

# LibReDE

A Library for Resource Demand Estimation

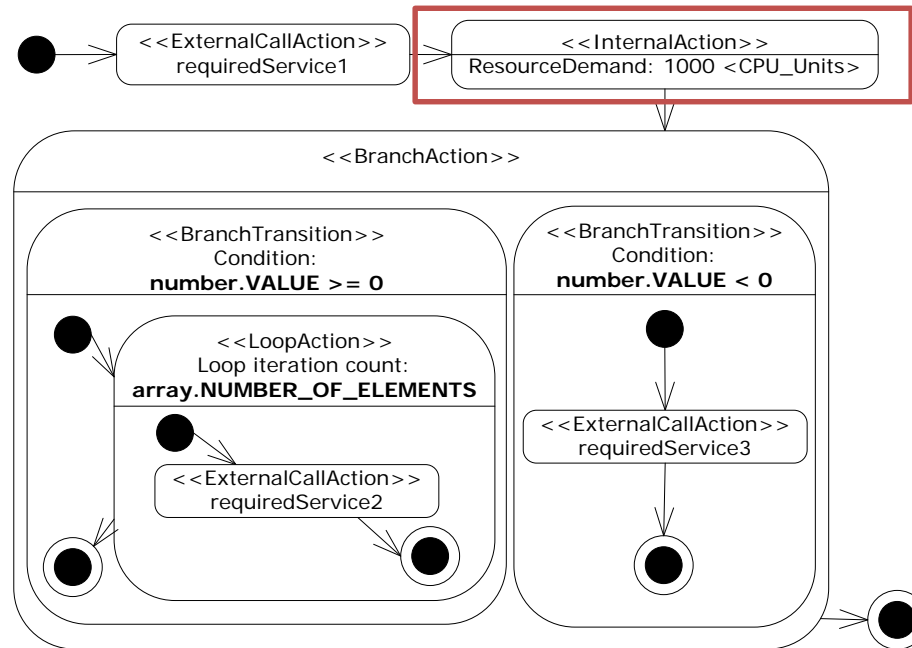
Simon Spinner, Jürgen Walter

*Dept. of Computer Science, University of Würzburg*

Symposium on Software Performance,  
Nov 27<sup>th</sup> 2014, Stuttgart, Germany

# What are resource demands?

Example SEFF in PCM:



A **resource demand** is the time a unit of work (e.g., request or internal action) spends obtaining service from a resource (e.g., CPU or hard disk) in a system.

## Direct Measurement

---

Requires specialized infrastructure to monitor low-level statistics.

Examples:

- TimerMeter [3] + ByCounter [2]
- Brunnert et al. [4]
- Magpie [1]

## Statistical Estimation

---

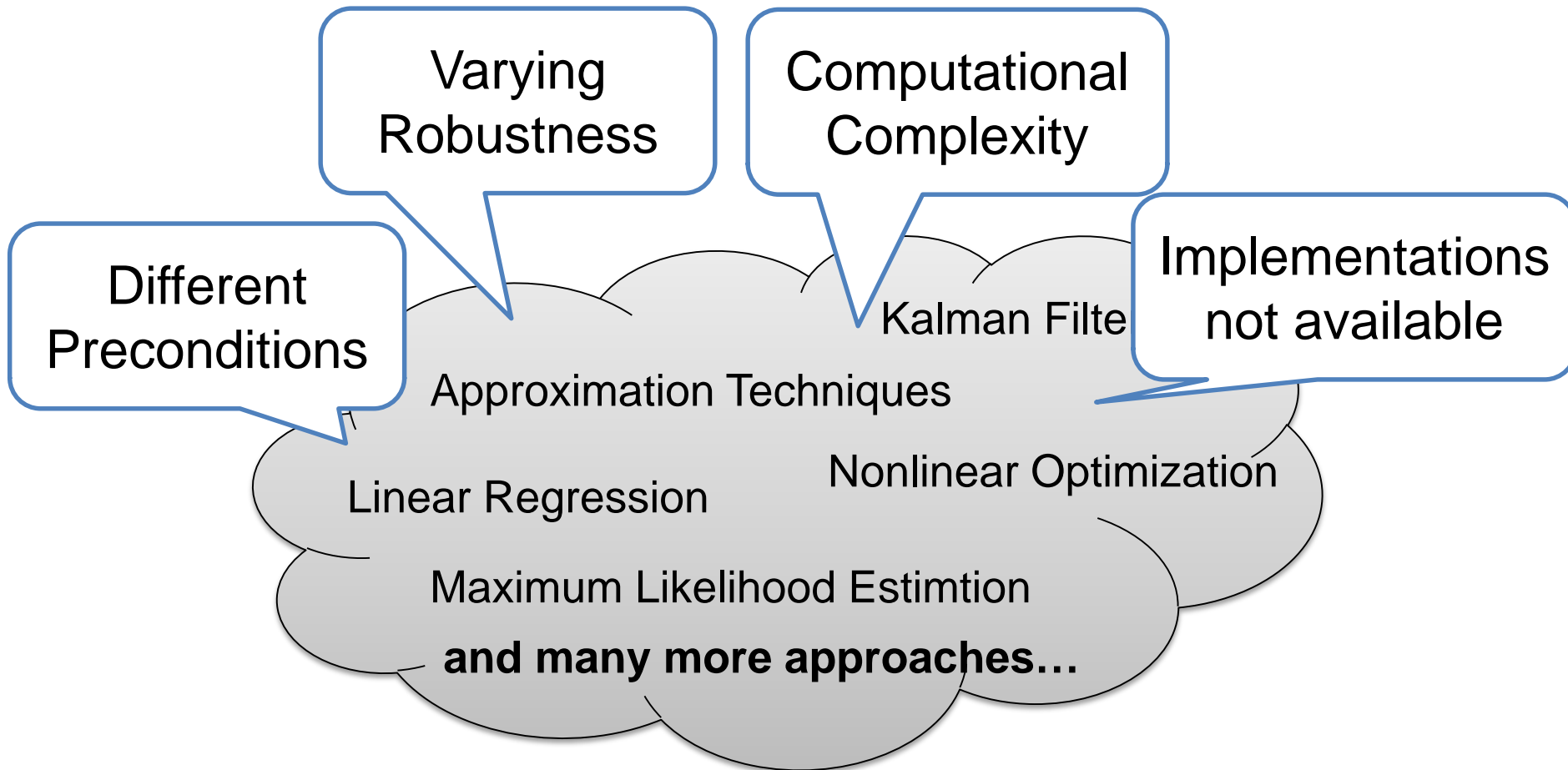
Use of statistical techniques on high-level monitoring statistics.

Examples:

- Linear regression [5-8]
- Kalman filtering [9-11]
- Nonlinear optimization [12-14]
- Maximum likelihood estimation [7] [15]
- Gibbs sampling [16]
- Independent Component Analysis [17]

# Why should I use statistical estimation?

- Direct measurements infeasible
  - Only aggregate resource usage statistics available
  - Unaccounted work in system or background threads
- Direct measurements too expensive
  - Monitoring of production system
  - Heterogeneous software stacks
- Coarse-grained models
  - Trade-off analysis speed vs. prediction accuracy
  - Usage of performance models at system runtime



**What is the best approach for a given scenario?**



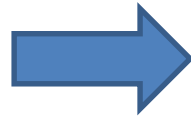
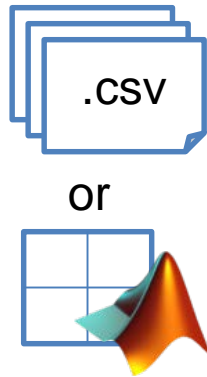
- Ready-to-use implementations of existing approaches
- Framework for implementing new approaches
- Available as open-source: <http://descartes.tools/librede>

## References

Simon Spinner, Giuliano Casale, Xiaoyun Zhu, and Samuel Kounev. LibReDE: A Library for Resource Demand Estimation (Demonstration Paper). In *Proc. of the 5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*, Dublin, Ireland, March 22-26, 2014, pages 227-228.

- Standalone version for offline analysis

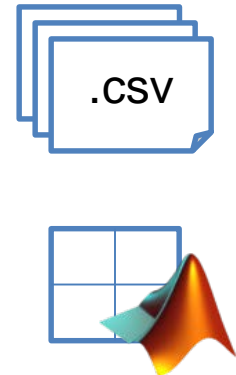
Measurement traces



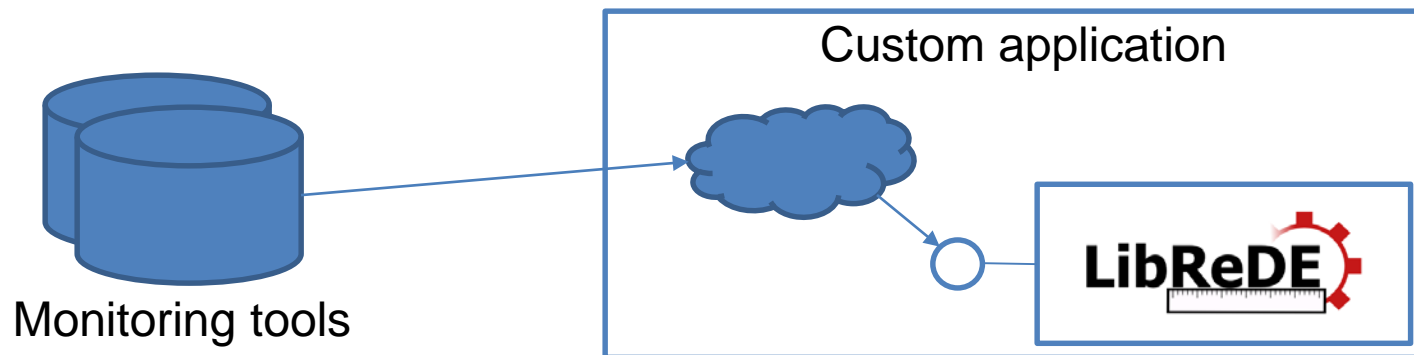
**LibReDE**

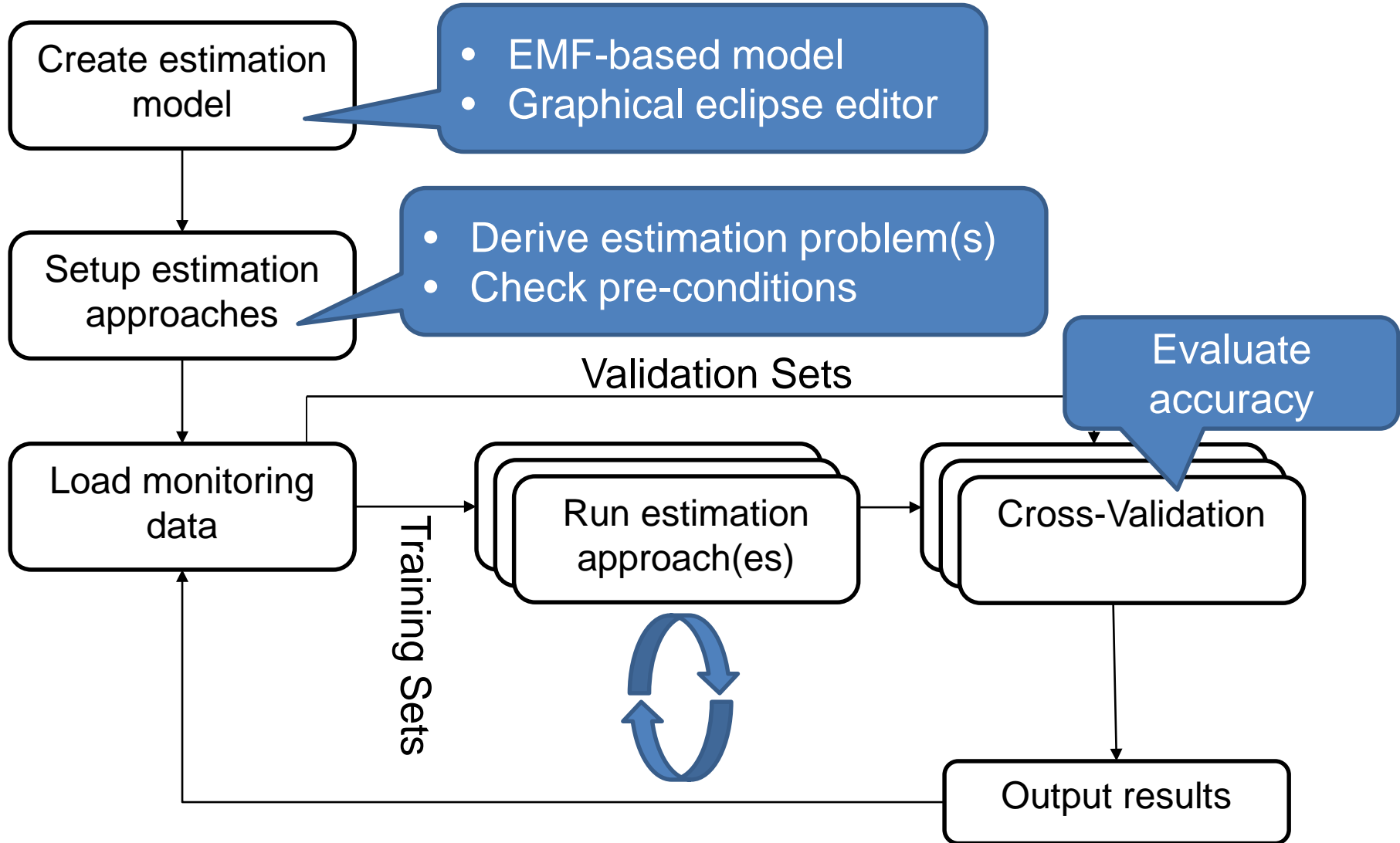


Estimated Demands



- Java library for online analysis









Demo

# MODEL EDITOR

# Step 1: Workload Description

Java - test/estimation.librede - Eclipse SDK

File Edit Navigate Search Project Librede Estimation Model Editor Run Window Help

estimation.librede

## Workload Description

**Services**

Services (or workload classes) are groups of requests with similar resource demand behaviors.

Name	Add
WC0	Remove
WC1	
WC2	

**Services/  
workload classes**

**Resources**

List all processing resources for which resource demands should be determined.

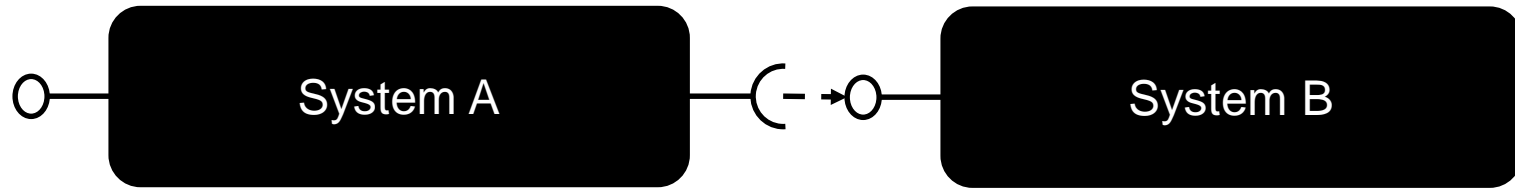
Name	Number of ...	Scheduling ...	Add
host1	1	Unkown	Remove

**Resources**

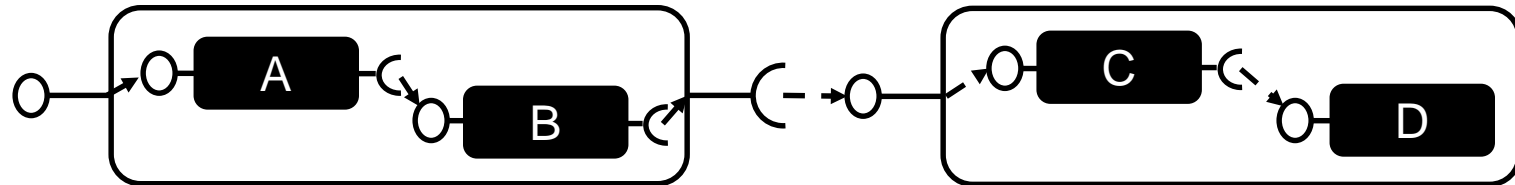
Workload Description | Data Sources | Traces | Estimation | Validation | Output

Selected Nothing

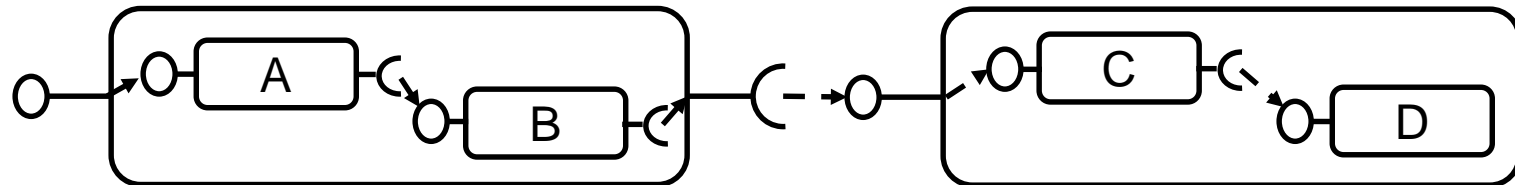
Black-box or System-level: System Entry Points  $\rightarrow$  Services

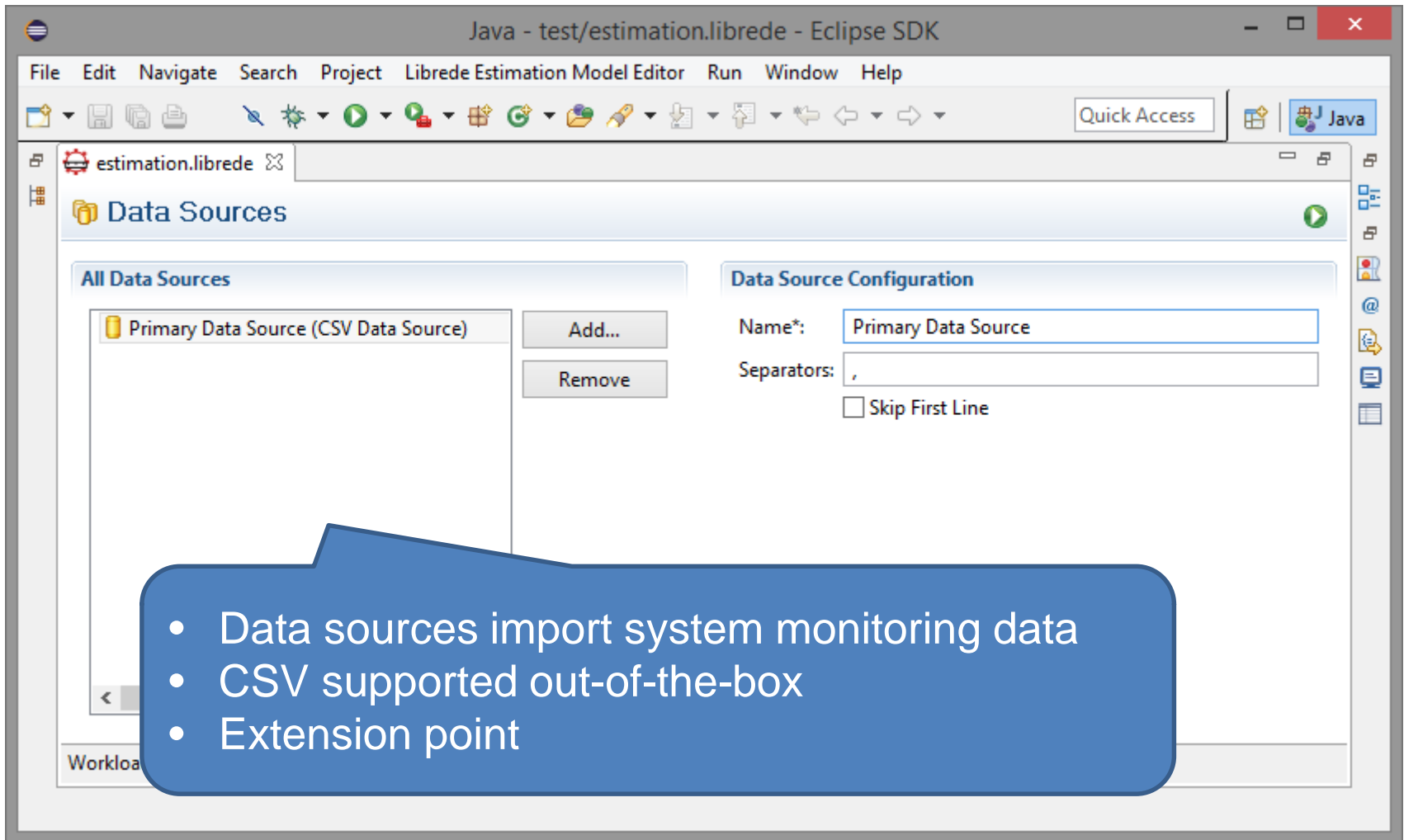


Coarse-grained: Service Operations  $\rightarrow$  Services



Fine-grained: Internal Actions  $\rightarrow$  Services





- Data sources import system monitoring data
- CSV supported out-of-the-box
- Extension point

# Step 3: Traces

Java - test/estimation.librede - Eclipse SDK

File Edit Navigate Search Project Librede Estimation Model Editor Run Window Help

estimation.librede

### Traces

**All Measurement Traces**

- experiment1\_WC0\_RESPONSE\_TIME.csv (rep... Add...
- experiment1\_WC1\_RESPONSE\_TIME.csv (rep... Remove
- experiment1\_WC2\_RESPONSE\_TIME.csv (rep...
- host1\_CPU\_UTILIZATION.csv (utilization, ag

**Measurement Trace Details**

File: C:\Users\Simon\Desktop\example1\exp... Browse...

Data Source: Primary Data Source (CSV Data Source)

Metric: Reponse Time

Interval: 0 Seconds

Mapping:

Entity	Column Index
WC0	1

Input files with monitoring data

Mapping on services/resources

Workload Description Data Sources Traces Estimation Validation Output

# Step 4: Estimation

estimation.librede

## Estimation

**Activated Estimation Approaches**

- Service Demand Law
- Approximation with Response Times
- Kalman Filter using Utilization Law
- Learning-based Estimation using Utilization Law

**Interval Settings**

Step Size: 120 Seconds

Start Date: 01.06.2013 04:52:30 Read from

In Unix Time: 1370087550000

End Date: 01.06.2013 05:48:59

In Unix Time: 1370090939000

Recursive Execution

**Estimation Algorithm Configuration**

State Noise Covariance\*: 1.0

Noise Coupling\*: 1.0

Observe Noise Covariance\*: 0.0001

Workload Description | Data Sources | Traces | Estimation | Validation | Output

- 6 estimation approaches
- Extension point

Time interval settings

- Parameters of underlying statistical techniques

Java - test/estimation.librede - Eclipse SDK

File Edit Navigate Search Project Librede Estimation Model Editor Run Window Help

estimation.librede

## Validation

▼ Cross-Validation Settings

Run k-Fold Cross-Validation

Number of Folds k: 5

All Validators

- Response Time Validator
- Utilization Law Validator

Workload Description Data Sources Traces Estimation **Validation** Output

**K-fold cross-validation**

- Validators based on Operational Laws
- Extension point

Java - test/estimation.librede - Eclipse SDK

File Edit Navigate Search Project Librede Estimation Model Editor Run Window Help

estimation.librede

Output

All Exporters

Default (CSV Export) Add... Remove

Exporter Configuration

Name\*: Default

Output Directory\*: C:\User...

File Name Prefix\*: estimates

Run estimation

- Output results to files
- CSV supported out-of-the-box
- Extension point

Workload Description Data Sources Traces Estimation Validation Output



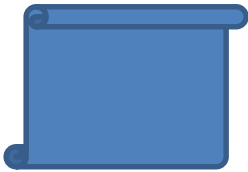


# ESTIMATION

# Estimation Approach

- Derives a set of tuples  $\langle S, O, A \rangle$
- State model  $S$ :
  - Knowledge about the values of the resource demands
  - State constraints
  - Initial value
- Observation Model  $O$ :
  - Relationship between observations and resource demands
  - E.g., Utilization Law
- Estimation Algorithm  $A$ :
  - E.g., Least-squares regression

# Example: Linear Regression with Utilization Law Approach



Workload Description:

- Resources: CPU0, CPU1, HD0
- Services: WC0, WC1, WC2

State Model 1:  
Resource: CPU0



Observation Model 1:  
Utilization Law



Least-squares  
regression

State Model 2:  
Resource: CPU1



Observation Model 2:  
Utilization law

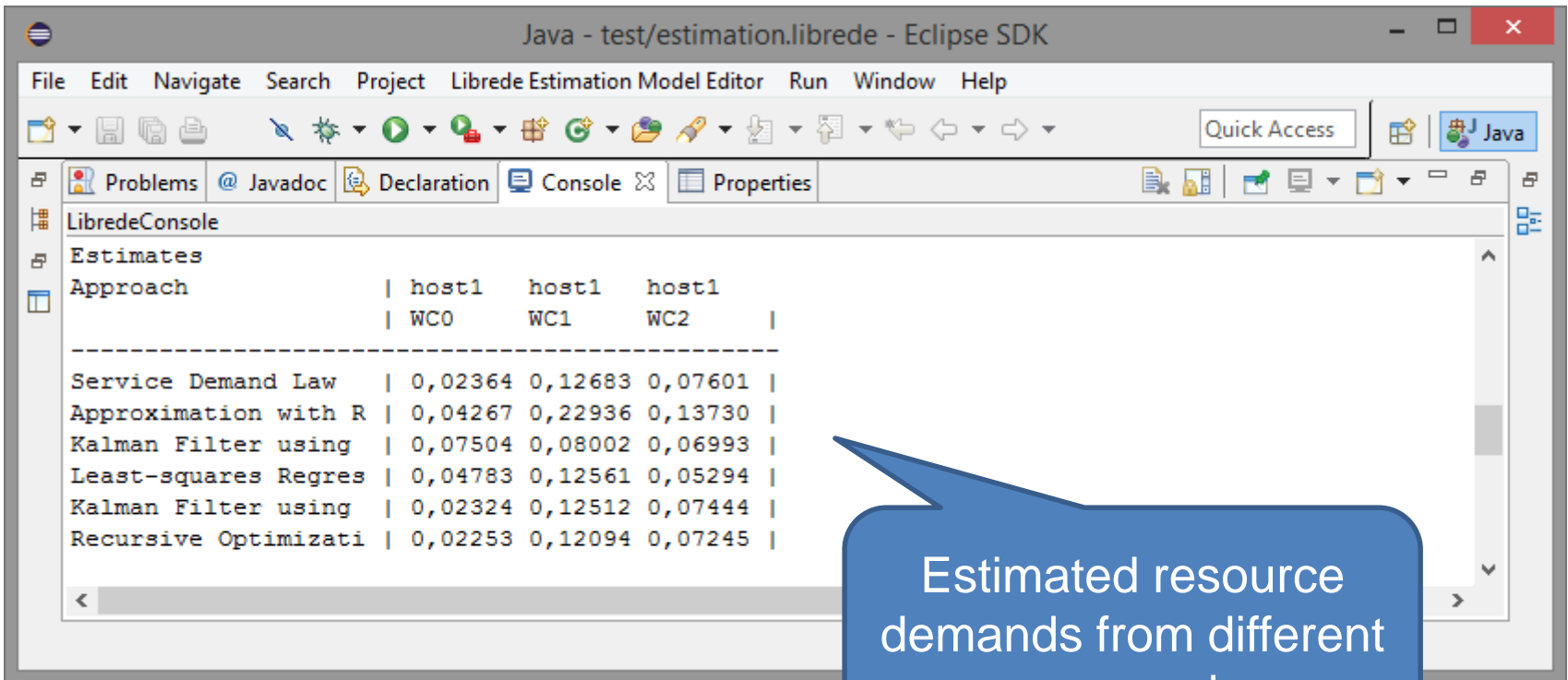


Least-squares  
regression

State Model 3:  
Resource: HD0



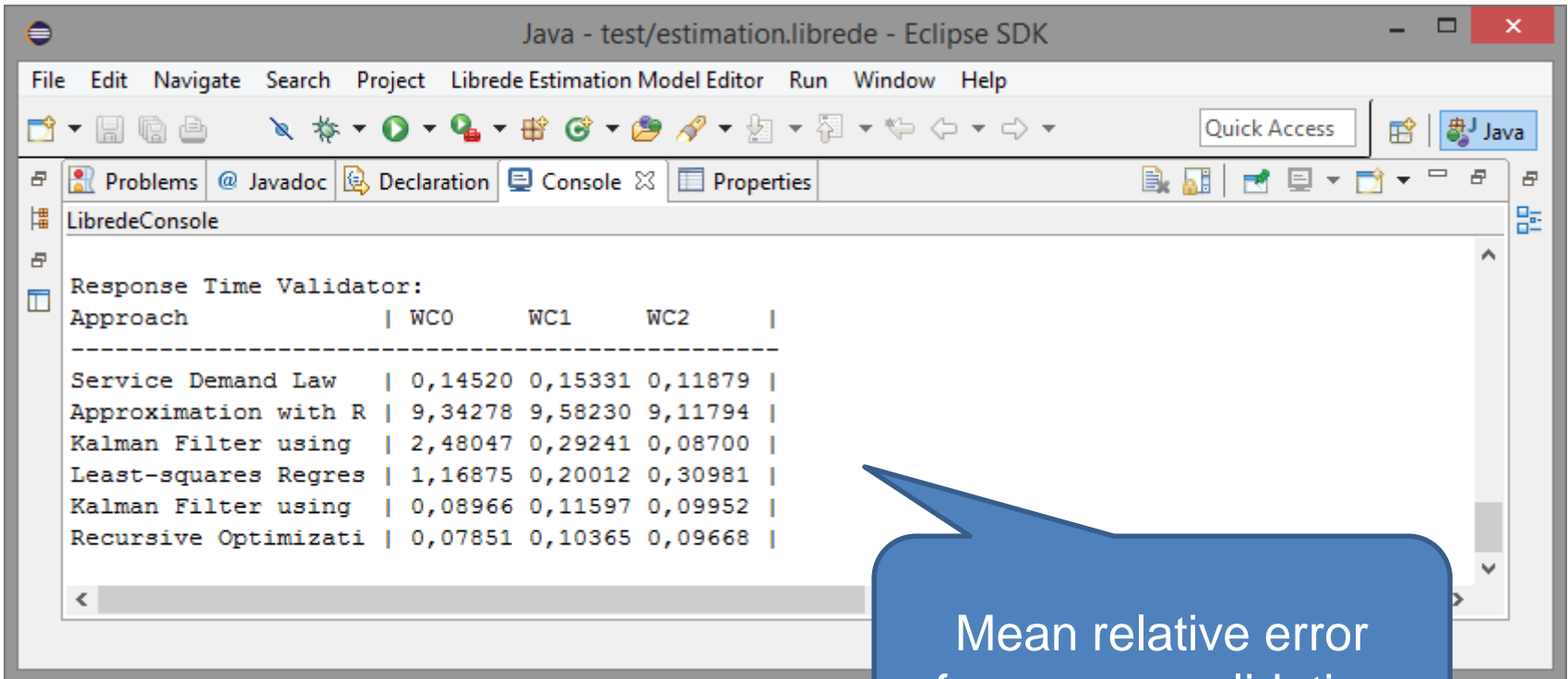
No utilization  
measurements



The screenshot shows the Eclipse IDE interface with the console window open. The console displays the following table of estimated resource demands:

Approach	host1 WC0	host1 WC1	host1 WC2
Service Demand Law	0,02364	0,12683	0,07601
Approximation with R	0,04267	0,22936	0,13730
Kalman Filter using	0,07504	0,08002	0,06993
Least-squares Regres	0,04783	0,12561	0,05294
Kalman Filter using	0,02324	0,12512	0,07444
Recursive Optimizati	0,02253	0,12094	0,07245

Estimated resource demands from different approach



The screenshot shows the Eclipse IDE interface with the console window open. The console displays the output of a 'Response Time Validator' which has generated a table of mean relative errors for various approaches. A blue callout bubble points to the table with the text 'Mean relative error from cross-validation'.

```
Response Time Validator:
Approach          | WC0      WC1      WC2      |
-----
Service Demand Law | 0,14520  0,15331  0,11879  |
Approximation with R | 9,34278  9,58230  9,11794  |
Kalman Filter using | 2,48047  0,29241  0,08700  |
Least-squares Regres | 1,16875  0,20012  0,30981  |
Kalman Filter using | 0,08966  0,11597  0,09952  |
Recursive Optimizati | 0,07851  0,10365  0,09668  |
```

Mean relative error  
from cross-validation



# CASE STUDIES

# Case studies (1/3): SPECjEnterprise2010

- Extraction of PCM models (all domains)
- Monitoring
  - WebLogic Diagnostics Framework (WLDF) → Response times
  - Operating system → Aggregate CPU utilization
- Resource demand estimation
  - Response time approximation
  - Service Demand Law

## References

Fabian Brosig, Nikolaus Huber, and Samuel Kounev. Automated Extraction of Architecture-Level Performance Models of Distributed Component-Based Systems. In *26th IEEE/ACM International Conference On Automated Software Engineering (ASE 2011)*, November 2011. Oread, Lawrence, Kansas.

# Case studies (2/3): Multi-tenant applications

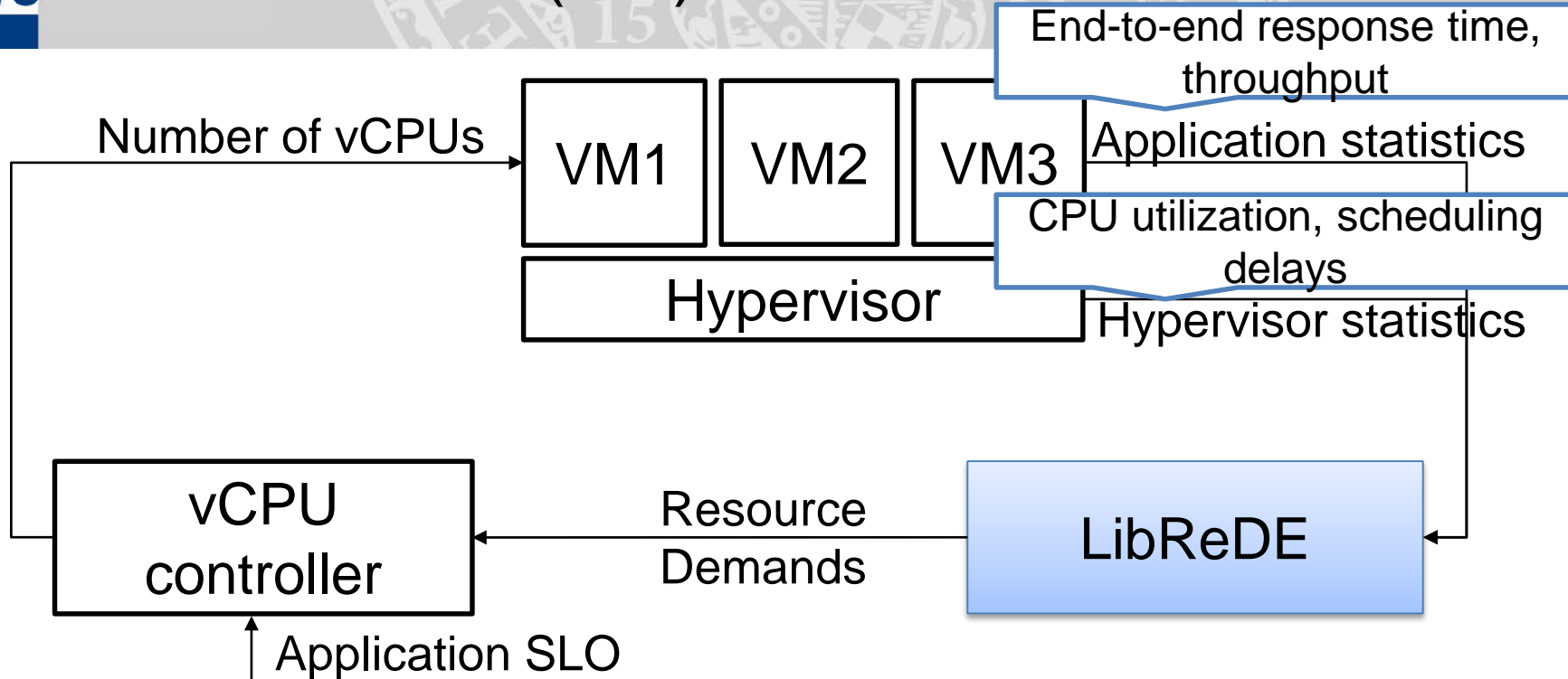
- Admission control of requests based on estimated resource demands
  - Performance isolation
  - QoS differentiation
- Multi-tenant TPC-W in SAP HANA Cloud
- Includes evaluation of resource demand estimators for high number of workload classes

## References

Rouven Krebs, Simon Spinner, Nadia Ahmed, and Samuel Kounev. Resource Usage Control In Multi-Tenant Applications. In *Proceedings of the 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2014)*, Chicago, IL, USA, May 26, 2014. IEEE/ACM. May 2014.



# Case studies (3/3): Zimbra Server



## References

Simon Spinner, Samuel Kounev, Xiaoyun Zhu, Lei Lu, Mustafa Uysal, Anne Holler, and Rean Griffith. Runtime Vertical Scaling of Virtualized Applications via Online Model Estimation. In *Proceedings of the 2014 IEEE 8th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, London, UK, September 8-12, 2014.

# Planned Extensions

- Automatic parameterization of performance models
  - Bridges to DML, QPME, PCM
  - Use performance models for validation
- Additional estimation approaches [7], [15-16]
- Automatic optimization of estimation algorithm parameters

- License: Eclipse Public License (EPL)
- More information at: <http://descartes.tools/librede>
  - Eclipse update site
  - User guide
  - Examples
- Source code available on Bitbucket:
  - <https://bitbucket.org/librede/librede>



- [1] P. Barham, A. Donnelly, R. Isaacs, R. Mortier, Using magpie for request extraction and workload modelling, in: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation – Volume 6, OSDI'04, USENIX Association, Berkeley, CA, USA, 2004, pp. 18.
- [2] M. Kuperberg, M. Krogmann, R. Reussner, ByCounter: Portable Runtime Counting of Bytecode Instructions and Method Invocations, in: Proceedings of the 3rd International Workshop on Bytecode Semantics, Verification, Analysis and Transformation, Budapest, Hungary, 5th April 2008 (ETAPS 2008, 11th European Joint Conferences on Theory and Practice of Software), 2008.
- [3] M. Kuperberg, M. Krogmann, R. Reussner, TimerMeter: Quantifying Accuracy of Software Times for System Analysis, in: Proceedings of the 6<sup>th</sup> International Conference on Quantitative Evaluation of Systems (QEST) 2009, 2009.
- [4] A. Brunnert, C. Vogele, H. Krmar, Automatic performance model generation for java enterprise edition (ee) applications, in: EPEW, 2013, pp. 74-88.
- [5] Y. Bard, M. Shatzoff, Statistical Methods in Computer Performance Analysis, Current Trends in Programming Methodology III.
- [6] J. Rolia, V. Vetland, Parameter estimation for performance models of distributed application systems, in: CASCON '95: Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research, IBM Press, 1995, p. 54.
- [7] S. Kraft, S. Pacheco-Sanchez, G. Casale, S. Dawson, Estimating service resource consumption from response time measurements, in: VALUETOOLS '09: Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, 2009, pp. 1-10.
- [8] G. Pacifici, W. Segmuller, M. Spreitzer, A. Tantawi, CPU demand for web serving: Measurement analysis and dynamic estimation, Performance Evaluation 65 (6-7) (2008) 531-553.
- [9] T. Zheng, C. Woodside, M. Litoiu, Performance Model Estimation and Tracking Using Optimal Filters, Software Engineering, IEEE Transactions on 34 (3) (2008) 391-406.

- [10] D. Kumar, A. Tantawi, L. Zhang, Real-time performance modeling for adaptive software systems, in: VALUETOOLS '09: Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, 2009, pp. 1-10.
- [11] W. Wang, X. Huang, X. Qin, W. Zhang, J. Wei, H. Zhong, Application-Level CPU Consumption Estimation: Towards Performance Isolation of Multi-tenancy Web Applications, in: Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing, 2012, pp. 439 {446.
- [12] Z. Liu, L. Wynter, C. H. Xia, F. Zhang, Parameter inference of queueing models for IT systems using end-to-end measurements, Performance Evaluation 63 (1) (2006) 36-60.
- [13] D. Kumar, L. Zhang, A. Tantawi, Enhanced inferencing: estimation of a workload dependent performance model, in: VALUETOOLS '09: Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, 2009, pp. 1-10.
- [14] D. Menasce, Computing missing service demand parameters for performance models, in: CMG Conference Proceedings, 2008, pp. 241-248.
- [15] J. F. Perez, S. Pacheco-Sanchez, G. Casale, An offline demand estimation method for multi-threaded applications, in: Proceedings of the 2012 IEEE 20th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2013.
- [16] W. Wang, G. Casale, Bayesian service demand estimation using gibbs sampling, in: Proceedings of the 2012 IEEE 20th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2013.
- [17] A. B. Sharma, R. Bhagwan, M. Choudhury, L. Golubchik, R. Govindan, G. M. Voelker, Automatic request categorization in internet services, SIGMETRICS Perform. Eval. Rev. 36 (2008) 16-25.