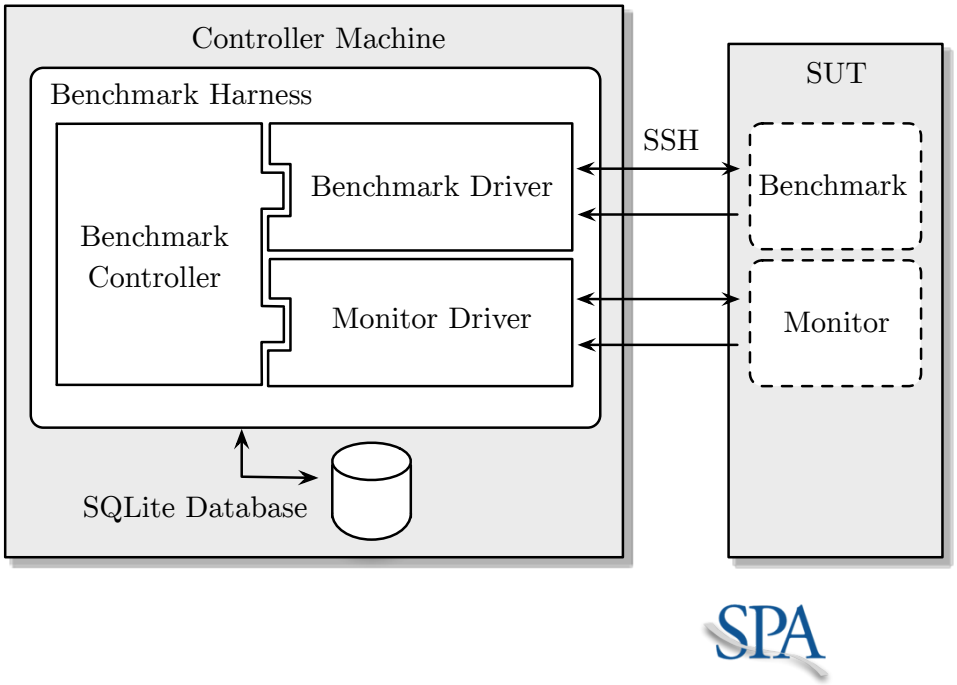
http://research.spec.org/tools/overview/spa.html

Systematic experiments

SPA extension allowing automatized

- Workload execution
- Monitoring mechanisms
- Workload characteristics extraction

} synchronized

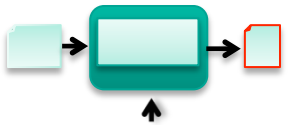


Workload characterization results [avg]:

Workload results		
	File server	**Mail server**
File size	17 KiB	130 KiB
File set size	684 MiB	1163 MiB
Workload intensity	16	50
Request mix	56 %	42 %
Request size (r)	14 KiB	103 KiB
Request size (w)	15 KiB	79 KiB
Access pattern (r)	29 %	97 %
Access pattern (w)	57 %	99 %

- Measurements performed using 1 min warm up + 20x 5 min benchmark time
- Low standard deviations

Evaluation Scenarios



Karlsruher Institut für Technologie

Workload characterization approach evaluated by two case studies

I) Workload Characterization

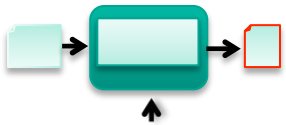
How accurate is the estimation of the workload characterization approach?

II) Scenarios

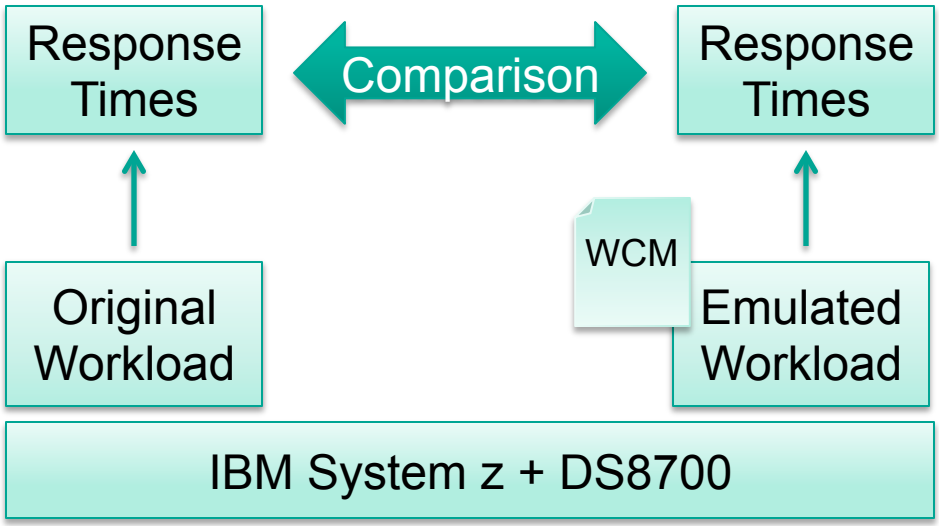
a) Migration scenario
How accurate is the estimation in migration scenarios?

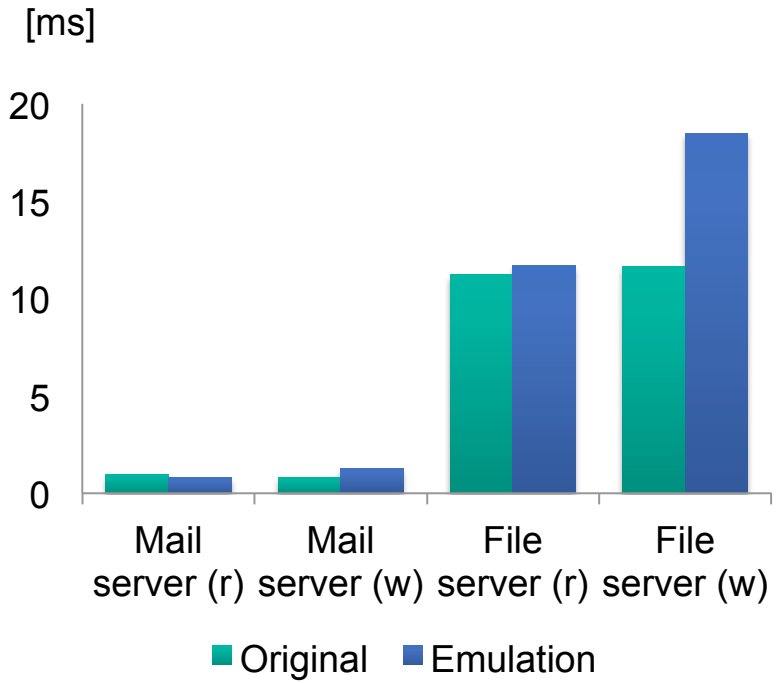
b) Consolidation scenario
How accurate is the estimation in consolidation scenarios?

Evaluation: Estimation Accuracy



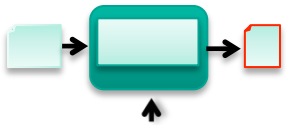
How accurate is the estimation of the workload characterization approach?



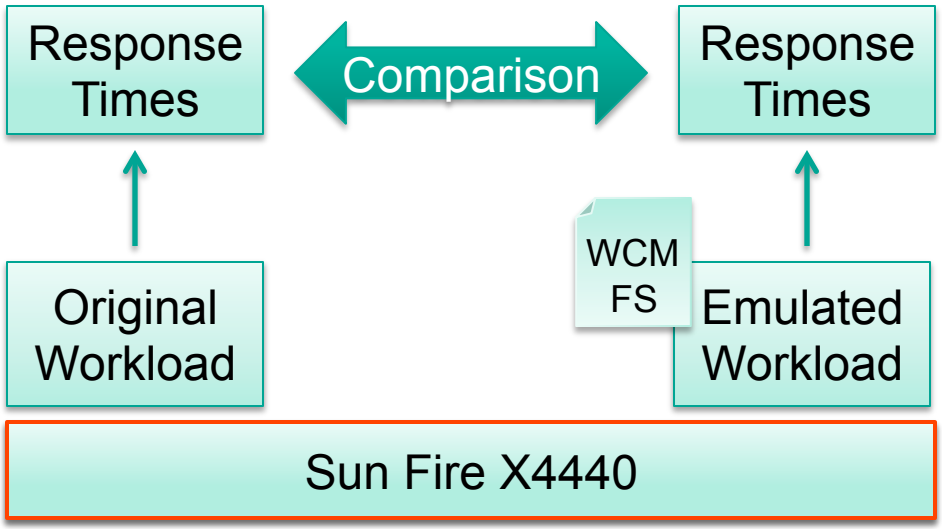


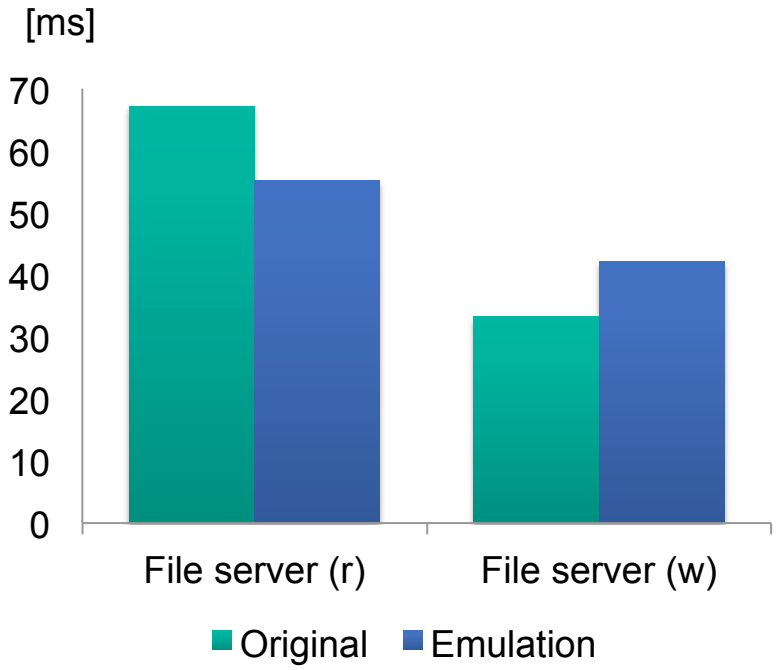
	Read error	Write error
Mail server	20.82 %	35.72 %
File server	3.93 %	36.96 %

Evaluation: Migration Scenario



How accurate is the estimation in migration scenarios?

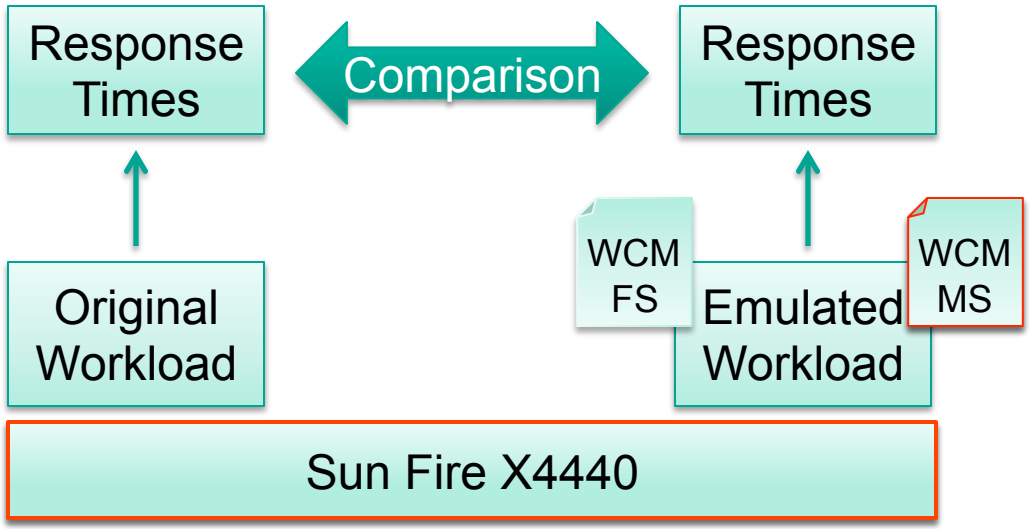


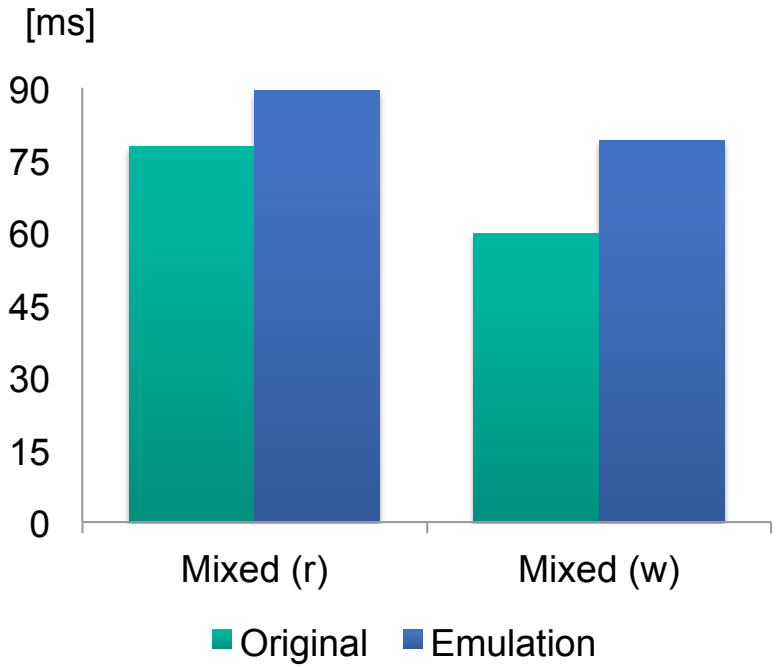


	Read error	Write error
File server	21.59 %	20.98 %

Evaluation: Consolidation Scenario

How accurate is the estimation in consolidation scenarios?





	Read error	Write error
Mixed	12.95 %	24.51 %

Evaluation: Summary

- ☑ Estimation Accuracy
- ☑ Migration Scenario
- ☑ Migration + Consolidation Scenario

Accurate results for a fast initial performance estimation in typical scenarios

Conclusion

Summary

- Fully automated approach to derive a workload characteristics model
- Capturing I/O performance-relevant workload parameters using a formalized metrics set
- Approach applied in real-world scenarios using state-of-the art virtualization hardware

Evaluation results

- Promising accuracy for fast initial performance estimation
- Migration and consolidation scenarios show low error rates < 25 %

Future Work

- Using workload characterization approach as a basis for other performance models.
- Applying scenarios when interpolate and extrapolate workload parameters