Improving the availability of web services

D. Cotroneo, M. Gargiulo, S. Russo, and G. Ventre

University of Naples ‘Federico II’

Consorzi CINI - Laboratorio ITEM
Outline

✓ Rationale
✓ Goals
✓ cosSocket middleware description
✓ Experimental results
✓ Conclusions
In order to maintain the popularity and reputation of a web site, the **Quality of Service (QoS)** perceived by users, especially the **service availability**, is a success factor.

A service that is frequently unavailable may have negative effects on the **reputation** of the service provider, or may result in loss of **opportunities**.
Goals

**Improve the availability of a web service**

**Availability** as...
...the readiness for correct service, which quantifies the alternation between deliveries of **correct** service and **incorrect** service.

But...

? When is the system unable to deliver a correct service?

? What is a correct service?
The service delivered by a system can be defined in terms of a sequence of service items

\[ s_i \quad i = 1,2... \]

each characterized by a tuple

\[ < vs_i, ts_i > \]

where:

- \( vs_i \) is the value of the service item \( s_i \)
- \( ts_i \) is the time interval or instant of observation of service \( s_i \)
Service availability (2)

A service item $s_i$ is defined correct iff:

$$(v_{s_i} \in SV_i) \text{ and } (t_{s_i} \in ST_i)$$

where $SV_i$ and $ST_i$ (functions of the inputs to the system) are respectively the specified sets of correct values and times for service item $s_i$.

In the context of modern Web services is this definition still suitable?
Web-service availability

Considering modern Web services implemented over a QoS-enabled infrastructure...

- ...different users want to get services with different quality at different prices (specified in SLA)

- ...to a special group (premium users) the system has to guarantee the service in a certain period of time
Web-service availability (2)

Web-Service item $s_i$ is defined correct iff:

$$( v_{s_i} \in SV^*_{i,j} ) \text{ and } ( t_{s_i} \in ST^*_{i,j} )$$

where:

- $SV^*_{i,j} = f( SV_i, CU_j )$
- $ST^*_{i,j} = f( ST_i, CU_j )$

$CU_j$ is the Class of Users $j$
Failure mode assumptions

- A system cannot deliver a correct service due to the following reasons:
  - the presence of faults, which cause a system error
  - the presence of overloading condition, i.e. the server is too busy to deliver a correct service

- We investigate only timing failures caused by server overloads conditions

- Effects of value errors are not considered afterwards
Proposal

In order to improve the availability of a web-service we propose an...

...operating system extension for QoS differentiation among classes of communication-bound processes.

Such a middleware allows to define classes of services with different quality attributes concerning the network data delivery behavior.
Middleware description

We used a Real Time Operating System (Linux RT by TimeSys)
Class Service model

We defined a class service model which consists of two kinds of service classes:

- **Adaptive Class**
  - without Admission Control
  - resource sharing in a weighted way

- **Guaranteed Class**
  - with Admission Control
  - resource assignment with self-regulating control loop
Class of Users

Three Classes of Users:

- Normal (Cn)
- Medium (Cm)
- Premium (Cp)

Adaptative CoS

Guaranteed CoS
Timing Errors Assertions

- **No timing errors (τ\textsubscript{none})**
  - Service delivered in $ST^*_{i,j}$

- **Omission errors (τ\textsubscript{O})**
  - Fail silent behavior

- **Late timing errors (τ\textsubscript{L})**
  - Service delivered after a MaxTime threshold depending on user class
  - Possible only for Medium and Premium users
Testbed description

HTTP server

HTTP client

HTTP clients

Workload generators
Experimental results

Service Time Ratio

Ratio (effective ST / expected ST)

Workload (# Normal Clients - Cn)

τ₀

τ₁

ICSE-WADS '02
Conclusions

- Our strategy relies on the capability of controlling the number of system calls issued for requesting an I/O task.
- The middleware effectively improves the availability of the delivered service for some Classes of Users.
- We are currently:
  - Refining measurements in order to demonstrate that proposed middleware effectively improves service availability in wide area scenario.
  - Investigating about a new redundant model which is capable of preventing the system from external faults as well.
Thanks for your attention

Questions

Contact info: mgargiul@unina.it