Dependable Web Services

Present By
Gaspar Modelo Howard
Ratsameetip Wita

DCSL Reading Group 11/07/2007

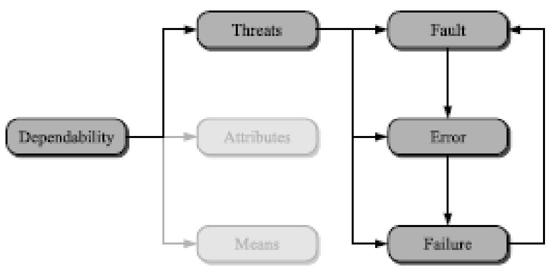
Outline

- Dependability Concept
- Challenges in Web Service Environment
- Research in dependable webservice
  - Conti, M., et al., ***Load distribution among replicated Web servers: a QoS-based approach***, ACM Sigmetrics 2000
  - Moser, L., et al. ***Making Web Services Dependable***. 2006 International Conference on Availability, Reliability, and Security(ARES'06).

Dependability Concept

- Widely understood as reliable ability of the system in supplying user with provided service.
- The working group of the International Federation for Information Processing (IFIP) identified more systematic interpretation of dependability concept in terms of:
  - Attributes of dependability
  - Threats to dependability
  - Means to attain dependability

Dependability Threat



- Fault - hypothesized cause of an error.
- Error - part of the system state that may cause a subsequent failure.
- Failure - delivered service deviates from correct service

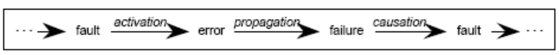
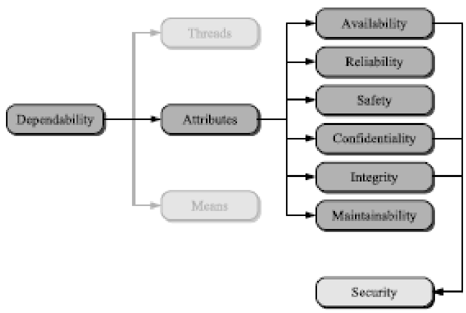


Figure 5 - The fundamental chain of threats to dependability

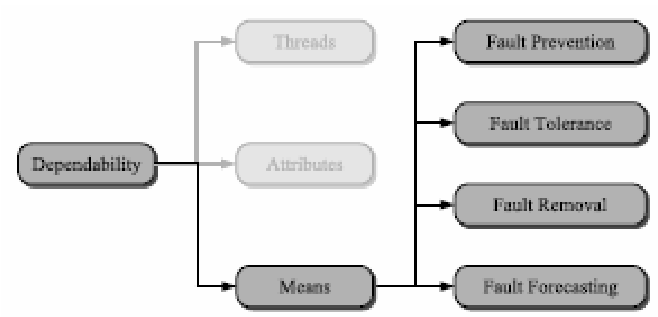
Dependability attributes



- Safety and Security emphasize on the avoidance of a specific class of failures (catastrophic failures, unauthorized access or handling of information, respectively).
- Reliability and Availability emphasize on the avoidance or minimization of service outage.

Means to Attain Dependability

- Appropriate balancing of technique in order to maintain conflicted attributes (e.g. Availability and Security, Availability and Safety).



Means to Attain Dependability

- **Fault prevention** -- attained by deploying proper policy and configurations.
- **Fault tolerance** -- intended to preserve the delivery of correct service in the presence of active faults (error)
  - Error detection and recovery, fault handling
- **Fault removal** -- performed in both development and operation phase of the system
- **Fault forecasting** -- conducted by performing an evaluation of the system behavior with respect to fault occurrence or activation.

Challenges in Web Service Environment

- Practical Implementation.
  - general-purpose, low-cost components such as commodity servers and middleware.
- Out of control resource.
  - Internet bandwidth connection.
  - Internet environment is subject to(real or virtual) partitions

Load distribution among replicated Web servers: a QoS-based approach

Marco Conti, Enrico Gregori, and Fabio Panzieri
ACM Sigmatrics 2000

Outline

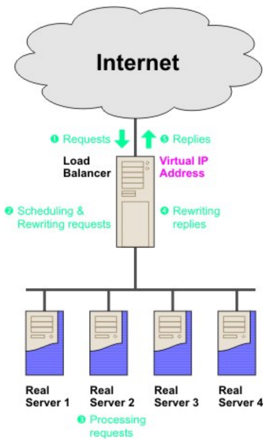
- Introduction
- Load distribution strategies
- Performance comparison simulation and result
- QoS based Architecture design
- Conclusion

Introduction

- A practical approach to provide a responsive Web services is based on introducing redundancy.
- To ensure that each client (or browser) gets bound to the *"most convenient"* WS replica.
- The word *"most convenient"* is defined as a particular replica that can provide the client with the shortest User Response Time –URT.
- URT includes both communication and processing time.
- In this paper, automatically binding between client and most convenient will be discussed.
  - Not include maintain data consistency among replicas issue.

Load Distribution Strategies

- DNS-based
  - Use DNS as scheduler of browser's request.
  - Using Round-Robin discipline.
  - Maximize data throughput, rather than minimizing URT.



Load Distribution Strategies

- Mirror-based
  - Provide most geographically closer to that browser.
  - Automatically redirect/ User manually selects a replica.

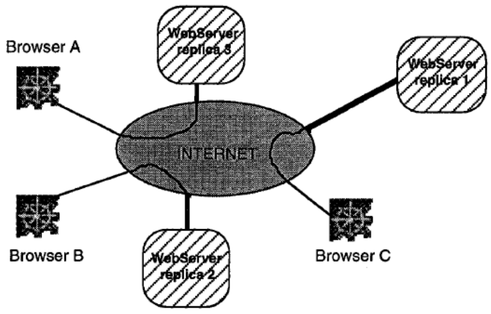
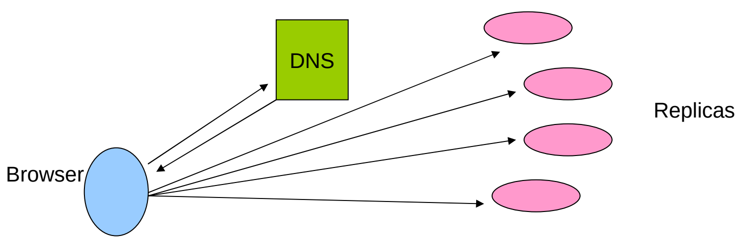


Figure 6: Replicated Web service

QOS based Load Distribution Strategy

- Implemented at browser (client side)
- On assumption that DNS provides all replicas addresses in service negotiate process. (change DNS implementation)
- Before sending a query, browser sent out “dummy request” in order to get URT of replicas.
- Browser select satisfactory URT replica.



Performance Comparison

- QOS-based, DNS based and Mirror based load distribution.
- Simulation considering four replicated service, located in four distinct geographical areas.
- Time interval between consecutive queries are independent and exponentially distributed.
- A query corresponds to retrieval of 10 web pages with median size 3000 byte, in average.
- Dummy request in QOS-based corresponds to the retrieval of a 1000 byte page.
- Compare by analyzing relative impact on Query Response time *(Query RT)*

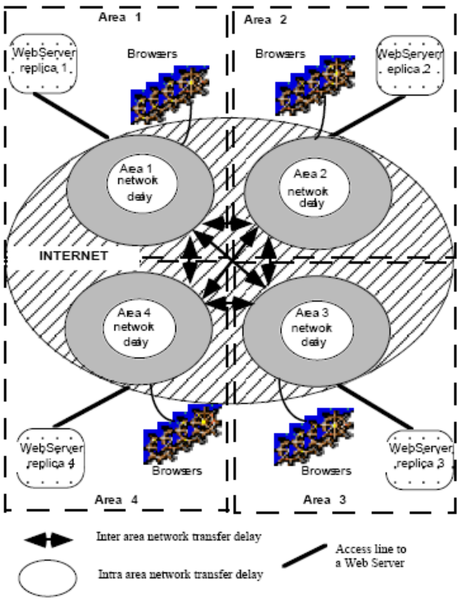


Figure 1: Simulation scenario

- Intra-area delay
  - RTT, Queuing delay, transmission delay
- Inter-area delay
  - Communication latency (i.i.d. random variable)

Simulation Scenario

Exp1 Intra-area network congestion

- Congested router in area 1 (98% utilization)
- Maximum utilization routers in other areas (80%)

Exp2 A heavily load area

- Saturation point of WS replica in area 1 (0.98 of server capacity)
- 0.80 of local WS replicas capacity in other areas.

Exp3 Symmetric cases

- Each are has network utilization at 80% and WS load at 0.8 of server capacity.

Exp 4 A realistic Scenario

- Load difference among areas. (due to different in period of time among distinct geographical area)

	Area 1	Area 2	Area 3	Area 4
network	0.98	0.80	0.50	0.10
query rate	0.98	0.80	0.50	0.10

Table 3: Load configuration for the realistic scenario

Load distribution with the QoS Strategy

	Area 1 Server	Area 2 Server	Area 3 Server	Area 4 Server
Exp 1	0.58	0.91	0.92	0.92
Exp 2	0.95	0.86	0.85	0.85
Exp 3	0.83	0.83	0.83	0.84
Exp 4	0.44	0.85	0.69	0.51

Table 1: Load distribution with the QoS strategy

Exp1 Intra-area network congestion

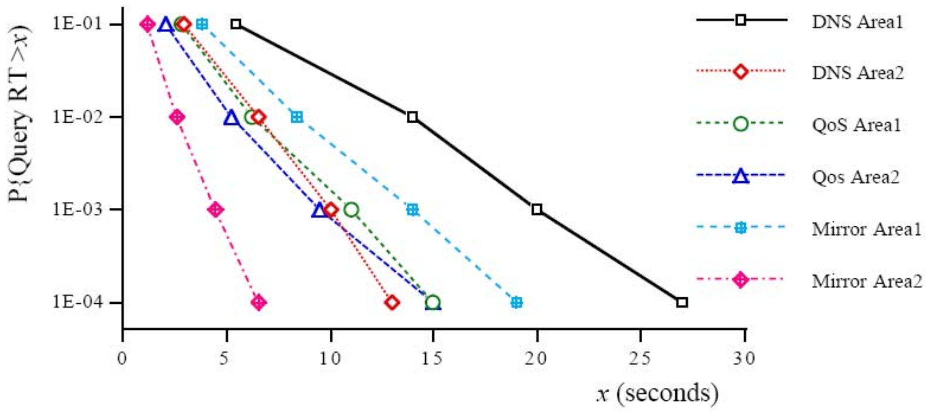


Figure 2: Impact of intra-area network congestion on the query response time

Exp 2 Heavily loaded area in area 1

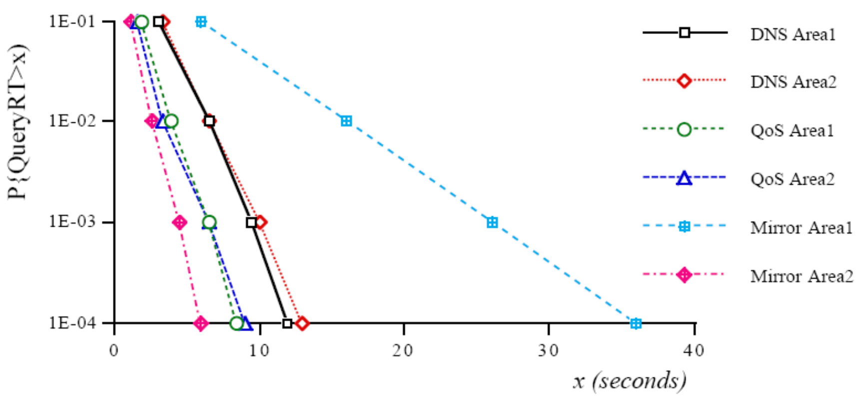


Figure 3: Impact of a heavily loaded area on the query response time

Symmetric case

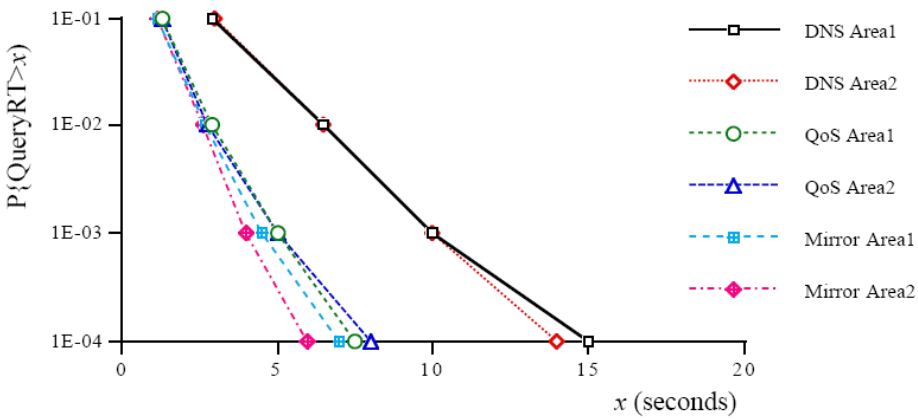


Figure 4: Query response time in a symmetric case

Realistic scenario

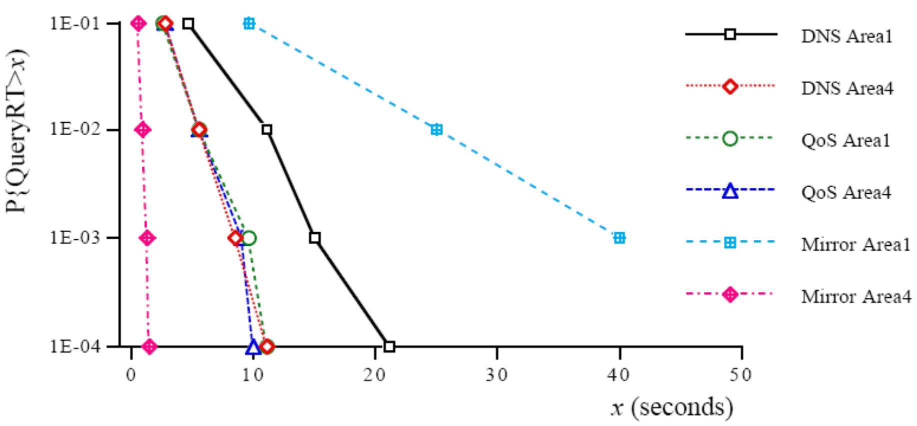


Figure 5: Query response time in the *realistic scenario*

A QoS-based Architecture –w/o DNS modification

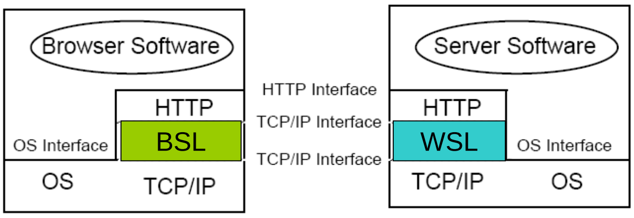
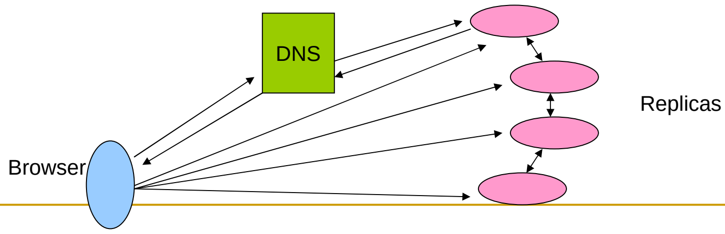


Figure 7: Example of browser and server software structuring.



Conclusion

- From performance comparison of the three load distribution strategies indicated that QoS based strategy outperforms the other two strategies.
- In architectural design, DSN modification needed or facing polling overhead with the alternative configuration.
- URT_{ws} is estimated based on single measurement which may introduce fluctuations of traffic while query among WS replicas.
- Further architectural component – Autonomous Load distribution service responsible for continuous monitoring and providing WSs response time.