A QoS Aware Fault Tolerant Middleware for Dependable Service Composition

Zibin Zheng and Michael R. Lyu

Department of Computer Science & Engineering
The Chinese University of Hong Kong
Hong Kong, China

DSN 2009, Lisbon, Portugal, June 29-July 2, 2009
Outlines

1. Introduction
2. Preliminaries
3. Optimal Fault Tolerance Strategy Selection
4. Experiments
5. Conclusion and Future Work
1. Introduction
1.1 Web Services

- Self-description
- Loosely-coupled
- Highly-dynamic
- Cross-domain
- Compositional nature
1.2 A Motivating Example

- \( SP=(T,P,B) \)
  - SP: service plan
  - T: a set of tasks
  - P: settings
  - B: Structure information

- **Challenges:**
  - Optimal FT strategy selection
  - Local and global constraints
  - Stateful Web services

- **Local constraint:**
  Response time of \( t_1 \) < 1000 ms.

- **Global constraint:**
  Success-rate of the whole service plan > 99%.

How to employ these alternative candidates for reliability enhancement?
1.3 Fault Tolerant Web Services

- Web services are becoming popular for building distributed Internet systems.
- It is difficult to build reliable service-oriented systems.
  - Reliability of the system is highly dependent on the remote Web service components.
  - Web services are usually hosted by other organizations.
    - may contain faults.
    - may become unavailable suddenly.
    - Source codes of the Web services are usually unavailable.
  - The Internet environment is unpredictable.
1.3 Fault Tolerant Web Services

- Traditional software reliability engineering
  - Software fault tolerance by design diversity is a major approach for building highly reliable systems.
  - It is expensive to develop redundant components.

- Service reliability engineering
  - Web services with identical/similar functionality are abundant in the Internet.
  - Cost becomes less of the concern.

How to employ these redundant Web services for building fault tolerant services reliably and effectively?
1.4 Contributions

- **A systematic framework of fault tolerant Web services:**
  - User-collaborative QoS model of Web services.
  - Various commonly-used fault tolerance strategies for Web services.
  - Web service QoS composition model.

- **Optimal fault tolerance strategy selection algorithms for stateless and stateful Web services.**

- **Large-scale real-world experiments.**
2. Preliminaries
2.1 System Architecture

YouTube: sharing videos.
Wikipedia: sharing knowledge.
WS-DREAM: sharing QoS data of Web services.
http://www.wsdream.net
2.2 QoS Model of WS

1. **Availability (av)** \(q^1\): the percentage of time that a Web service is operating during a certain time interval.

2. **Price (pr)** \(q^2\): the fee that a service user has to pay for invoking a Web service.

3. **Popularity (po)** \(q^3\): the number of received invocations of a Web service during a certain time interval.

4. **Data-size (ds)** \(q^4\): the size of the Web service invocation response.

5. **Success-rate (sr)** \(q^5\): the probability that a request is correctly responded within the maximum expected time. \(q^5 = \frac{\text{succInvocationNum}}{\text{totalInvocationNum}}\).

6. **Response-time (rt)** \(q^6\): the time duration between service user sending a request and receiving a response.

7. **Overall Success-rate (osr)** \(q^7\): the average value of the invocation success rate \(q^5\) of all service users.

8. **Overall Response-time (ort)** \(q^8\): the average value of the response-time \(q^6\) of all service users.

\[ q = (q^1, ..., q^8). \]
2.3 Web Service Composition

<table>
<thead>
<tr>
<th>QoS Properties</th>
<th>Basic Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>rt, ort (x=6, 8)</td>
<td>[ \sum_{i=1}^{n} q_i^x \quad \max_{i=1}^{k} q_i^x \quad \sum_{i=1}^{n} p_i q_i^x \quad \sum_{i=0}^{n} p_i q_i^x i ]</td>
</tr>
<tr>
<td>av, sr, osr (x=1, 5, 7)</td>
<td>[ \prod_{i=1}^{n} q_i^x \quad \sum_{i=1}^{n} S^x(i) \quad \sum_{i=1}^{n} p_i q_i^x \quad \sum_{i=0}^{n} p_i(q_i^x)^i ]</td>
</tr>
<tr>
<td>pr, po, ds (x=2, 3, 4)</td>
<td>[ \sum_{i=1}^{n} q_i^x \quad \sum_{i=1}^{n} q_i^x \quad \sum_{i=1}^{n} p_i q_i^x \quad \sum_{i=0}^{n} p_i q_i^x i ]</td>
</tr>
</tbody>
</table>

Figure 2. Basic Compositional Structures

Figure 3. Service Plan Decomposition
2.4 Fault Tolerance Strategies

• **Basic fault tolerance strategies:**
  - **Retry:** The original Web service will be tried for a certain number of times if it fails.
  - **Recovery Block (RB):** Another standby Web service will be tried sequentially if the primary Web service fails.
  - **N-Version Programming (NVP):** all the $n$ candidates are invoked in parallel and the final result will be determined by majority voting.
  - **Active.** All the $n$ candidates are invoked in parallel and the first returned response will be selected as the final result.

• **Combination of the basic fault tolerance strategies**
  - More complex strategies by combining the basic strategies.
3. Optimal Fault Tolerance Strategy Selection
3.1 Utility Function

• Positive QoS properties (larger for better):
  – Availability, popularity, success-rate, overall success-rate.

• Negative QoS properties (smaller for better):
  – Response time, price, data-size, overall response-time.

• Transfer Negative QoS values to positive QoS.
  \[ q_{i,j}^k = \max q_i^k - q_{i,j}^k, \]

• Normalization of the QoS values.
  \[ \tilde{q}_{i,j}^k = \begin{cases} 
    \frac{q_{i,j}^k - \min q_i^k}{\max q_i^k - \min q_i^k} & \text{if } \max q_i^k \neq \min q_i^k \\
    1 & \text{if } \max q_i^k = \min q_i^k 
  \end{cases} \]

• Utility function:
  \[ u_{i,j} = utility(q_{i,j}) = \sum_{k=1}^{c} w_k \times \tilde{q}_{i,j}^k, \]
### 3.2 Notations

#### Table 3. Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SP$</td>
<td>a service plan, which is a triple $(T, P, B)$.</td>
</tr>
<tr>
<td>$T$</td>
<td>a set of tasks in the service plan, $T = SLT \cup SFT$.</td>
</tr>
<tr>
<td>$SLT$</td>
<td>a set of stateless tasks, $SLT = {t_i}$ for $i = 1$ to $n_l$.</td>
</tr>
<tr>
<td>$SFT$</td>
<td>a set of stateful tasks, $SFT = {SFT_i}$ for $i = n_l$ to $n$.</td>
</tr>
<tr>
<td>$SFT_i$</td>
<td>a set of related tasks of the $i^{th}$ stateful task.</td>
</tr>
<tr>
<td>$n$</td>
<td>the number of tasks in SP, $n = n_l + n_f$.</td>
</tr>
<tr>
<td>$n_l$</td>
<td>the number of the stateless tasks in SP, $n_l =</td>
</tr>
<tr>
<td>$n_f$</td>
<td>the number of the stateful tasks in SP, $n_f =</td>
</tr>
<tr>
<td>$n_i$</td>
<td>number of state related tasks of $SFT_i$, $n_i =</td>
</tr>
<tr>
<td>$S_i$</td>
<td>a set of candidates for $t_i$, $S_i = {s_{ij}}$ for $j = 1$ to $m_i$.</td>
</tr>
<tr>
<td>$m_i$</td>
<td>the number of candidates for $t_i$, $m_i =</td>
</tr>
<tr>
<td>$p_i$</td>
<td>the optimal candidate index for $t_i$.</td>
</tr>
<tr>
<td>$LC_i$</td>
<td>local constraints for task $t_i$, $LC_i = {l_{ck}}$ for $k = 1$ to $c$.</td>
</tr>
<tr>
<td>$GC$</td>
<td>global constraints for $SP$, $GC = {g_{ck}}$ for $k = 1$.</td>
</tr>
<tr>
<td>$c$</td>
<td>the number of quality properties.</td>
</tr>
<tr>
<td>$q_{ij}$</td>
<td>a quality vector for $s_{ij}$, $q_{ij} = (q_{ij}^k)$ for $k = 1$ to $c$.</td>
</tr>
<tr>
<td>$ER$</td>
<td>a set of execution routes of $SP$, $ER = {ER_i}$ for $i = 1$ to $n_e$.</td>
</tr>
<tr>
<td>$n_e$</td>
<td>the number of execution routes of a service plan.</td>
</tr>
<tr>
<td>$pro(ER_i)$</td>
<td>the execution probability of $ER_i$.</td>
</tr>
<tr>
<td>$SR$</td>
<td>a set of sequential routes of $SP$, $SR = {SR_i}$ for $i = 1$ to $n_s$.</td>
</tr>
<tr>
<td>$n_s$</td>
<td>the number of sequential routes of $SP$.</td>
</tr>
<tr>
<td>$pct$</td>
<td>a user defined threshold for $ER$.</td>
</tr>
</tbody>
</table>
3.3 Optimal Selection With Local Constraints

- Selection problem

Problem 1

Minimize: \[ \sum_{j=1}^{m_i} u_{ij} x_{ij} \]

Subject to:

- \[ \sum_{j=1}^{m_i} q_{ij}^k x_{ij} \leq l c_i^k (k = 1, 2, ..., c) \]
- \[ \sum_{j=1}^{m_i} x_{ij} = 1 \]
- \( x_{ij} \in \{0, 1\} \)

Data: Service plan SP, local constraints LC, candidates S
Result: Optimal candidate index \( \rho \) for SP.

\begin{align*}
1 & n_t=|SLT|; n_f=|SFT|; n=n_t+n_f; n_i=|SFT_i|; m_i=|S_i|; \\
2 & \text{for } (i = 1; i \leq n_t; i++) \text{ do} \\
3 & \quad \text{for } (j = 1; j \leq m_i; j++) \text{ do} \\
4 & \quad \quad \text{if } \forall x(q_{ij}^k \leq l c_i^k) \text{ then } u_{ij} = \text{utility}(q_{ij}); \\
5 & \quad \text{end} \\
6 & \quad \text{if } \text{no candidate meet } l c_i \text{ then } \text{Throw exception; } \\
7 & \quad u_{ix} = \min\{u_{ij}\}; \\
8 & \quad \rho_i = x; \\
9 & \text{end} \\
10 & \text{for } (i=n_t+1; i \leq n; i++) \text{ do} \\
11 & \quad \text{for } (j=1; j \leq m_i; j++) \text{ do} \\
12 & \quad \quad \text{if } \forall x \forall y(q_{ijy}^k \leq l c_i^k) \text{ then} \\
13 & \quad \quad \quad q = \text{flowQoS}(SP, q_{i1j}, ..., q_{in_ij}); \\
14 & \quad \quad \quad u_{ij} = \text{utility}(q); \\
15 & \quad \quad \text{end} \\
16 & \quad \text{end} \\
17 & \quad \text{if } \text{no candidate meet } l c_i \text{ then } \text{Throw exception; } \\
18 & \quad u_{ix} = \min\{u_{ij}\}; \\
19 & \quad \text{for all tasks in } SFT_i \text{ do } \rho_{ik} = x; \\
20 & \text{end}
\end{align*}

Algorithm 2: FT Selection with Local Constraints
3.4 Selection With Global Constraints

- 0-1 Integer Programming Problem

**Problem 2:** Minimize:

$$
\sum_{i \in ER_i} \sum_{j \in S_i} u_{ij} x_{ij}
$$

Subject to:

$$
\sum_{i \in ER_i} \sum_{j \in S_i} q_{ij}^y x_{ij} \leq g e^y (y = 2, 3, 4)
$$

$$
\forall k, \sum_{i \in S_{R_k}} \sum_{j \in S_i} q_{ij}^y x_{ij} \leq g e^y (y = 6, 8)
$$

$$
\prod_{i \in ER_i} \prod_{j \in S_i} (q_{ij}^y)^{x_{ij}} \leq g e^y (y = 1, 5, 7)
$$

$$
\forall SFT_i, x_{v_{y_1} j} = x_{v_{y_2} j} = \ldots = x_{u_{y_1} j} (t_{y_1} \in SFT_i)
$$

$$
\forall i, \sum_{j \in S_i} x_{ij} = 1; x_{ij} \in \{0, 1\}
$$

$$
\sum_{i \in ER_i} \sum_{j \in S_i} x_{ij} \ln (\frac{q_{ij}^y}{\tilde{q}_{ER_i}^y}) \leq \ln (ge^y) (y = 1, 5, 7),
$$

$$
\tilde{q}_{ER_i}^y = \frac{\min \ln (q^y)}{\max \ln (q^y) - \min \ln (q^y)},
$$

$$
\tilde{q}_{ER_i}^y = \ln (\tilde{q}_{ER_i}^y) = \sum_{i \in ER_i} \sum_{j \in S_i} x_{ij} \ln (q_{ij}^y).
$$
3.5 Hybrid Algorithm

As the IP problem is NP-complete, we propose a more effective hybrid algorithm.

\begin{algorithm}
\DontPrintSemicolon
\KwData{SP, ER, Constraints GC, LC, Candidates S, pct}
\KwResult{Optimal candidates index $\rho$ for SP.}
\n\begin{algorithmic}[1]
\State $n=n_{l}+n_{r}$; $n_{l}=|SLT|$; $m_{i}=|s_{i}|$; $n_{e}=|ER|$; $T_{e} = \{ \}$
\For{$(i = 1; i \leq n_{e}; i++)$}
\If{$ER_{i} \in$ the first pct major routs}
\State $FTBAB(ER_{i})$
\State $T_{e} = T_{e} \cup T_{i}$
\EndIf
\EndFor
\If{$T_{e} == T$}
\Return $\rho$;
\EndIf
\State $\rho = findInitialSolution(T, GC, LC, S, T_{e}, \rho)$;
\State $q_{att} = flowQoS(SP, q_{1 \rho_{1}}, \ldots, q_{n \rho_{n}})$;
\While{$\exists x (\frac{q_{att}^{x}}{q_{c}_{x}} > 1)$}
\State $S' = findExchangeCandidate(T, GC, LC, \rho)$;
\If{$|S'| == 0$}
\Return No Feasible Solution Exist!
\Else
\ForAll{$s_{xy} \in S'$}
\State $\rho_{x} = y$
\EndFor
\EndIf
\EndWhile
\State $\rho = feasibleUpgrade(SP, GC, LC, S, \rho)$;
\Until{$\rho$ do not change}
\Return $\rho$;
\end{algorithmic}
\end{algorithm}

Algorithm 3: Hybrid Algorithm: FT-BABHEU
4. Experiments
4.1 Experimental Setup

• Obtain 21,197 publicly available Web services from the Internet.
• Generate client stub classes for 18,102 Web services. A total of 343,917 Java classes are generated.
• Randomly select 100 Web services for conducting experiment.
• 150 distributed computer nodes from PlanetLab.
• More than 1.5 millions Web service invocations
4.2 Location Information

- PlanetLab ([http://www.planet-lab.org](http://www.planet-lab.org)) is a global research network, which consists of 1016 distributed computers.

PlanetLab currently consists of 1016 nodes at 473 sites.
4.3 QoS of Web Services

Further information and the detailed Web service QoS dataset is available in http://www.wsdream.net
4.4 Case Studies

Table 5.2. QoS Values of the Stateless Task ($t_1$)

<table>
<thead>
<tr>
<th>WS</th>
<th>Q</th>
<th>CN</th>
<th>AU</th>
<th>US</th>
<th>SG</th>
<th>TW</th>
<th>HK</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>aus rt</td>
<td>3659</td>
<td>1218</td>
<td>121</td>
<td>544</td>
<td>934</td>
<td>491</td>
<td>681</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.819</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.977</td>
<td>0.989</td>
<td></td>
</tr>
<tr>
<td>ajp rt</td>
<td>3310</td>
<td>1052</td>
<td>338</td>
<td>472</td>
<td>824</td>
<td>469</td>
<td>686</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.788</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.980</td>
<td>0.987</td>
<td></td>
</tr>
<tr>
<td>ade rt</td>
<td>3233</td>
<td>1076</td>
<td>303</td>
<td>596</td>
<td>1178</td>
<td>612</td>
<td>846</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.813</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.973</td>
<td>0.987</td>
<td></td>
</tr>
<tr>
<td>aca rt</td>
<td>3530</td>
<td>1190</td>
<td>130</td>
<td>456</td>
<td>916</td>
<td>599</td>
<td>714</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.807</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.983</td>
<td>0.988</td>
<td></td>
</tr>
<tr>
<td>afr rt</td>
<td>3289</td>
<td>1309</td>
<td>306</td>
<td>600</td>
<td>1193</td>
<td>630</td>
<td>864</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.844</td>
<td>0.998</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.974</td>
<td>0.989</td>
<td></td>
</tr>
<tr>
<td>auk rt</td>
<td>3550</td>
<td>1326</td>
<td>305</td>
<td>671</td>
<td>1178</td>
<td>633</td>
<td>862</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.837</td>
<td>0.997</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.971</td>
<td>0.988</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3. QoS Values of the Stateful Task ($t_2$-$t_6$)

<table>
<thead>
<tr>
<th>WS</th>
<th>Q</th>
<th>CN</th>
<th>AU</th>
<th>US</th>
<th>SG</th>
<th>TW</th>
<th>HK</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>aus rt</td>
<td>16434</td>
<td>5625</td>
<td>717</td>
<td>2708</td>
<td>4166</td>
<td>2328</td>
<td>3297</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.450</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.972</td>
<td>0.940</td>
<td></td>
</tr>
<tr>
<td>ajp rt</td>
<td>14763</td>
<td>4980</td>
<td>1751</td>
<td>2505</td>
<td>3730</td>
<td>2058</td>
<td>3335</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.450</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.998</td>
<td>0.973</td>
<td>0.944</td>
</tr>
<tr>
<td>ade rt</td>
<td>14640</td>
<td>6718</td>
<td>1646</td>
<td>3038</td>
<td>5209</td>
<td>2730</td>
<td>3985</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.438</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.972</td>
<td>0.935</td>
<td></td>
</tr>
<tr>
<td>aca rt</td>
<td>15602</td>
<td>5527</td>
<td>1403</td>
<td>2488</td>
<td>4150</td>
<td>2305</td>
<td>3427</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.452</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.996</td>
<td>0.979</td>
<td>0.944</td>
</tr>
<tr>
<td>afr rt</td>
<td>14560</td>
<td>5983</td>
<td>2211</td>
<td>3069</td>
<td>5175</td>
<td>2862</td>
<td>4045</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.496</td>
<td>0.992</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.969</td>
<td>0.937</td>
<td></td>
</tr>
<tr>
<td>auk rt</td>
<td>15898</td>
<td>6066</td>
<td>1630</td>
<td>3044</td>
<td>5209</td>
<td>2819</td>
<td>4048</td>
<td></td>
</tr>
<tr>
<td>sr</td>
<td>0.484</td>
<td>0.988</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.998</td>
<td>0.970</td>
<td>0.939</td>
</tr>
</tbody>
</table>

(a) Response-time of $t_1$
(b) Success-rate of $t_1$
(c) Response-time of $t_2$-$t_6$
(d) Success-rate of $t_2$-$t_6$
4.5 Performance Study (1)

Figure 6. Performance of Computation Time
4.5 Performance Study (2)

Figure 7. Performance of Selection Results
5. Conclusion and Future Work
5.1 Conclusion and Future Work

- **Conclusion**
  - Fault tolerance strategies
  - A QoS model for Web services
  - A QoS composition model for Web services
  - Optimal fault tolerance strategy selection algorithms
  - Large-scale real-world experiments

- **Future work**
  - Investigation on more QoS properties
  - Experiments with more service users on more real-world Web services