A Distributed Replication Strategy Evaluation and Selection Framework for Fault Tolerant Web Services

Zibin Zheng and Michael R. Lyu

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

ICWS 2008, Beijing, China, 24 September, 2008
Outlines

1. Introduction
2. Distributed Evaluation Framework
3. Fault Tolerance Replication Strategies
4. An Optimal Strategy Selection Algorithm
5. Experiments
6. Conclusion
1. Introduction

- Web services are becoming popular.
- Reliability of the service-oriented applications becomes difficult to be guaranteed.
  - Remote Web services may contain faults.
  - Remote Web services may become unavailable.
  - The Internet environment is unpredictable.
1. Introduction

• Traditional software reliability engineering
  – Fault Tolerance is a major approach for building highly reliable system.
  – Expensive.

• Service reliability engineering
  – Web service with identical/similar interface
  – Less expensive & less time-consuming

• A lot of fault tolerance replication strategies
  – Time redundancy
  – Space redundancy

Which fault tolerance replication strategy is optimal?
1. Introduction

  – User-collaborated evaluation
    • YouTube: sharing videos.
    • Wikipedia: sharing knowledge.
    • Sharing evaluation results of target Web services.
  – Evaluation of individual Web service
  – Evaluation of fault tolerance strategies
  – Optimal fault tolerance strategy selection.
2. Distributed Evaluation Framework

- **Service users**
  - Web service selection
  - Overall performance of target Web services
    - Different locations
    - Long time duration
- **Service providers**
  - Providing better services
  - Overall performance of their own Web services
- **Overall performance of Web services is not easy to be obtained**
  - Time-consuming
  - Expensive
2. Distributed Evaluation Framework

1. Evaluation request
2. Load Applet
3. Create test cases
4. Schedule test tasks
5. Assign test cases
6. Client run test cases
7. Send back results
8. Analyze and return final results to client.

- Evaluation results from different locations
- Don’t need good knowledge on FT strategies, test case generation, and so on.
- Don’t need to implement evaluation mechanism.
3. Replication Strategies

• Basic replication strategies.
  1. **Active.** The application sends requests to different replicas at the same time and uses the first properly returned response as final result.
  2. **Time.** The same Web Service will be tried one more time if it fails at first.
  3. **Passive.** Another standby Web Service will be tried in sequence if the primary Web Service fails.

<table>
<thead>
<tr>
<th></th>
<th>Active</th>
<th>Time</th>
<th>Passive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>1. Active</td>
<td>4. Active+Time</td>
<td>6. Active+Passive</td>
</tr>
<tr>
<td>Time</td>
<td>5. Time+Active</td>
<td>2. Time</td>
<td>8. Time+Passive</td>
</tr>
</tbody>
</table>
3. Replication Strategies

4. Active+Time

6. Active+Passive.

8. Time+Passive

5. Time+Active

7. Passive+Active

9. Passive+Time
### 3. Replication Strategies

<table>
<thead>
<tr>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>r</strong> 1 ( r = 1 - \prod_{i=1}^{n} (1 - r_i) ; )</td>
</tr>
</tbody>
</table>
| \( t = \begin{cases} 
\min\{T_c\} : |T_c| > 0 ; T = \{t_1, ..., t_m\} = T_c \cup T_f \\
\max\{T_f\} : |T_c| = 0 
\end{cases} \) |
| **r** 2 \( r = 1 - (1 - r_1)^m ; t = \sum_{i=1}^{m} t_i (1 - r_1)^{i-1} ; \)  |
| **r** 3 \( r = 1 - \prod_{i=1}^{m} (1 - r_i) ; t = \sum_{i=1}^{m} t_i \prod_{k=1}^{i-1} (1 - r_k) \)  |
| **r** 4 \( r = 1 - (\prod_{i=1}^{n} (1 - r_i))^m ; \)  |
| \( t = \sum_{i=1}^{m} t_i (\prod_{j=1}^{v} (1 - r_j))^{i-1} ; t_i = \begin{cases} 
\min\{T_c^i\} : |T_c^i| > 0 \\\n\max\{T_f^i\} : |T_c^i| = 0 
\end{cases} \) |
| **r** 5 \( r = 1 - \prod_{i=1}^{v} (1 - r_i)^m ; \)  |
| \( t = \begin{cases} 
\min\{T_c\} : |T_c| > 0 ; t_i \in T = \sum_{j=1}^{m} t_{ij} (1 - r_i)^{j-1} \\
\max\{T_f\} : |T_c| = 0 
\end{cases} \) |
| **r** 6 \( r = 1 - \prod_{i=1}^{m} \prod_{j=1}^{v} (1 - r_{ij}) ; \)  |
| \( t = \sum_{i=1}^{m} t_i \prod_{k=1}^{i} \prod_{j=1}^{v} (1 - r_{kj}) ; t_i = \begin{cases} 
\min\{T_c^i\} : |T_c^i| > 0 \\\n\max\{T_f^i\} : |T_c^i| = 0 
\end{cases} \) |
| **r** 7 \( r = 1 - \prod_{j=1}^{v} \prod_{i=1}^{m} (1 - r_{ij}) ; \)  |
| \( t = \begin{cases} 
\min\{T_c\} : |T_c| > 0 \\\n\max\{T_f\} : |T_c| = 0 
\end{cases} \) |
| **r** 8 \( r = 1 - \prod_{i=1}^{u} (1 - r_i)^m ; \)  |
| \( t = \sum_{i=1}^{u} (t \prod_{j=1}^{m} (1 - r_i)^{j-1}) \prod_{k=1}^{i} (1 - r_k)^m ; \) |
| **r** 9 \( r = 1 - \prod_{i=1}^{u} (1 - r_i)^m ; \)  |
| \( t = \sum_{i=1}^{m} (t \prod_{j=1}^{u} (1 - r_k) (\prod_{j=1}^{u} (1 - r_j))^{i-1}) ; \) |
4. Selection Algorithm

• Objective evaluation results of Web services.
• Subjective requirement of service users
  – *t-user*: 
    • represents the user requirement on response time improvement of increasing one parallel replica.
    • designed to facilitate the user to make a tradeoff between the response time performance and resource consuming.
  – *f-user*:
    • the failure-rate requirement provided by users.
4. Selection Algorithm

- Determining parallel replica number: $v$.
- Determining detailed optimal strategy based on: $p1,p2,p3$. 

$$S_i = \frac{t_i}{t_{user}} + \frac{f_i}{f_{user}}.$$ 

$$W = \{w s_i | s_i \leq a \& \& 1 \leq i \leq n\}$$ 

$$p_1 = s_2 - s_1$$ 

$$p_2 = \frac{1}{v} \sum_{i=1}^{v} (s_{i+v} - s_i)$$ 

$$p_3 = \frac{1}{v} \sum_{i=1}^{v} f_i$$
5. Experiments

- **JDK + Eclipse**
- **Client-side:**
  - Java Applet
- **Server-side:**
  - an HTTP Web site (Apache HTTP Server)
  - a TestCaseGenerator (JDK6.0 + Axis library)
  - a TestCoordinator (Java Servlet + Tomcat 6.0)
  - a MySQL database (Record testing results)
5. Experiments

1. Evaluating the performance of individual Web Services.
2. Evaluating the performance of different fault tolerance strategies employing the six identical Web services provided by Amazon.
3. Determining the optimal fault tolerance strategy.
5.1 Results-individual WS

Table 3. Evaluation Results of the Eight Target Web Services

<table>
<thead>
<tr>
<th>Location</th>
<th>Cases</th>
<th>RTT (ms)</th>
<th>Location</th>
<th>Cases</th>
<th>RTT (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg</td>
<td>Std</td>
<td>Min</td>
<td>Max</td>
<td>Avg</td>
</tr>
<tr>
<td>cn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-us</td>
<td>484</td>
<td>109</td>
<td>22.52</td>
<td>4184</td>
<td>2348</td>
</tr>
<tr>
<td>a-jp</td>
<td>482</td>
<td>128</td>
<td>26.55</td>
<td>3892</td>
<td>2515</td>
</tr>
<tr>
<td>a-de</td>
<td>487</td>
<td>114</td>
<td>23.40</td>
<td>3666</td>
<td>2604</td>
</tr>
<tr>
<td>a-ca</td>
<td>458</td>
<td>111</td>
<td>24.23</td>
<td>4074</td>
<td>2539</td>
</tr>
<tr>
<td>a-fr</td>
<td>498</td>
<td>96</td>
<td>19.27</td>
<td>3654</td>
<td>2514</td>
</tr>
<tr>
<td>a-uk</td>
<td>493</td>
<td>100</td>
<td>20.28</td>
<td>3985</td>
<td>2386</td>
</tr>
<tr>
<td>GW</td>
<td>409</td>
<td>337</td>
<td>82.39</td>
<td>6643</td>
<td>2003</td>
</tr>
<tr>
<td>GIP</td>
<td>540</td>
<td>32</td>
<td>5.92</td>
<td>2125</td>
<td>1927</td>
</tr>
</tbody>
</table>

| au       |       |          |         |       |        |          |         |       |        |          |       |        |         |       |
| a-us     | 1140  | 0       | 0       | 705   | 210   | 500      | 3782    | a-us   | 1895  | 0     | 0      | 561  | 353   | 297  | 4406   |
| a-jp     | 1143  | 0       | 0       | 577   | 161   | 406      | 2594    | a-jp   | 1120  | 0     | 0      | 503  | 322   | 250  | 3687   |
| a-de     | 1068  | 0       | 0       | 933   | 272   | 672      | 6094    | a-de   | 1511  | 0     | 0      | 638  | 409   | 375  | 4735   |
| a-ca     | 1113  | 0       | 0       | 697   | 177   | 500      | 2672    | a-ca   | 1643  | 0     | 0      | 509  | 249   | 297  | 4125   |
| a-fr     | 1090  | 0       | 0       | 924   | 214   | 672      | 2906    | a-fr   | 1635  | 0     | 0      | 638  | 310   | 390  | 5468   |
| a-uk     | 1172  | 3       | 0.25    | 921   | 235   | 672      | 3859    | a-uk   | 1615  | 0     | 0      | 650  | 308   | 375  | 4297   |
| GW       | 1104  | 5       | 0.45    | 503   | 344   | 672      | 9975    | GW     | 1363  | 0     | 0      | 1403 | 1544  | 265  | 9937   |
| GIP      | 1125  | 0       | 0       | 355   | 609   | 234      | 9360    | GIP    | 1312  | 0     | 0      | 571  | 878   | 265  | 9594   |

| lk       |       |          |         |       |        |          |         |       |        |          |       |        |         |       |
| a-us     | 21002 | 81      | 0.38    | 448   | 304   | 230      | 9547    | a-us   | 3725  | 0     | 0      | 74   | 135   | 31   | 3171   |
| a-jp     | 20944 | 11      | 0.05    | 388   | 321   | 203      | 9937    | a-jp   | 3578  | 0     | 0      | 317  | 224   | 109  | 9219   |
| a-de     | 21130 | 729     | 3.45    | 573   | 346   | 343      | 9360    | a-de   | 3766  | 0     | 0      | 298  | 271   | 109  | 9390   |
| a-ca     | 21255 | 125     | 0.58    | 440   | 286   | 250      | 9915    | a-ca   | 3591  | 0     | 0      | 239  | 260   | 31   | 9515   |
| a-fr     | 21091 | 743     | 3.52    | 575   | 349   | 343      | 9703    | a-fr   | 3933  | 0     | 0      | 433  | 222   | 187  | 3906   |
| a-uk     | 20830 | 807     | 3.87    | 570   | 348   | 328      | 9734    | a-uk   | 3614  | 0     | 0      | 293  | 260   | 124  | 9157   |
| GW       | 21148 | 1426    | 6.74    | 1563  | 1560  | 406      | 9999    | GW     | 3837  | 0     | 0      | 1290 | 1346  | 125  | 9828   |
| GIP      | 21007 | 1263    | 6.01    | 849   | 1582  | 203      | 9999    | GIP    | 3621  | 0     | 0      | 675  | 1348  | 125  | 9938   |

- Timeout: 3865; Unavailable service (http 503): 2456; Bad gateway (http 502): 1
- Failure-rates are vary from location to location
5.1 Results-individual WS

- Response time performance (RTT) are vary from location to location.
5.2 Results-FT strategies

<table>
<thead>
<tr>
<th>Type</th>
<th>Cases All</th>
<th>Fail</th>
<th>R%</th>
<th>RTT(ms) Avg</th>
<th>Std</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21556</td>
<td>6</td>
<td>0.027</td>
<td>279</td>
<td>153</td>
<td>203</td>
<td>3296</td>
</tr>
<tr>
<td>2</td>
<td>22719</td>
<td>0</td>
<td>0</td>
<td>389</td>
<td>333</td>
<td>203</td>
<td>17922</td>
</tr>
<tr>
<td>3</td>
<td>23040</td>
<td>0</td>
<td>0</td>
<td>374</td>
<td>299</td>
<td>203</td>
<td>8312</td>
</tr>
<tr>
<td>4</td>
<td>21926</td>
<td>4</td>
<td>0.018</td>
<td>311</td>
<td>278</td>
<td>203</td>
<td>10327</td>
</tr>
<tr>
<td>5</td>
<td>21926</td>
<td>1</td>
<td>0.004</td>
<td>312</td>
<td>209</td>
<td>203</td>
<td>10828</td>
</tr>
<tr>
<td>6</td>
<td>21737</td>
<td>2</td>
<td>0.009</td>
<td>311</td>
<td>225</td>
<td>203</td>
<td>10282</td>
</tr>
<tr>
<td>7</td>
<td>21737</td>
<td>2</td>
<td>0.009</td>
<td>310</td>
<td>240</td>
<td>203</td>
<td>13953</td>
</tr>
<tr>
<td>8</td>
<td>21735</td>
<td>0</td>
<td>0</td>
<td>411</td>
<td>1130</td>
<td>203</td>
<td>51687</td>
</tr>
<tr>
<td>9</td>
<td>21808</td>
<td>0</td>
<td>0</td>
<td>388</td>
<td>304</td>
<td>203</td>
<td>9360</td>
</tr>
</tbody>
</table>

• Strategy 1 (Active) has the best RTT performance, and the worst fail-rate.

5.3 Optimal strategy selection

Table 5. Scenario 1: Selection Procedure

\[
\begin{align*}
& a = 20; \quad b = 5; \quad c = 5\% \\
& n = 6; \quad g = 21587; \quad t_{user} = 100; \quad f_{user} = 0.1\% \\
& \{ws_i\}_{i=1}^{6} = \{\text{a-jp, a-us, a-ca, a-de, a-fr, a-uk}\}; \\
& \{t_i\}_{i=1}^{6} = \{388, 448, 440, 573, 575, 570\}; \\
& \{f_i\}_{i=1}^{6} = \{0.05\%, 0.38\%, 0.58\%, 3.45\%, 3.52\%, 3.87\%\}; \\
& \{s_i\}_{i=1}^{6} = \{4.38, 8.28, 10.2, 40.23, 40.95, 44.4\}; \\
& \{T(i)\}_{i=1}^{6} = \{321, 285, 282, 281, 280, 279\}; \\
& v = 1; \\
& W = \{ws_i | s_i = \leq 20 & \& 1 \leq i \leq 6\} = \{4.38, 8.28, 10.2\}; \\
& |W| = 3; \\
& p_1 = s_2 - s_1 = 3.9; \\
& v = 1 & \& 1 < |W| < 6 & \& p_1 < 5 \Rightarrow \text{Strategy 9}; \\
\end{align*}
\]

Strategy 9: Passive+Time

\[
S_i = \frac{t_i}{t_{user}} + \frac{f_i}{f_{user}}. \\
W = \{ws_i | s_i \leq a & \& 1 \leq i \leq n\} \\
p_1 = s_2 - s_1 \\
p_2 = \frac{1}{v} \sum_{i=1}^{v} (s_i + v - s_i) \\
p_3 = \frac{1}{v} \sum_{i=1}^{v} f_i
\]
6. Conclusion and future work

● Conclusion
  ○ Distributed evaluation framework
  ○ Fault tolerance replication strategies.
  ○ Optimal replication strategy selection algorithm.
  ○ Experiment
    ● More than 1,000,000 test cases.
    ● Users from six locations.
    ● Web Services located in six countries.

● Future work
  ○ Evaluation of stateful Web services.
  ○ Tuning of the selection algorithm
  ○ Investigating more QoS properties.
A Distributed Replication Strategy Evaluation and Selection Framework for Fault Tolerant Web Services

Zibin Zheng and Michael R. Lyu

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

ICWS 2008, Beijing, China, 24 September, 2008