Performance Modeling in Industry
A Case Study on Storage Virtualization

Nikolaus Huber nikolaus.huber@kit.edu
Steffen Becker stbecker@mail.upb.de
Christoph Rathfelder rathfelder@fzi.de
Jochen Schweflinghaus schwefel@de.ibm.com
Ralf Reussner reussner@kit.edu
Storage Virtualization on SystemZ

- Evaluate design alternatives
- Optimize performance through identifying bottlenecks
- Assess applicability of model-based performance prediction

Motivation ► Approach ► Validation ► Evaluation ► Conclusion
Storage Virtualization on SystemZ

- Evaluate design alternatives
- Optimize performance through identifying bottlenecks
- Assess applicability of model-based performance prediction

Motivation ► Approach ► Validation ► Evaluation ► Conclusion
Synchronous Request Handling

Motivation ► Approach ► Validation ► Evaluation ► Conclusion

Client 1
Request Queue 1
Request Queue n

Client 2
Request Queue 1
Request Queue n

Client n
Request Queue 1
Request Queue n

VL

I/O Thread

I/O Interface
Storage Hardware
Asynchronous Request Handling

Client 1  
Request Completion Queue Pair 1  
Request Completion Queue Pair n

Client 2  
Request Completion Queue Pair 1  
Request Completion Queue Pair n

Client n  
Request Completion Queue Pair 1  
Request Completion Queue Pair n

Motivation  ►  Approach  ►  Validation  ►  Evaluation  ►  Conclusion
Asynchronous Request Handling

Performance Questions:
- Would the asynchronous design alternative perform better?
- How many I/O threads are required?
- How many CPUs are sufficient?

Client 1
Request Completion Queue Pair 1
Request Completion Queue Pair n

Client 2
Request Completion Queue Pair 1
Request Completion Queue Pair n

Client n
Request Completion Queue Pair 1
Request Completion Queue Pair n

VL

I/O Interface

Storage Hardware

Motivation ➤ Approach ➤ Validation ➤ Evaluation ➤ Conclusion
Palladio Approach

- Approach for model-based performance prediction
- Simulation & Analysis Tools
- Target domain: Business Information Systems

<table>
<thead>
<tr>
<th>Structural Model</th>
<th>Behavior Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment Model</td>
<td>Usage Model</td>
</tr>
</tbody>
</table>

Motivation ► Approach ► Validation ► Evaluation ► Conclusion
Approach

<table>
<thead>
<tr>
<th>Motivation</th>
<th>Approach</th>
<th>Validation</th>
<th>Evaluation</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

**Motivation**

**Approach**

**Validation**

**Evaluation**

**Conclusion**

---

**Synchronous Model**

**Asynchronous Model**

**Model Parameterization**

**Palladio Toolchain**

**Model Validation**

**Measurements**

**Measurements**

---

**Performance Modeling in Industry: A Case Study On Storage Virtualization**

Nikolaus Huber - 05.05.2010

Software Design and Quality Group – Descartes Research Group

Institute for Program Structures and Data Organization
Model Implementation

Motivation  ►  Approach  ►  Validation  ►  Evaluation  ►  Conclusion
Model Implementation

Motivation ► Approach ► Validation ► Evaluation ► Conclusion
Model Validation (Request Type Mix)

Overall relative prediction error < 22%

Motivation | Approach | Validation | Evaluation | Conclusion
Answering Performance Questions

<table>
<thead>
<tr>
<th>Expected</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected</td>
<td>Actual</td>
</tr>
</tbody>
</table>

**Evaluate design alternatives**

<table>
<thead>
<tr>
<th>Async. performs better</th>
<th>No difference in throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Async compensates peak load</td>
</tr>
</tbody>
</table>

**Optimize performance through identifying bottlenecks**

<table>
<thead>
<tr>
<th>I/O threads</th>
<th>Storage Hardware:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Throughput bottleneck at little load</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue blocking</th>
<th>I/O Interface:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Throughput bottleneck at high load</td>
</tr>
<tr>
<td></td>
<td>CPU increases throughput</td>
</tr>
</tbody>
</table>
Experiences Gained

✓ Assess applicability of model-based performance prediction

- High initial effort
  Tradeoff: accuracy ↔ modeling effort [6PM]
- Ease what-if analysis and design alternatives evaluation
- Valuable to identify performance bottlenecks
- Improve understanding of the system
- Cheaper than performance prototype [24PM]

---

Motivation ➤ Approach ➤ Validation ➤ Evaluation ➤ Conclusion

---

05.05.2010 Nikolaus Huber - Performance Modeling in Industry: A Case Study On Storage Virtualization

Software Design and Quality Group – Descartes Research Group
Institute for Program Structures and Data Organization
Conclusions

Case study results

- Model-based performance prediction valuable in realistic industrial scenarios (6PM ↔ 24PM)

- PCM mature and applicable beyond its target domain (Overall relative prediction error <22%)

- Surprising answers to performance questions
Outlook

- Descartes Research Project
  - Models capturing aspects of dynamic system (e.g. virtualization)
  - Model-based performance prediction at runtime
  - Autonomic and self-aware software systems

http://www.descartes-research.net
Conclusions

- Case study results
  - Model-based performance prediction valuable in realistic industrial scenarios \((6PM \leftrightarrow 24PM)\)
  - PCM mature and applicable beyond its target domain \((Overall\ relative\ prediction\ error \ <22\%)\)
  - Surprising answers to performance questions

http://www.descartes-research.net
Any Questions? Thank you!

http://www.descartes-research.net
http://www.palladio-approach.net
Measurements complex, conducted by IBM

Prediction error <10% after model calibration
Model-based Performance Prediction

Modeling process:
- **Design**
- **Annotated Design**
- **Estimation / Measurement**
- **Simulation**
- **Analysis**
- **Response Time, Utilization, Throughput**
- **Feedback**
- **Model Transformation**
- **Performance Model**
Virtualization and Performance Modeling

I/O Virtualization
- Scale-up [W JW07]
- Scale-out [WRJ07]

Performance Modeling
- PCM & CoCoME [KR08]
- PCM at CAS Software AG [And08]
- Queuing Networks [P GGG06]

No intersection of performance modeling and virtualization!


Assumptions (Request Queues)

- **RequestGenerator**
  - Represents request queues
  - Probability functions for queue locking
Assumptions (Request Queues) II

- Call of `getRequest` delayed

- Blocking probability
  \[ m(t, q) = 1 - \left(1 - \frac{1}{q}\right)^{t-1} \]

- Delay calculated by
  \[ delay(s, t, q) = s \cdot \left[1 + \text{Binom}(1, m(t, q)) \cdot \text{Pois}(m(t, q))\right] \]
  Blocked or not
  Amount of blocked queue accesses
Answering Performance Questions

☑ Evaluate design alternatives
  - No difference in throughput
  - Async. can compensate peak loads
  - No overload situation in sync.

☑ Optimize performance through identifying bottlenecks
  - Not the I/O threads
  - No queue blocking influences
  - Storage Hardware -- Throughput bottleneck at little load
  - I/O Interface -- Throughput bottleneck at maximum load
  - CPU power increases throughput