A Runtime Dependability Evaluation 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

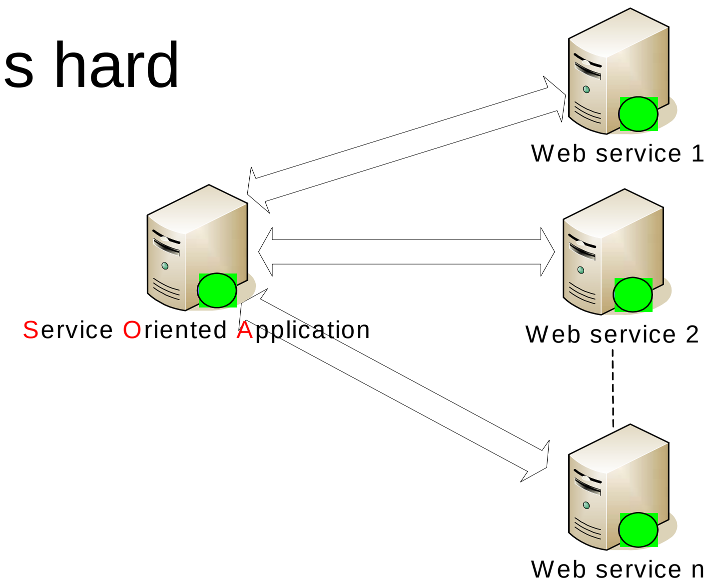
PFARM 2009, Lisbon, Portugal, June 29, 2009

Outlines

1. Introduction
2. System Architecture
3. Methodology
4. Experiments
5. Conclusion

1.1 Web services

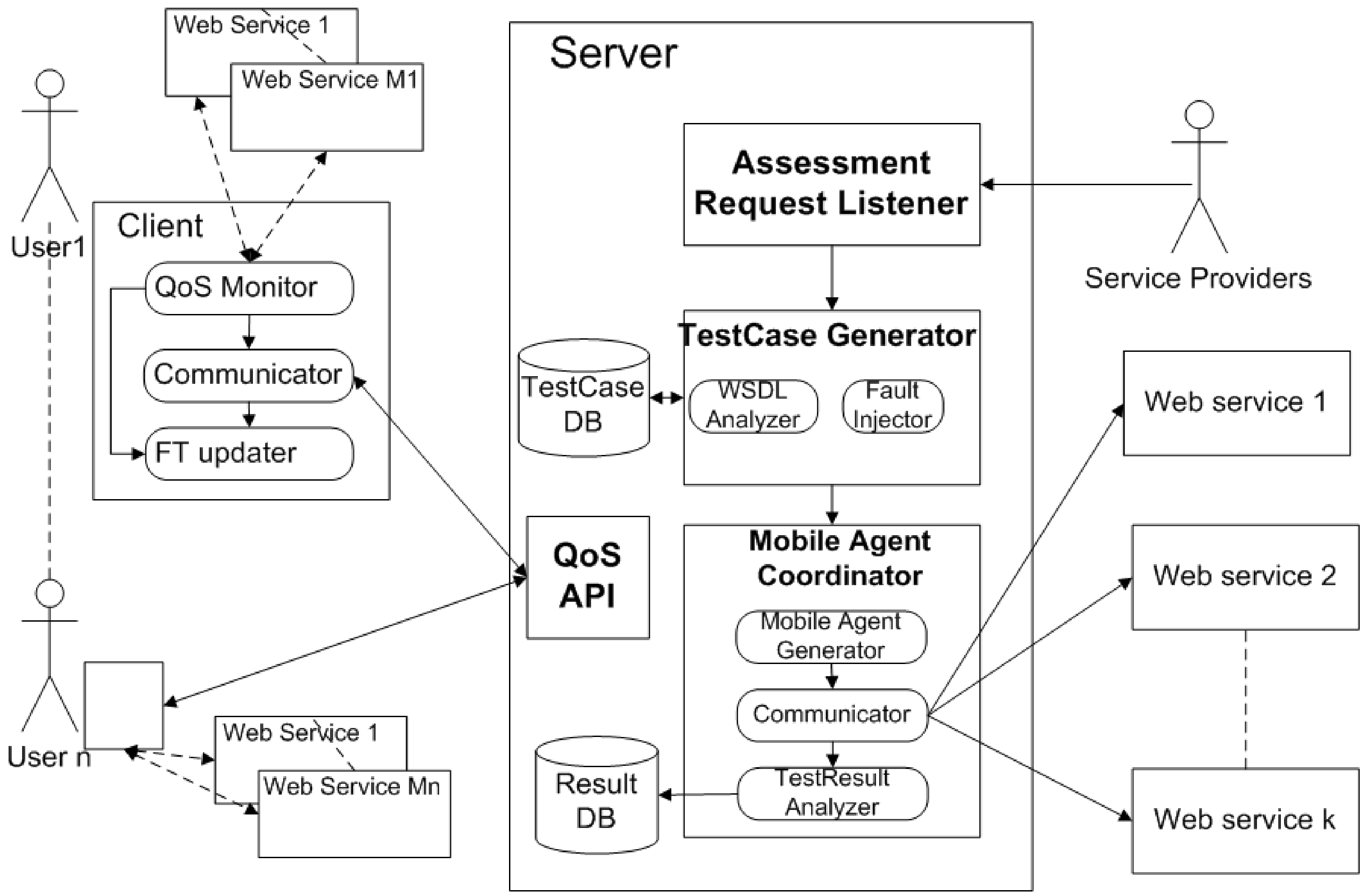
- Web service is becoming popular.
- Difficult to build reliable service-oriented systems
  - Internet and Web services are highly dynamic
  - Web service components are provided by different organizations
  - The compositional nature is hard to predict



1.2 Fault Tolerant Web Services

- Fault tolerant Web services by design diversity
  – Becomes less expensive, as the alternative components are available in the Internet
  – Requires adaptation, as the traditional fault tolerance strategies are too static
- Runtime evaluation on the target Web services
- Runtime proactive reconfiguration of the fault tolerance strategy

2 System Architecture



3.1 QoS of Web Services

- QoS offered by service provider:
  - Availability
  - Price
  - Popularity
  - Datasize
- QoS observed by service users:
  - Failure-rate
  - Response-time
  - Overall failure-rate
  - Overall response-time

$$q = (q^1, ..., q^m),$$

3.2 Fault Tolerance Strategy

1) Time

2) Active

3) Passive

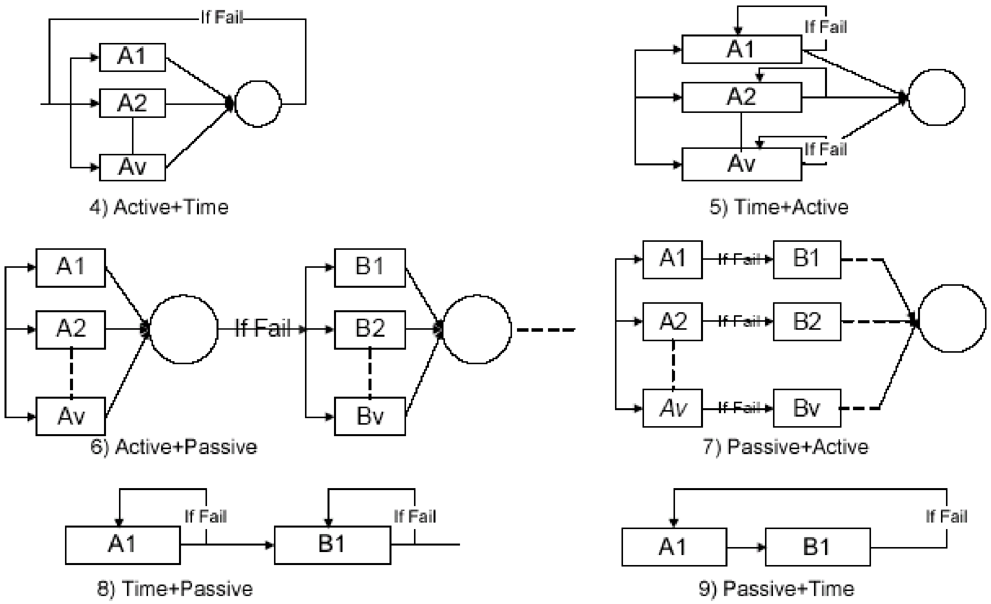


Table 1. Formula for FT Strategies

	Formula
1	$r = 1 - \prod_{i=1}^{n}(1 - r_i);$ $t = \begin{cases} \min\{T_c\} : \lvert T_c \rvert > 0 \\ \max\{T_f\} : \lvert T_c \rvert = 0 \end{cases} ; T = \{t_1, ..., t_n\} = T_c \cup T_f$
2	$r = 1 - (1 - r_1)^m; t = \sum_{i=1}^{m} t_i (1 - r_1)^{i-1};$
3	$r = 1 - \prod_{i=1}^{m}(1 - r_i); t = \sum_{i=1}^{m} t_i \prod_{k=1}^{i-1}(1 - r_k)$
4	$r = 1 - (\prod_{i=1}^{v}(1 - r_i))^m;$ $t = \sum_{i=1}^{m} t_i (\prod_{j=1}^{v}(1 - r_j))^{i-1}; t_i = \begin{cases} \min\{T_c^i\} : \lvert T_c^i \rvert > 0 \\ \max\{T_f^i\} : \lvert T_c^i \rvert = 0 \end{cases}$
5	$r = 1 - \prod_{i=1}^{v}(1 - r_i)^m;$ $t = \begin{cases} \min\{T_c\} : \lvert T_c \rvert > 0 \\ \max\{T_f\} : \lvert T_c \rvert = 0 \end{cases} ; t_i \in T = \sum_{j=1}^{m} t_{ij}(1 - r_i)^{j-1}$
6	$r = 1 - \prod_{i=1}^{m}\prod_{j=1}^{v}(1 - r_{ij});$ $t = \sum_{i=1}^{m} t_i \prod_{k=1}^{i-1}\prod_{j=1}^{v}(1 - r_{kj}); t_i = \begin{cases} \min\{T_c^i\} : \lvert T_c^i \rvert > 0 \\ \max\{T_f^i\} : \lvert T_c^i \rvert = 0 \end{cases}$
7	$r = 1 - \prod_{j=1}^{v}\prod_{i=1}^{m}(1 - r_{ij});$ $t = \begin{cases} \min\{T_c\} : \lvert T_c \rvert > 0 \\ \max\{T_f\} : \lvert T_c \rvert = 0 \end{cases} ; t_i = \sum_{j=1}^{m} t_{ij} \prod_{k=1}^{j-1}((1 - r_{ik})$
8	$r = 1 - \prod_{i=1}^{u}(1 - r_i)^m;$ $t = \sum_{i=1}^{u}((\sum_{j=1}^{m} t_i (1 - r_i)^{j-1}) \prod_{k=1}^{i-1}(1 - r_k)^m);$
9	$r = 1 - (\prod_{i=1}^{u}(1 - r_i))^m;$ $t = \sum_{i=1}^{m}((\sum_{j=1}^{u} t_j \prod_{k=1}^{j-1}(1 - r_k)(\prod_{j=1}^{u}(1 - r_j))^{i-1});$

3.3 Optimal Web Service Selection

- Method 1: rank target Web services by their overall QoS performance (OP)

$$OP = \sum_{i=1}^{m} w_i \tilde{q}^i, \qquad \tilde{q}^i = \frac{1}{n} \sum_{j=1}^{n} q^{ij},$$

- Weak point: new service users may not obtain similar performance
  - The network condition is different
  - The user geographic location is different

3.3 Optimal Web Service Selection

- Method 2: collaborative filtering algorithm
  - Similarity computation

$$Sim(a,u) = \frac{\sum_{i \in I}(r_{a,i} - \overline{r}_a)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I}(r_{a,i} - \overline{r}_a)^2}\sqrt{\sum_{i \in I}(r_{u,i} - \overline{r}_u)^2}},$$

  - Similar user selection
  - QoS performance prediction

$$P(r_{u,i}) = \overline{u} + \frac{\sum_{u_a \in S(u)} Sim'(u_a,u)(r_{u_a,i} - \overline{u}_a)}{\sum_{u_a \in S(u)} Sim'(u_a,u)},$$

  - Optimal Web service selection

3.4 Dynamic FT Strategy Reconfiguration

- Update QoS performance of target Web service candidates dynamically.
- Update user requirements.
  - e.g., response-time < 1000 ms.
  - e.g., failure-rate < 5%.
- Dynamic optimal fault tolerance strategy determination
  - Zibin Zheng, Michael R. Lyu, “A QoS-Aware Fault Tolerant Middleware for Dependable Service Composition”, *in DSN2009*.

4.1 Experimental setup

- Prototype: http://www.wsdream.net
  - Client: Java applet.
  - Server: Java servlet, MySQL
- Service users: CN, TW, AU, SG, HK, US
- Functionally equivalent Amazon Web services: a-us, a-jp, a-de, a-ca, a-fr, a-uk.

http://ecs.amazonaws.com
http://ecs.amazonaws.jp
http://ecs.amazonaws.de
http://ecs.amazonaws.ca
http://ecs.amazonaws.fr
http://ecs.amazonaws.co.uk

4.2 Evaluation of Individual WS

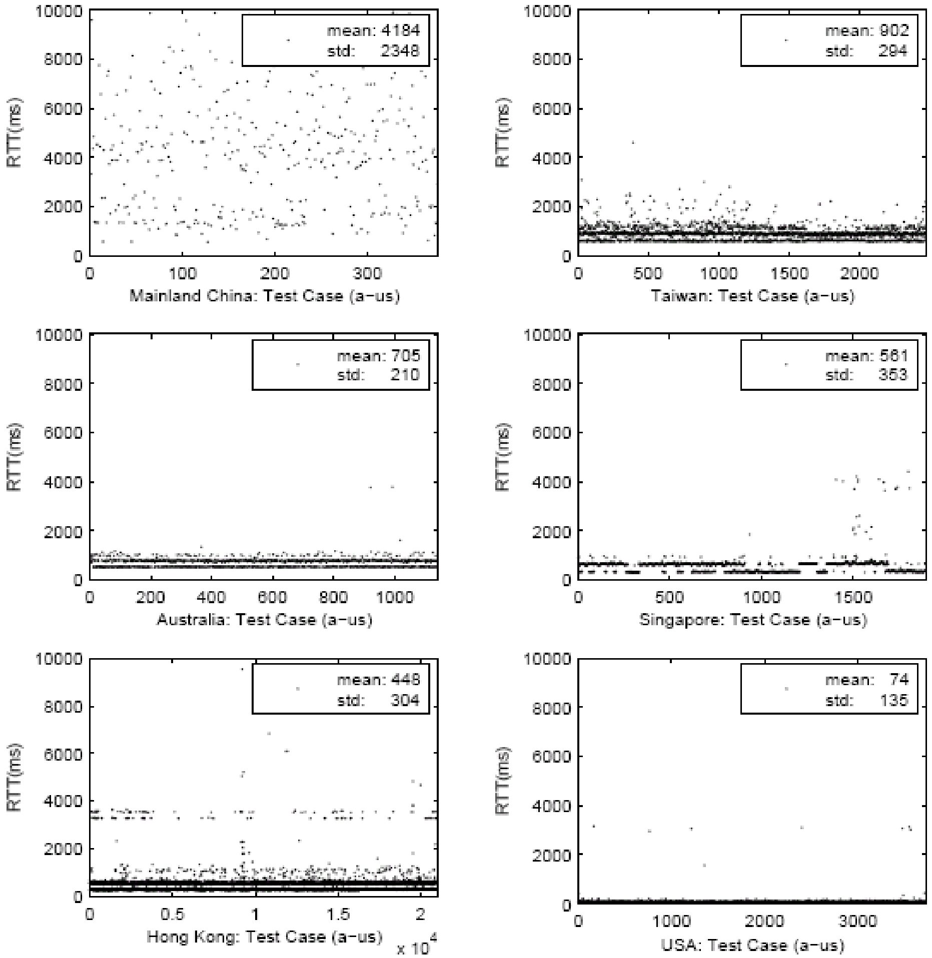


Table 2. Failure-rate of the Web Services

WS	CN	TW	AU	SG	HK	US
a-us	22.52	0	0	0	0.38	0
a-jp	26.55	0.03	0	0	0.05	0
a-de	23.40	0	0	0	3.45	0
a-ca	24.23	0.19	0	0	0.58	0
a-fr	19.27	0	0	0	3.52	0
a-uk	20.28	0.03	0.25	0	3.87	0

4.3 Evaluation of FT Strategies

Table 3. Evaluation of FT Strategies

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

5.1 Conclusion

- QoS of Web services
- Fault tolerance strategies
- Web service selection framework
- Proactive reconfiguration of optimal FT strategy