LibReDE

A Library for Resource Demand Estimation

Simon Spinner, Jürgen Walter
Dept. of Computer Science, University of Würzburg

Symposium on Software Performance,
Nov 27th 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:
- 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
Challenges

Approximation Techniques
- Linear Regression
- Kalman Filter
- Nonlinear Optimization
- Maximum Likelihood Estimation
and many more approaches...

Varying Robustness

Computational Complexity

Implementations not available

Different Preconditions

Approximation Techniques

Linear Regression

Nonlinear Optimization

Maximum Likelihood Estimation

What is the best approach for a given scenario?
Library for Resource Demand Estimation

LibReDE

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

References

LibReDE usage

- Standalone version for offline analysis

  Measurement traces

  ![Image of CSV files]

  Estimated Demands

  ![Image of CSV files]

- Java library for online analysis

  Monitoring tools

  ![Image of cloud]

  Custom application

  ![Image of LibReDE library]

Introduction  Overview  Model Editor  Estimation  Case Studies  Conclusions

Jürgen Walter - A Library for Resource Demand Estimation
Estimation Process

Create estimation model
- EMF-based model
- Graphical eclipse editor

Setup estimation approaches
- Derive estimation problem(s)
- Check pre-conditions

Load monitoring data

Validation Sets
- Run estimation approach(es)
- Cross-Validation
- Output results

Evaluation accuracy
Demo

MODEL EDITOR
Step 1: Workload Description

Services/workload classes

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

Name |
--- |
WC0 |
WC1 |
WC2 |

Resources

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

| Name   | Number of | Scheduling | |
|--------|-----------|------------|
| host1  | 1         | Unknown    | |

Selected Nothing
Abstraction Level

Black-box or System-level: System Entry Points → Services

Coarse-grained: Service Operations → Services

Fine-grained: Internal Actions → Services
Step 2: Data Sources

- Data sources import system monitoring data
- CSV supported out-of-the-box
- Extension point
Step 3: Traces

Input files with monitoring data

Mapping on services/resources

Jürgen Walter - A Library for Resource Demand Estimation
Step 4: Estimation

- 6 estimation approaches
- Extension point

Time interval settings

- Parameters of underlying statistical techniques
Step 5: Validation

- K-fold cross-validation
- Validators based on Operational Laws
- Extension point
Step 6: Output

- Output results to files
- CSV supported out-of-the-box
- Extension point
ESTIMATION
Derives a set of tuples \(<S, O, A>\)

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

Introduction Overview Model Editor Estimation Case Studies Conclusions

Jürgen Walter - A Library for Resource Demand Estimation
### Estimation results

<table>
<thead>
<tr>
<th>Approach</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
<th>host1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Demand Law</td>
<td>0.02364</td>
<td>0.12683</td>
<td>0.07601</td>
<td>0.02364</td>
<td>0.12683</td>
<td>0.07601</td>
<td>0.02364</td>
<td>0.12683</td>
<td>0.07601</td>
<td>0.02364</td>
<td>0.12683</td>
<td>0.07601</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approximation with R</td>
<td>0.04267</td>
<td>0.22936</td>
<td>0.13730</td>
<td>0.04267</td>
<td>0.22936</td>
<td>0.13730</td>
<td>0.04267</td>
<td>0.22936</td>
<td>0.13730</td>
<td>0.04267</td>
<td>0.22936</td>
<td>0.13730</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kalman Filter using</td>
<td>0.07504</td>
<td>0.08002</td>
<td>0.06993</td>
<td>0.07504</td>
<td>0.08002</td>
<td>0.06993</td>
<td>0.07504</td>
<td>0.08002</td>
<td>0.06993</td>
<td>0.07504</td>
<td>0.08002</td>
<td>0.06993</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Least-squares Regres</td>
<td>0.04783</td>
<td>0.12561</td>
<td>0.05294</td>
<td>0.04783</td>
<td>0.12561</td>
<td>0.05294</td>
<td>0.04783</td>
<td>0.12561</td>
<td>0.05294</td>
<td>0.04783</td>
<td>0.12561</td>
<td>0.05294</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kalman Filter using</td>
<td>0.02324</td>
<td>0.12512</td>
<td>0.07444</td>
<td>0.02324</td>
<td>0.12512</td>
<td>0.07444</td>
<td>0.02324</td>
<td>0.12512</td>
<td>0.07444</td>
<td>0.02324</td>
<td>0.12512</td>
<td>0.07444</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recursive Optimization</td>
<td>0.02253</td>
<td>0.12094</td>
<td>0.07245</td>
<td>0.02253</td>
<td>0.12094</td>
<td>0.07245</td>
<td>0.02253</td>
<td>0.12094</td>
<td>0.07245</td>
<td>0.02253</td>
<td>0.12094</td>
<td>0.07245</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Validation results

#### Mean relative error from cross-validation

<table>
<thead>
<tr>
<th>Approach</th>
<th>WC0</th>
<th>WC1</th>
<th>WC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Demand Law</td>
<td>0,14520</td>
<td>0,15331</td>
<td>0,11879</td>
</tr>
<tr>
<td>Approximation with R</td>
<td>9,34278</td>
<td>9,58230</td>
<td>9,11794</td>
</tr>
<tr>
<td>Kalman Filter using</td>
<td>2,48047</td>
<td>0,29241</td>
<td>0,08700</td>
</tr>
<tr>
<td>Least-squares Regres</td>
<td>1,16875</td>
<td>0,20012</td>
<td>0,30981</td>
</tr>
<tr>
<td>Kalman Filter using</td>
<td>0,08966</td>
<td>0,11597</td>
<td>0,09952</td>
</tr>
<tr>
<td>Recursive Optimizati</td>
<td>0,07851</td>
<td>0,10365</td>
<td>0,09668</td>
</tr>
</tbody>
</table>
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

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

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
References (1/2)


