# 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**

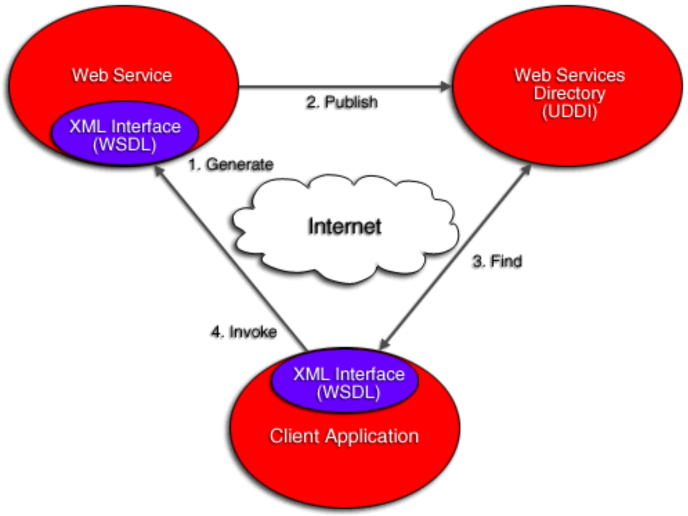
# Outlines

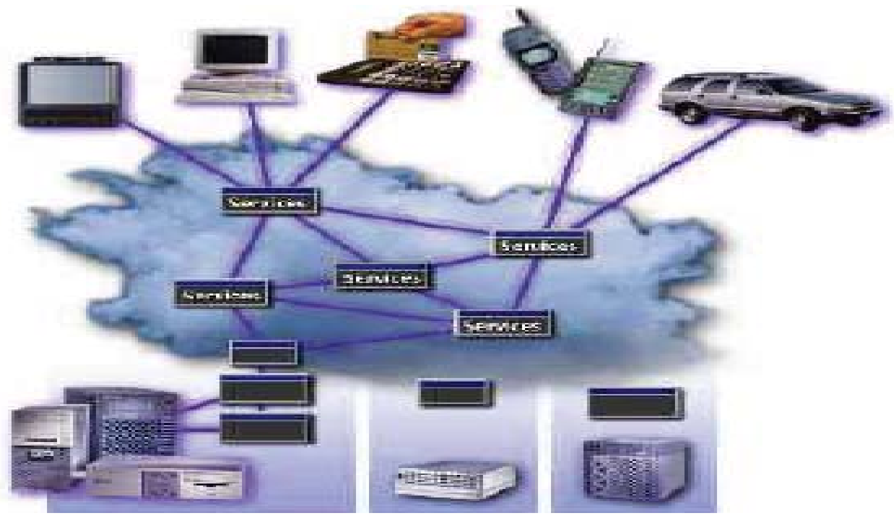
1. Introduction
2. Preliminaries
3. Optimal Fault Tolerance Strategy Selection
4. Experiments
5. Conclusion and Future Work

# 1. Introduction

# 1.1 Web Services

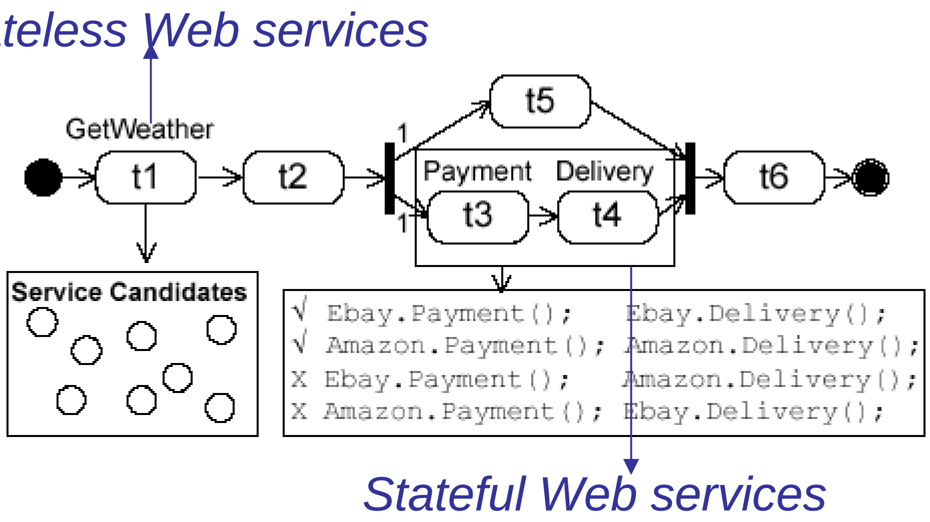
- Self-description
- Loosely-coupled
- Highly-dynamic
- Cross-domain
- Compositional nature

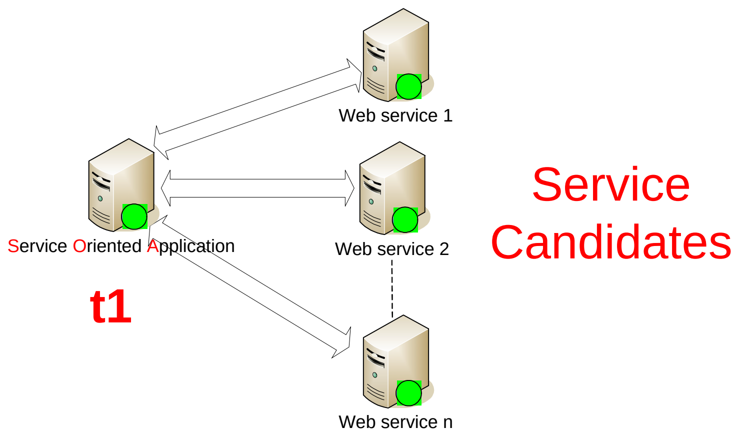


# 1.2 A Motivating Example

- 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 t1 < 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 system.
  - 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.

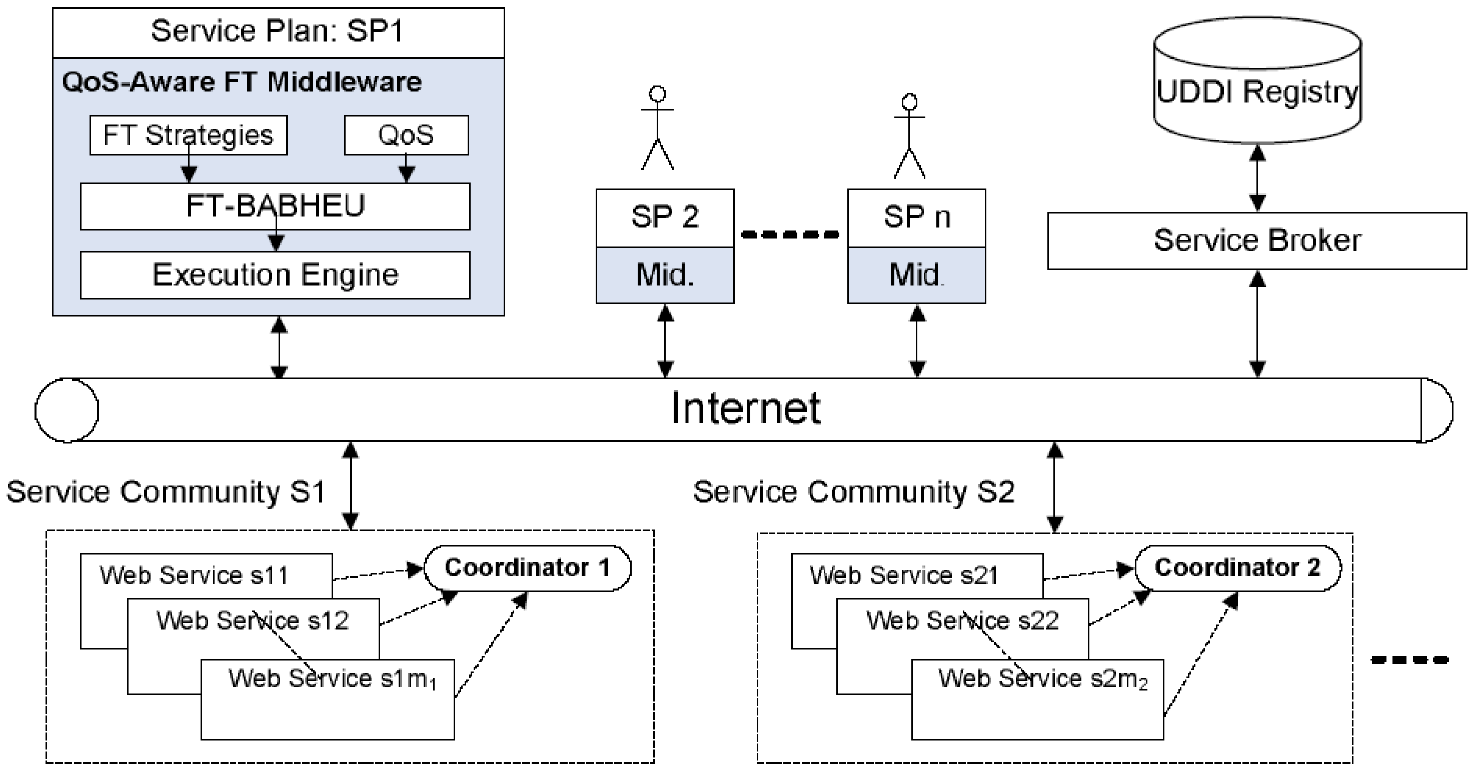
How to employ these redundant Web services for building fault tolerant services reliably and effectively?

# 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. $q^5 = \frac{succInvocationNum}{totalInvocationNum}$.

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.

$$q = (q^1, \ldots, q^8).$$

# 2.3 Web Service Composition

| QoS Properties | Basic Structures | | | |
|---|---|---|---|---|
| | sequence | parallel | branch | loop |
| rt, ort (x=6, 8) | $\sum_{i=1}^{n} q_i^x$ | $\max_{i=1}^{k} q_i^x$ | $\sum_{i=1}^{n} p_i q_i^x$ | $\sum_{i=0}^{n} p_i q_1^x i$ |
| av, sr, osr (x=1, 5, 7) | $\prod_{i=1}^{n} q_i^x$ | $\sum_{i=k}^{n} S^x(i)$ | $\sum_{i=1}^{n} p_i q_i^x$ | $\sum_{i=0}^{n} p_i (q_1^x)^i$ |
| pr, po, ds (x=2, 3, 4) | $\sum_{i=1}^{n} q_i^x$ | $\sum_{i=1}^{n} q_i^x$ | $\sum_{i=1}^{n} p_i q_i^x$ | $\sum_{i=0}^{n} p_i q_1^x i$ |

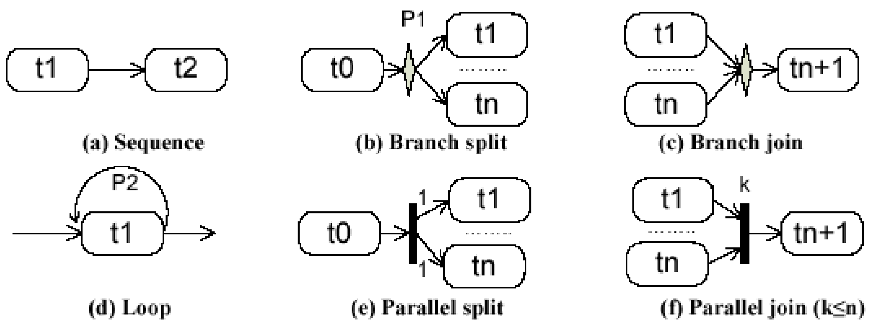



Figure 2. Basic Compositional Structures

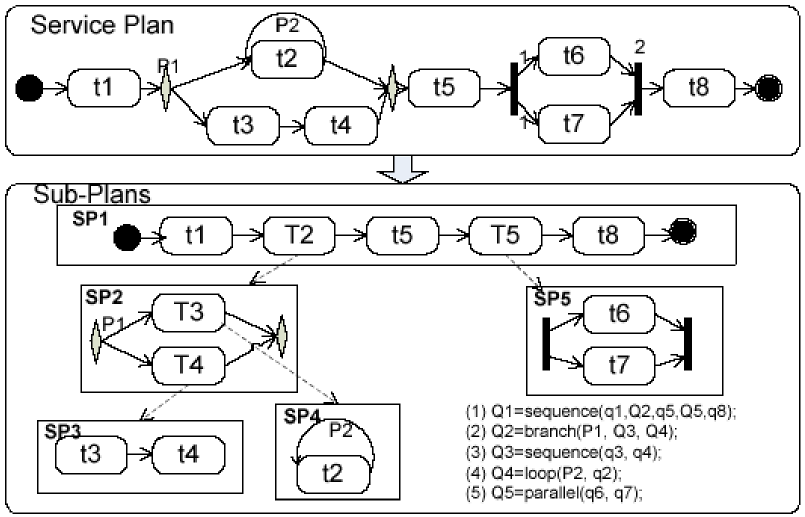



Figure 3. Service Plan Decomposition

# 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.

$$q_{ij}^k = \max q_i^k - q_{ij}^k,$$

- Normalization of the QoS values.

$$\tilde{q}_{ij}^k = \begin{cases} \frac{q_{ij}^k - \min q_i^k}{\max q_i^k - \min q_i^k} & \text{if} \quad \max q_i^k \neq \min q_i^k \\ 1 & \text{if} \quad \max q_i^k = \min q_i^k \end{cases}$$

- Utility function:

$$u_{ij} = utility(q_{ij}) = \sum_{k=1}^{c} w_k \times \tilde{q}_{ij}^k,$$

# 3.2 Notations

**Table 3. Notations**

| Symbol | Description |
|---|---|
| $SP$ | a service plan, which is a triple $(T, P, B)$. |
| $T$ | a set of tasks in the service plan, $T = SLT \cup SFT$. |
| $SLT$ | a set of stateless tasks, $SLT$=$\{t_i\}_{i=1}^{n_l}$. |
| $SFT$ | a set of stateful tasks, $SFT$=$\{SFT_i\}_{i=n_l}^{n}$ |
| $SFT_i$ | a set of related tasks of the $i^{th}$ stateful task. |
| $n$ | the number of tasks in SP, $n$=$n_l$+$n_f$. |
| $n_l$ | the number of the stateless tasks in SP, $n_l$=$\|SLT\|$. |
| $n_f$ | the number of the stateful tasks in SP, $n_f$=$\|SFT\|$. |
| $n_i$ | number of state related tasks of $SFT_i$, $n_i$=$\|SFT_i\|$. |
| $S_i$ | a set of candidates for $t_i$,$S_i$=$\{s_{ij}\}_{j=1}^{m_i}$. |
| $m_i$ | the number of candidates for $t_i$, $m_i = \|S_i\|$. |
| $\rho_i$ | the optimal candidate index for $t_i$. |
| $LC_i$ | local constraints for task $t_i$, $LC_i$=$\{lc_k^i\}_{k=1}^{c}$. |
| $GC$ | global constraints for $SP$, $GC = \{gc^k\}_{k=1}^{c}$. |
| $c$ | the number of quality properties. |
| $q_{ij}$ | a quality vector for $s_{ij}$, $q_{ij}$=$(q_{ij}^k)_{k=1}^{c}$. |
| $ER$ | a set of execution routes of $SP$, $ER = \{ER_i\}_{i=1}^{n_e}$. |
| $n_e$ | the number of execution routes of a service plan. |
| $pro(ER_i)$ | the execution probability of $ER_i$. |
| $SR$ | a set of sequential routes of $SP$, $SR = \{SR_i\}_{i=1}^{n_s}$. |
| $n_s$ | the number of sequential routes of $SP$. |
| $pct$ | a user defined threshold for $ER$. |

# 3.3 Optimal Selection With Local Constraints

- Selection problem

**Problem 1 Minimize:** $\sum_{j=1}^{m_i} u_{ij} x_{ij}$

**Subject to:**

- $\sum_{j=1}^{m_i} q_{ij}^k x_{ij} \leq lc_i^k (k = 1, 2, ..., c)$
- $\sum_{j=1}^{m_i} x_{ij} = 1$
- $x_{ij} \in \{0, 1\}$

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

$n_l$=$|SLT|$; $n_f$=$|SFT|$; n=$n_l$+$n_f$; $n_i$=$|SFT_i|$; $m_i$=$|S_i|$;
**for** *(*$i = 1; i \leq n_l; i$*++)* **do**
  **for** *(*$j = 1; j \leq m_i; j$*++)* **do**
    **if** $\forall x(q_{ij}^x \leq lc_i^x)$ **then** $u_{ij} = utility(q_{ij})$;
  **end**
  **if** *no candidate meet* $lc_i$ **then** Throw exception;
  $u_{ix} = \min\{u_{ij}\}$;
  $\rho_i = x$;
**end**
**for** *(i=*$n_l + 1$*; i*$\leq$ *n; i++)* **do**
  **for** *(j=1; j* $\leq m_i$*; j++)* **do**
    **if** $\forall x \forall y(q_{iyj}^x \leq lc_{iy}^x)$ **then**
      $q = flowQoS(SP, q_{i1j}, .., q_{in_ij})$;
      $u_{ij} = utility(q)$;
    **end**
  **end**
  **if** *no candidate meet* $lc_i$ **then** Throw exception;
  $u_{ix} = \min\{u_{ij}\}$;
  **forall** *tasks in* $SFT_i$ **do** $\rho_{ik} = x$;
**end**

**Algorithm 2**: FT Selection with Local Constraints

# 3.4 Selection With Gobal 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:**

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

$$\forall k, \sum_{i \in SR_{ik}} \sum_{j \in S_i} q_{ij}^y x_{ij} \le gc^y (y = 6, 8)$$

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

$$\forall SFT_i, x_{y_1 j} = x_{y_2 j} = ... = x_{y_{n_i} j} (t_{y_i} \in SFT_i)$$

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

$$\sum_{i \in ER_i} \sum_{j \in S_i} x_{ij} ln(q_{ij}^y) \le ln(gc^y)(y = 1, 5, 7),$$

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

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

# 3.5 Hybrid Algorithm

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

```
Data: SP, ER, Constraints GC, LC, Candidates S, pct
Result: Optimal candidates index ρ for SP.
n=n_l+n_f; n_l=|SLT|; m_i=|s_i|; n_e=|ER|; T_e={};
for (i=1;i ≤ n_e;i++) do
    if ER_i ∈ the first pct major routs then
        FTBAB(ER_i);
        T_e = T_e ∪ T_i;
    end
end
foreach t_k ∈ T_e do
    if t_k ∈ only one ER_i then
        ρ_k = ER_i.ρ_k;
    else if t_k ∈ multiply ER_i then
        pro(ER_x) = max{pro(ER_i)};
        ρ_k = ER_x.ρ_k;
    end
end
if T_e == T then return ρ;
ρ = findInitialSolution(T, GC, LC, S, T_e, ρ);
q_all = flowQoS(SP, q_{1ρ_1}, ..., q_{nρ_n});
while ∃x(q_all^x / gc^x > 1) do
    S'=findExchangeCandidate(T,GC,LC,ρ);
    if |S'| == 0 then
        return No Feasible Solution Exist!
    else
        forall s_xy ∈ S' do ρ_x = y;
    end
end
repeat
    ρ=feasibleUpgrade(SP,GC,LC,S,ρ);
until ρ do not change ;
return ρ;
```

**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) is a global research network, which consists of 1016 distributed computers.

PlanetLab currently consists of 1016 nodes at 479 sites.

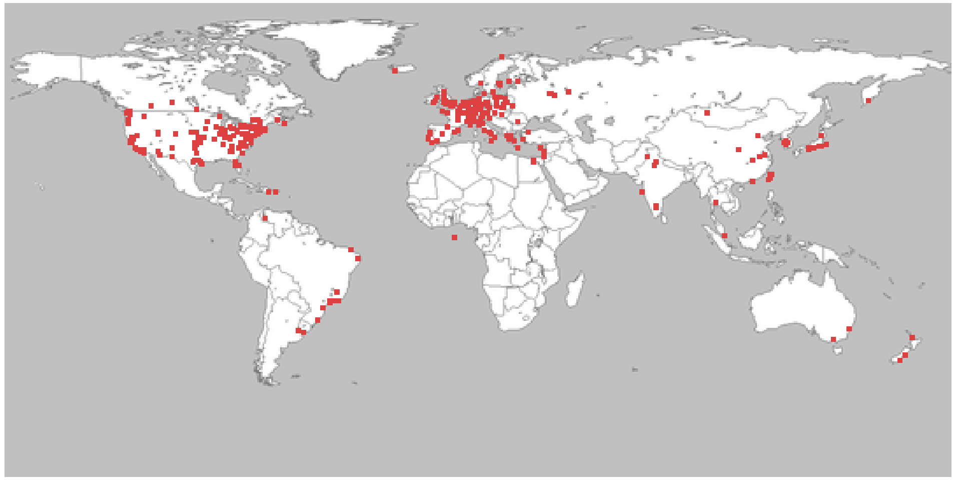
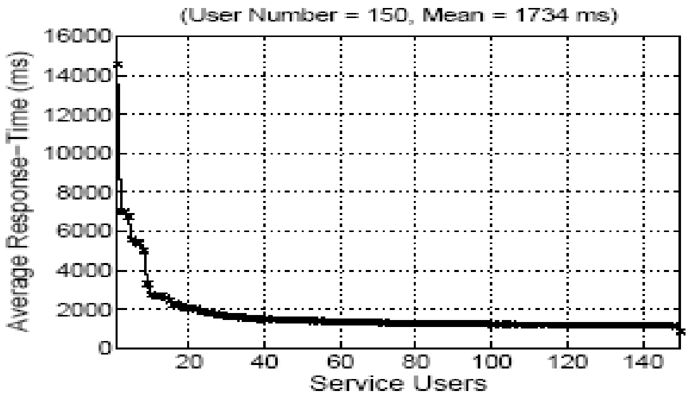


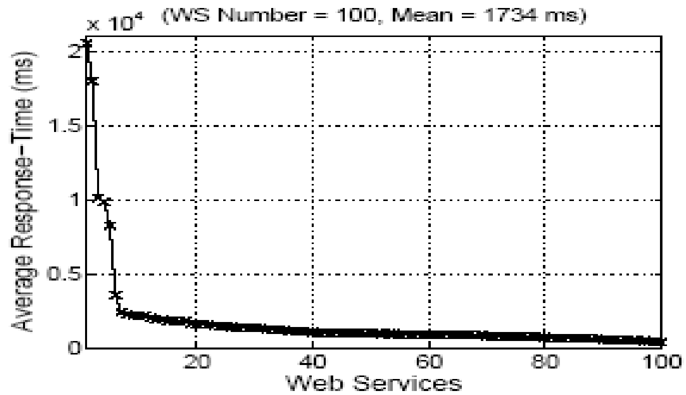
Table 5.1. Locations of the Service Users and Web Services

| User Locations | Num | WS Locations | Num |
|---|---|---|---|
| United States | 72 | United States | 33 |
| European Union | 37 | Canada | 10 |
| Japan | 6 | China | 8 |
| Canada | 5 | Germany | 7 |
| Germany | 4 | France | 6 |
| Brazil | 3 | Spain | 6 |
| France | 3 | United Kingdom | 5 |
| United Kindom | 3 | Netherlands | 4 |
| Republic of Korea | 2 | Poland | 3 |
| Belgium | 1 | Republic of Korea | 3 |
| Cyprus | 1 | Switzerland | 3 |
| Republic of Czech | 1 | Italy | 2 |
| Finland | 1 | Australia | 1 |
| Greece | 1 | Belgium | 1 |
| Hungary | 1 | Ireland | 1 |
| Ireland | 1 | Islamic Republic of Iran | 1 |
| Norway | 1 | Japan | 1 |
| Poland | 1 | New Zealand | 1 |
| Portugal | 1 | Norway | 1 |
| Puerto Rico | 1 | Serbia and Montenegro | 1 |
| Slovenia | 1 | South Africa | 1 |
| Spain | 1 | Thailand | 1 |
| Taiwan | 1 | | |
| Uruguay | 1 | | |
| **Total** | **150** | **Total** | **100** |

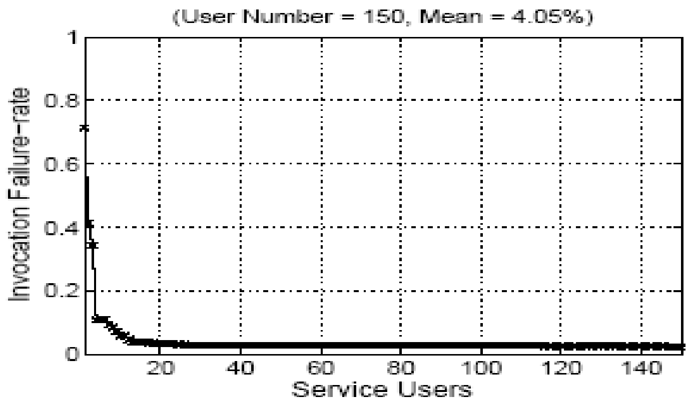
# 4.3 QoS of Web Services



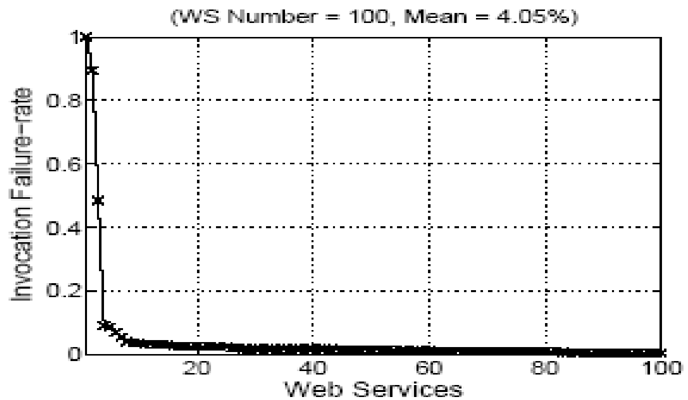

(a) Response-time of Users




(b) Response-time of WS




(c) Failure-rate of Users




(d) Failure-rate of WS

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

# 4.4 Case Studies

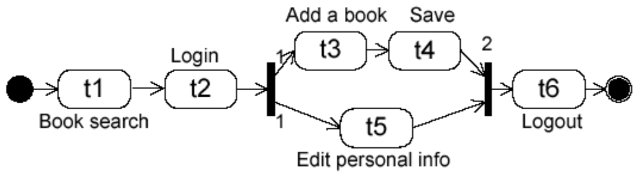
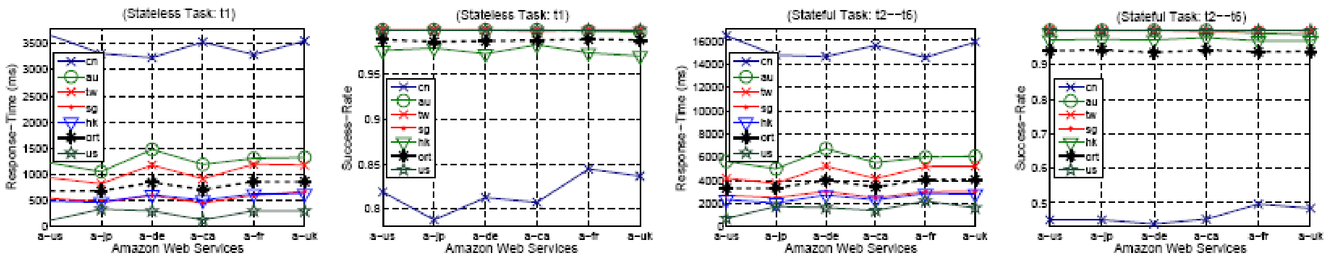



Table 5.2. QoS Values of the Stateless Task ($t_1$)

| WS | Q | CN | AU | US | SG | TW | HK | ALL |
|---|---|---|---|---|---|---|---|---|
| aus | rt | 3659 | 1218 | 121 | 544 | 934 | 491 | 681 |
| | sr | 0.819 | 1.000 | 1.000 | 1.000 | 1.000 | 0.977 | 0.989 |
| ajp | rt | 3310 | 1052 | 338 | 472 | 824 | 469 | 686 |
| | sr | 0.788 | 1.000 | 1.000 | 1.000 | 1.000 | 0.980 | 0.987 |
| ade | rt | 3233 | 1476 | 303 | 596 | 1178 | 612 | 846 |
| | sr | 0.813 | 1.000 | 1.000 | 1.000 | 1.000 | 0.973 | 0.987 |
| aca | rt | 3530 | 1190 | 130 | 456 | 916 | 509 | 714 |
| | sr | 0.807 | 1.000 | 1.000 | 1.000 | 0.998 | 0.983 | 0.988 |
| afr | rt | 3289 | 1309 | 306 | 600 | 1193 | 630 | 864 |
| | sr | 0.844 | 0.998 | 1.000 | 1.000 | 1.000 | 0.974 | 0.989 |
| auk | rt | 3550 | 1326 | 305 | 671 | 1178 | 633 | 862 |
| | sr | 0.837 | 0.997 | 1.000 | 1.000 | 1.000 | 0.971 | 0.988 |

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

| WS | Q | CN | AU | US | SG | TW | HK | ALL |
|---|---|---|---|---|---|---|---|---|
| aus | rt | 16434 | 5625 | 717 | 2708 | 4166 | 2328 | 3297 |
| | sr | 0.450 | 1.000 | 1.000 | 1.000 | 1.000 | 0.972 | 0.940 |
| ajp | rt | 14763 | 4980 | 1751 | 2505 | 3730 | 2058 | 3335 |
| | sr | 0.450 | 1.000 | 1.000 | 1.000 | 0.998 | 0.973 | 0.944 |
| ade | rt | 14640 | 6718 | 1646 | 3038 | 5209 | 2730 | 3985 |
| | sr | 0.438 | 1.000 | 1.000 | 1.000 | 1.000 | 0.972 | 0.935 |
| aca | rt | 15602 | 5527 | 1403 | 2488 | 4150 | 2305 | 3427 |
| | sr | 0.452 | 1.000 | 1.000 | 1.000 | 0.996 | 0.979 | 0.944 |
| afr | rt | 14560 | 5983 | 2211 | 3009 | 5175 | 2862 | 4045 |
| | sr | 0.496 | 0.992 | 1.000 | 1.000 | 1.000 | 0.969 | 0.937 |
| auk | rt | 15898 | 6066 | 1630 | 3044 | 5209 | 2819 | 4048 |
| | sr | 0.484 | 0.988 | 1.000 | 1.000 | 0.998 | 0.970 | 0.939 |




(a) Response-time of t1 (b) Success-rate of t1 (c) Response-time of t2-t6 (d) Success-rate of t2-t6

# 4.5 Performance Study (1)

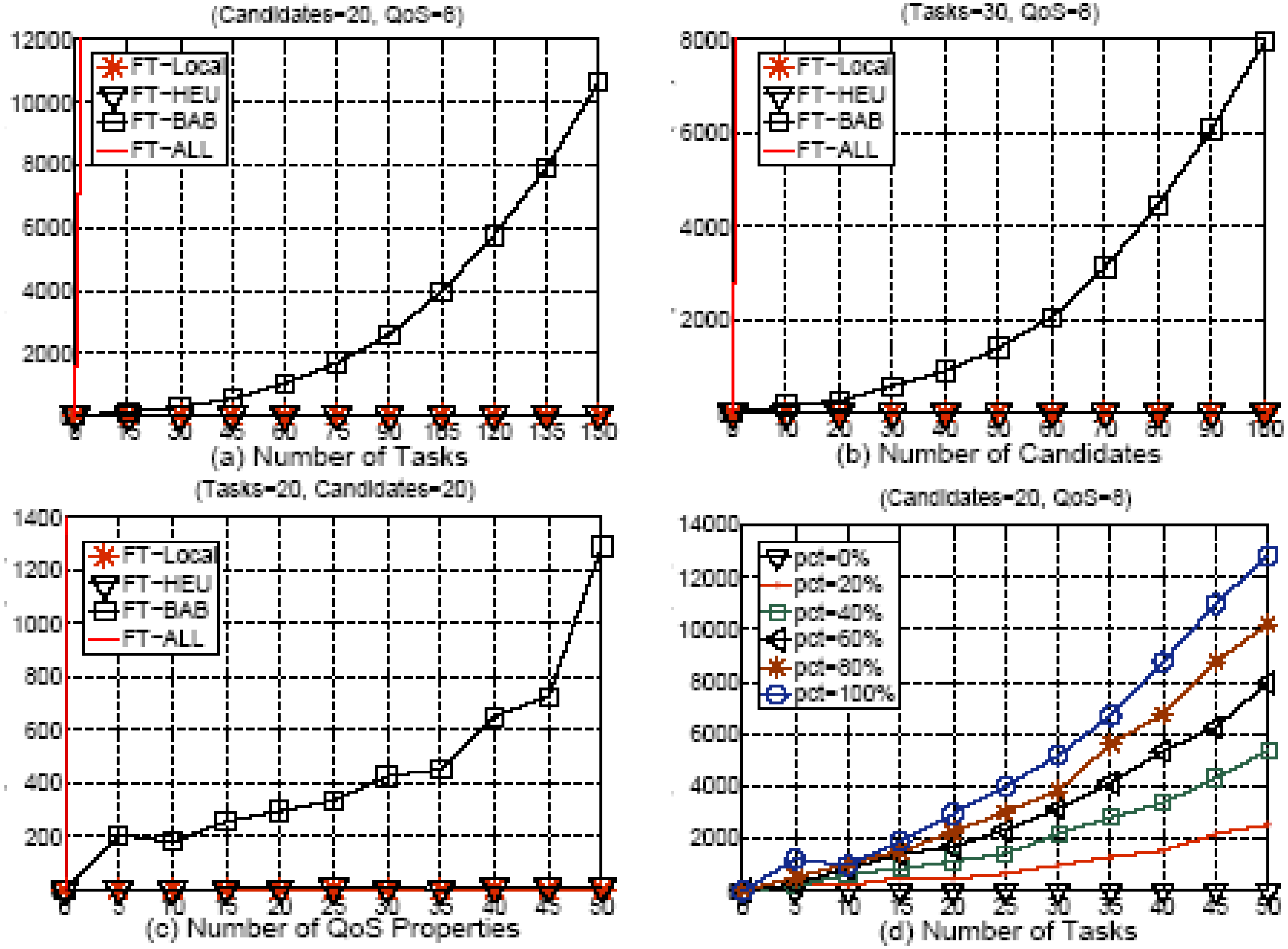



**Figure 6. Performance of Computation Time**

# 4.5 Performance Study (2)

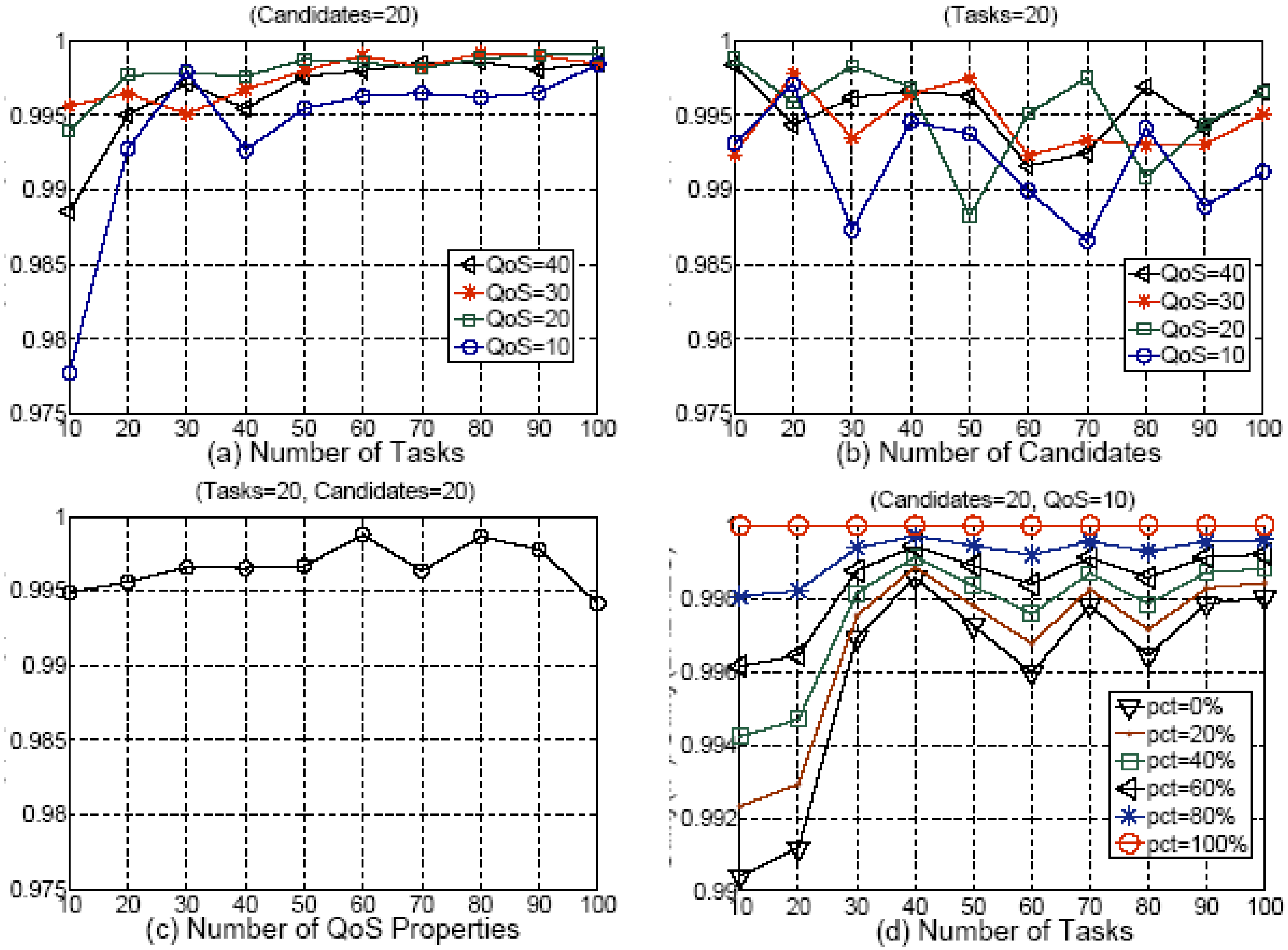



**Figure 7. Performance of Selection Results**

# 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