Software Performance Models for Distributed Systems

Overview
- SPE approach for early assessments of distributed systems
- Approximation approach
- Modeling synchronization and coordination
  - Approximate analytic solution
  - Simulation solution
- Case study

Part 1: Distributed System Performance Approximation

Sequence Diagram

Partition the Distributed System Model
- Identify primary facility
- Construct software/system model
- Estimate delays for other system interactions
- Iterative solutions refine estimates

© 1984-2007 by Performance Engineering Services Div., L&S Computer Technology, Inc. All rights reserved.
Early Life Cycle Models
- Focus on one scenario/processor at a time
- Approximate delay for "blips" in other scenarios

Distributed System Model Solutions
- 1. Use software execution model approximation techniques for estimating the response time of remote calls
- 2. Use iterative solutions - solve for response time of remote calls and substitute for delay estimates
- 3. Use "advanced system execution model" simulation to study the effect of queueing, contention for shared processes, and other delays for inter-process coordination.

1. Server Software Model
- CPU?
- I/Os?
- Delay?

Server QNM
- Delay
- Disk1
- Server CPU
- Disk2

Results
- Best case end-to-end response time, 116 sec., is too high.
Distributed System Performance Models

More Results

Resource utilization:
- CPU: 0.18
- DEVS: 0.62
- Delay: 0.65

Disk utilization 62% is too high for best case.

Case Study Results
- Detected end-to-end response time problem
- Initial configuration sizing problems
- Performance benchmarks provided initial model data

Client Model
- Scenario(s)?
  - In this example, the processing steps are the same
- Software resource requirements?
  - The resource requirements reflect client processing rather than server processing
- Delay time?
  - The server model solution provides an estimate for client delay
- Client QNM?
  - Represent client devices and a delay server for other processing (only)

"Simple" Combined Model
- Multiple clients?
- Multiple servers?
- Multiple workloads?
- Servers connected with WANs?
  - Defer the combined model until later in development.

Part 2: Modeling Synchronization and Coordination

Execution Graph Synchronization Nodes

© 1984-2007 by Performance Engineering Services Div., L&S Computer Technology, Inc. All rights reserved.
Advanced System Model

- Refine scenario composition
- Assignment of scenarios to processors
- Connect processing "blips"

Advanced Model Solution

- Detailed Simulation with CSIM
- Mailboxes for called processes
  - "Messages" may wait in mailbox until called process is free
- Events for responses
  - Event set when response is sent
  - Calling process "waits" for event
- Hybrid solution
  - Phases summarize processing requirements between synchronization points

Advanced Model Results

- Additional results from SPE•ED
  - Response time for called processes (mean, min, max)
  - Number of waiting requests and time in mailbox queue for called processes
  - Throughput of called processes
  - Proportion of elapsed time that depends on other processes
  - Amount of configuration resources used by each scenario
  - Overall device utilization
- Results show problems due to synchronization
  - Excessive delays for called processes
  - Excessive contention for system resources
  - Lock-step execution problems

Evaluate Performance Solutions

- Reduce processing requirements
- Determine number of threads
- Change assignment of objects to processes (scenarios)
- Change assignment of scenarios to processors
- Configuration alternatives

Case Study

- Electronic virtual storefront, wasteBucks.com
- Use cases:
  - Take customer order
  - Fulfill orders
  - Ship orders
  - Order merchandise
- Key object: Customer service component
  - collect completed orders,
  - initiate tasks in other components,
  - track status of orders in progress,
  - etc.
**NewOrder Scenario**

**New Order Execution Graph**

**Process Item Order**

**Best Case Results**

**Process Item Order Results (per item)**

**Performance Improvement**

- Of the 9.8 seconds per item, 7.5 is estimated delay for remote processing.
- Network congestion limits throughput.
- Alternative selected that processes work orders as a group rather than individual items.
- Model changes minor:
  - Number of repetitions → 2 (ready order + back order)
  - Resource requirements for groups rather than items
  - Message time increases → .06 (larger messages)
Distributed System Performance Models

Results - 0.1 jobs/second

Results

- Overhead and delays were significant portion of end-to-end time
- Architectural changes made significant improvement
- Simple models provide sufficient information for architectural evaluation
- Easy to formulate models and evaluate alternatives
- Resolve key performance problems before proceeding

Advanced System Model Solution

1. Simulation solution connects the processing across facilities:
   - If called process is busy, calling process queues
2. Results:
   - mean response time for called processes including time in queue (min & max)
   - Excessive dependent processing?
   - mean, variance, min, max number of requests in queue for called processes
   - Multi-threading?
   - throughput
   - Performance good?

Advanced Model Results

<table>
<thead>
<tr>
<th>Response Time (secs.)</th>
<th>TPut</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>CS: NewOrder</td>
<td>14.4</td>
<td>0</td>
</tr>
<tr>
<td>OE: OrderData</td>
<td>0.16</td>
<td>0</td>
</tr>
<tr>
<td>CS: WorkDetails</td>
<td>0.2</td>
<td>0</td>
</tr>
<tr>
<td>CS: UpdStatus</td>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>WH: WorkAlert</td>
<td>1.3</td>
<td>0</td>
</tr>
<tr>
<td>P: WorkAlert</td>
<td>1.4</td>
<td>0</td>
</tr>
</tbody>
</table>

- Max queue length & queue time suggest more threads for WH: WorkAlert for scalability
- Could show potential “lock-step” problems:
  - WH: WorkAlert response time slightly higher than WH: WorkAlert even though throughput is lower
- The most important results came from early models!

Case Study Conclusions

- Important to assess distributed systems early
- Model progression important
  - Simple approximate models for early life cycle assessment
  - Advanced models for realistic projections and details of interconnection performance
  - Techniques apply to other types of distributed systems such as Web Services and other middleware products

Summary

- SPE approach for early assessments of distributed systems
- Approximation approach
- Modeling synchronization and coordination
  - Approximate analytic solution
  - Simulation solution
- Case study

© 1984-2007 by Performance Engineering Services Div., L&S Computer Technology, Inc. All rights reserved.