Resource Management in the Autonomic Service-Oriented Architecture

Jussara Almeida, Virgilio Almeida
Universidade Federal de Minas Gerais, Brazil

Danilo Ardagna, Chiara Francalanci
Politecnico di Milano, Italy

Marco Trubian,
Università degli Studi di Milano, Italy
Reference Scenario

- In service oriented systems, Quality of Service (QoS) is a service selection driver
- Users evaluate QoS at run time to address their service invocation to the most suitable provider
- QoS requirements are difficult to satisfy because of the high variability of Internet workloads
- Many service centers have started employing autonomic computing self-managing techniques, which dynamically allocate resources among different services on the basis of short-term demand estimates
- Resource Virtualization: service differentiation and performance isolation of multiple Web services sharing the same physical resources
Multi-scale Resource Management Approach

- **SLA Management**: short-term resource allocation problem, i.e., how to allocate resources to different service invocations in order to maximize the revenues from SLA, while minimizing resource management costs.

- **Capacity planning**: a long-term problem, i.e., how to size the service center in order to maximize the long-term net revenue from SLA contracts, while minimizing the total cost of ownership (TCO) of resources.
Outline

- Autonomic Computing Environment
- SLA Management and Long Term Capacity Planning
- Short Term Resource Allocation
- Problem formulation
- Optimization technique
- Experimental results
- Conclusions and Future Work
Autonomic Computing Environment

- Multiple transactional Web services sharing the same service center
- Hosted services are modeled as independent WS classes
- Virtualization: physical resources are partitioned into isolated VMs, each running at a fraction of the total system capacity and dedicated to serve a single WS class

VMs employ an admission control schema that may reject requests
**SLA Management and Long-term Capacity Planning**

Dynamic Resource Allocation

- **Short term Workload Predictor**
  - Δ₁
  - WS throughput
  - Fraction of servers capacity allocated to WS

- **Optimization Model 1**
  - Performance Model

Analysis Module

- **Long term Workload Predictor**
  - Δ₂
  - %loss
  - %violations

- **Optimization Model 1**
  - Performance Model
  - Revenue
  - Time for capacity planning

Long Term Capacity Planning

- **Performance Model**
- **Optimization Model 2**
  - Set of new servers
  - Servers upgrade

---

ICAC 2006 - Dublin - 14th June 2006
System Performance Model

- Goal: estimate the probability that a service invocation response time violates the SLA contract of the corresponding WS class
- Each VM is modeled as a M/M/1 queue
- \( D_i^{(N)} = (D_i^p \cdot OH)/(f_i \cdot P) = D_i^{(1)}/(f_i \cdot P) = D_i/f_i \)
- \( E[R_i] = \frac{D_i}{f_i - D_iX_i} \)
- Probability of violation evaluated by the Markov inequality: \( P(R_i \geq R_i^{SLA}) \approx \min \left( \frac{E[R_i]}{R_i^{SLA}}, 1 \right) \)
- Throughput upper-bound: \( X_i \leq \min \left( \lambda_i, \frac{v_i f_i}{D_i} \right) \)
Short-term Dynamic Resource Allocation

- If a service invocation response time is above a given threshold $R_i^{SLA}$, then, the SLA is violated and the customer will not pay for the Web service. Vice versa, if the response time is lower than $R_i^{SLA}$, the customer will pay $\omega_i$ to the SP.
- $C$: cost per time unit associated to the use of total system resources.
- $T$: control time horizon.
- $\text{Min } \sum_{i=1}^{N} T \cdot \left\{ \omega_i \left[ (\lambda_i - X_i) + P(R_i \geq R_i^{SLA}) X_i \right] + C f_i \right\}$
Optimization Problem

P1) \( \min \sum_{i=1}^{N} \left\{ \omega_i \left( \min \left( \frac{D_i}{R_i} f_i - \frac{1}{D_i X_i}, 1 \right) - 1 \right) X_i + Cf_i \right\} \)

\[ \frac{D_i}{f_i} X_i \leq v_i < 1 \quad \forall i \]  
\[ X_i \leq \lambda_i \quad \forall i \]  
\[ \sum_{i=1}^{N} f_i \leq 1 \]  
\[ X_i, f_i \geq 0 \quad \forall i \]
Optimization Problem

\[ \text{P2}) \quad \min \sum_{i=1}^{N} \left\{ \omega_i \left( \frac{D_i}{R_i^{SLA}} \frac{1}{f_i - D_i X_i} - 1 \right) X_i + C f_i \right\} \]

\[ X_i \leq v_i \frac{f_i}{D_i} < \frac{f_i}{D_i} \quad \forall i \quad (1) \]

\[ X_i \leq \lambda_i \quad \forall i \quad (2) \]

\[ \sum_{i=1}^{N} f_i \leq 1 \quad (3) \]

\[ X_i > 0 \Rightarrow f_i - D_i X_i > \frac{D_i}{R_i^{SLA}} \quad \forall i \quad (4) \]

\[ X_i, f_i \geq 0 \quad \forall i \]

P2 has nonlinear objective function and linear constraints linked by logical conditions. The joint capacity allocation and admission control problem is difficult since the objective function is neither concave nor convex.
Optimization Technique

- P2 is a general nonlinear optimization problem
- Commercial nonlinear optimization tools can solve only small size instances
- A multi-start approach which embeds a Fixed Point Iteration (FPI) technique has been developed
- The FPI iteratively identifies the optimum value of a set of variables \((X_i \text{ or } f_i)\), while the value of the other one (alternatively \(f_i \text{ or } X_i\)) is held fixed
- The FPI stops when the difference between two consecutive objective function values is lower than a fixed threshold
- Within the multi-start framework, each run of the FPI is obtained by randomly generating the initial values of the \(f_i\) variables such that \(\Sigma f_i = 1\)
- The FPI will always converge
Admission Control Sub-problem

\[ P3) \quad \min \sum_{i=1}^{N} \left\{ \omega_i \left( \frac{D_i}{R_i^{SLA}} \frac{1}{f_i - D_i X_i} - 1 \right) X_i \right\} \]

\[ 0 \leq X_i \leq U_i = \min \left( \frac{\overline{f_i}}{D_i}, \lambda_i, \frac{\overline{f_i}}{D_i} - \frac{1}{R_i^{SLA}} \right) \quad \forall i \]

P3 is separable and N admission control sub-problems can be solved independently. Furthermore, the objective function of problem P3 is convex.
Admission Control Sub-problem

\[ P'_i \] \quad \min g_i = \omega_i \left( \frac{D_i}{R_i^{SLA}} \frac{1}{f_i - D_i X_i} - 1 \right) X_i \\
0 \leq X_i \leq U_i \quad \forall i

**Theorem 1.** In the optimum solution of problem \( P'_i \), the throughput for WS invocation \( i \) is either given by:

\[ X_i = \frac{1}{D_i} \left( \bar{f}_i - \sqrt{\frac{D_i \bar{f}_i}{R_i^{SLA}}} \right) \]

or is one of the edges of the interval \([0, U_i]\)
## Capacity Allocation Sub-problem

\[ \text{P4}) \quad \min g = \sum_{i=1}^{N} \left\{ \omega_i \left( \frac{D_i}{R_{i}^{SLA}} \frac{\overline{X}_i}{f_i - D_i \overline{X}_i} \right) + C f_i \right\} \]

\[ f_i > \frac{D_i}{R_{i}^{SLA}} + D_i \overline{X}_i \quad \forall i, \overline{X}_i > 0 \]

\[ f_i \geq \frac{D_i \overline{X}_i}{\nu_i} \quad \forall i \]

\[ \sum_{i=1}^{N} f_i \leq 1 \]

- A feasible solution for problem P4 exists and is given by setting \( f_i = f_i^{(1)} = \max \left( \frac{D_i}{R_{i}^{SLA}} + D_i \overline{X}_i, \frac{D_i \overline{X}_i}{\nu_i} \right) \), for each \( i \), such that \( \overline{X}_i > 0 \), and \( f_i = 0 \), otherwise.

- Stationary point: \( f_i^{(2)} = D_i \overline{X}_i + \sqrt{\frac{\omega_i D_i \overline{X}_i}{R_{i}^{SLA} C}} \)
Theorem 2. In the optimum solution of problem P4 the capacity for WS invocation \( i \) is given by: \( f_i = f_i^{(1)} \), for each \( i \in \overline{N} \). For each \( i \notin \overline{N} \) \( f_i \) is either \( f_i^{(2)} \) otherwise it belongs to the plane \( \Sigma f_i = 1 \) and can be determined by:

\[
f_i = \sqrt{\frac{\omega_i D_i R_{i}^{SLA} X_i}{\omega_u D_u R_{i}^{SLA} X_u}} (f_u - D_u X_u) + D_i X_i
\]

where \( f_u \), with \( u \notin \overline{N} \), is given by:

\[
f_u = \frac{F + D_u X_u \sum_{i \notin \overline{N}} \sqrt{\frac{\omega_i D_i R_{i}^{SLA} X_i}{\omega_u D_u R_{i}^{SLA} X_u}} - \sum_{i \notin \overline{N}} D_i X_i}{\sum_{i \notin \overline{N}} \sqrt{\frac{\omega_i D_i R_{i}^{SLA} X_i}{\omega_u D_u R_{i}^{SLA} X_u}}}
\]
Sketch of the Proof
Sketch of the Proof
Experimental Results

- Tests considered a large set of randomly generated problem instances.
- Both the total system capacity $P$ and the number of Web service classes $N$ are independently varied between 100 and 400.
- WS parameters (demanding time and virtualization overhead) were randomly generated according to the value considered in other literature approaches.
- The goal is to evaluate:
  - cost reduction which can be obtained by taking into account resource usage cost explicitly.
  - algorithm execution time.
FPI Iteration Execution Trace
## Algorithm Performance

### Execution time (sec)

<table>
<thead>
<tr>
<th>( P )</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2.14</td>
<td>5.13</td>
<td>8.28</td>
<td>11.57</td>
</tr>
<tr>
<td>200</td>
<td>3.16</td>
<td>5.98</td>
<td>9.19</td>
<td>12.71</td>
</tr>
<tr>
<td>300</td>
<td>3.14</td>
<td>6.14</td>
<td>8.57</td>
<td>12.43</td>
</tr>
<tr>
<td>400</td>
<td>3.04</td>
<td>6.28</td>
<td>8.71</td>
<td>11.98</td>
</tr>
</tbody>
</table>

### Service Center Utilization & Percentage Savings

<table>
<thead>
<tr>
<th>Service Center Utilization</th>
<th>Percentage Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>38.68%</td>
</tr>
<tr>
<td>0.3</td>
<td>32.98%</td>
</tr>
<tr>
<td>0.4</td>
<td>28.68%</td>
</tr>
<tr>
<td>0.5</td>
<td>24.25%</td>
</tr>
</tbody>
</table>
Conclusions and Future Work

- We considered the problem of resource management in autonomic service-oriented architectures, where multiple Web services share the same infrastructure. Two key problems interrelated problems have been identified.

- The short-term resource management problem has been analyzed in depth.

- The novelty of our proposed model lies in:
  - the minimization of resource usage costs
  - providing a solutions for the resource allocation and the admission control problems jointly.
Conclusions and Future Work

- We have proposed an ad hoc solver for the optimization model that is very efficient, solving reasonably large problem sizes (up to 400 WS classes) typically under 15 seconds
- Further experimentation with our proposed resource management schema, comparing it against alternative strategies
- Designing and implementation of the analysis and long-term capacity modules
Questions?