VM-Rejuvenation:
A cheap and easy self-rejuvenation technique for legacy application servers

Framework developed by:
- Univ. Coimbra (UCO)
- Technical University of Catalonia (UPC)
- Barcelona Supercomputing Center (BSC)
The downtime cost is inadmissible

<table>
<thead>
<tr>
<th>INDUSTRY</th>
<th>AVERAGE DOWNTIME COST PER HOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brokerage services</td>
<td>$6.48 million</td>
</tr>
<tr>
<td>Energy</td>
<td>$2.8 million</td>
</tr>
<tr>
<td>Credit card</td>
<td>$2.58 million</td>
</tr>
<tr>
<td>Telecomm</td>
<td>$2 million</td>
</tr>
<tr>
<td>Financial</td>
<td>$1.5 million</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>$1.6 million</td>
</tr>
<tr>
<td>Financial institutions</td>
<td>$1.4 million</td>
</tr>
<tr>
<td>Retail</td>
<td>$1.1 million</td>
</tr>
<tr>
<td>Pharmaceutical</td>
<td>$1.0 million</td>
</tr>
<tr>
<td>Chemicals</td>
<td>$704,000</td>
</tr>
<tr>
<td>Health care</td>
<td>$636,000</td>
</tr>
<tr>
<td>Media</td>
<td>$340,000</td>
</tr>
<tr>
<td>Airline reservations</td>
<td>$90,000</td>
</tr>
</tbody>
</table>

However, all complex systems are prone to suffer transient failures! …Then crashes…downtime…. 😞

TO AVOID FAILURES…. PREDICT THEM!

How to avoid unplanned downtime?
- Applying self-healing techniques.
This requires:

- Self-monitoring: early detection/prediction of anomalies;
- Take automatic corrective actions to avoid the failures;
Several failures are caused by software aging.

Software Aging: “progressive degradation of the running software that may lead to system crashes or undesirable hang ups”.

This problem has been reported in telecommunications, operating systems, web-servers, enterprise clusters, OLTP systems, spacecraft systems, grid middleware... It may happen in any complex software system.

“Software aging was responsible for the loss of the Patriot anti-missile in the Gulf-war. The solution for the problem was to reboot and restart the Patriot Software every 8 hours”.
Science, page 1347, March 13, 1992
Motivation

Is there a cure for software aging?
- Yes, software rejuvenation.

There are two families of solutions:
- time-based rejuvenation
- Predictive/proactive rejuvenation
A real observation...

- Software rejuvenation is a technique widely used by the industry with 24x7 applications....

- I have seen some highly respectful companies where the Sys Admin guys decide to do some preventive reboots of business-critical applications every day at 4am...

VM-Rejuvenation could be an initiative to start to resolve these situations automatically!! 😊
VM-Rejuvenation Technique

VM-Rejuvenation:
- It is a technique to avoid software aging process.
- It is focus on Web application servers and Web Services.
- VM-Rejuvenation has demonstrate his effectiveness in other environments like Grid Middlewares.
Our list of Guidelines …

- No need to re-engineer the existing web applications, web services or Grid Middlewares;
- Should provide a very fast recovery;
- When doing a “reboot” we cannot lose any on-going request or session-data in our Grid Middleware;
- Make it as-autonomous-as-possible;
- Don’t bring performance overhead to the Grid Middleware;
- No need to buy extra hardware;
- Should be easy to deploy and maintain
VM-Rejuvenation Technique

VM-Rejuv Framework:

- **VM1**: Load Balancer, Aging Detector, Anomaly Detector, Data Collector, Watchdog, SRA Coord
- **VM2**: Active App Server, SRA Agent, S-Probe, Log-Probe, P-Probe
- **VM3**: Standby App Server, SRA Agent, S-Probe, Log-Probe, P-Probe

Only one physical machine,
3 Virtual Machines!!
What is the secret?
Results of VM-Rejvu

Tomcat with TPC-W benchmark

(injecting memory leaks):

![Graph showing performance metrics over time for different conditions: NO REJUV, REJUV (SLA=75), REJUV (SLA=50).](image)
Results of VM-Rejvu

- Tomcat + Axis v1.2/1.3......

what happens with Axis 1.2??

0, 0, 0, 2, 0, 4, 0, 6, 0, 8, 1, 0, 1, 2, 1, 4, 1, 6, 1, 8, 2, 0, 2, 2, 2, 4, 2, 6, 2, 8, 3, 0, 3, 2, 3, 4, 3, 6, 3, 8

0, 0, 0, 2, 0, 4, 0, 6, 0, 8, 1, 0, 1, 2, 1, 4, 1, 6, 1, 8, 2, 0, 2, 2, 2, 4, 2, 6, 2, 8, 3, 0, 3, 2, 3, 4, 3, 6, 3, 8

Requests / sec

Time (hour)

NO REJUV
REJUV (SLA=75)
REJUV (SLA=50)
OGSA-DAI configured to use sessions
What happens with clusters???

Throughput

requests/sec

time (hour)

Default Cluster  Cluster with VM_Rejuvenation

It is useful!!! 😊
And always the client perception is:

What if we want to achieve 99.999% Availability?

- 25 restarts-per-year
- 5 restarts-per-year
- 1 reboot-per-year
Conclusions

- This Rejuvenation scheme can be applied to off-the-shelf application servers;
- It achieves a zero-downtime;
- Does not lose any on-going request at the time of restart;
- The performance penalty is quite acceptable;
- It is useful in single-server and cluster configurations.

AND IT IS CHEAP AND EASY TO INSTALL AND HAS SIMPLE CONFIGURATION
Thank your for your attention.

Developed by:  Computer Architecture Dept., Univ. Politècnica de Catalunya (UPC)  
Dep Eng Informática, University of Coimbra (UCO)

Contacts:  Luis Silva (luis@dei.uc.pt), Jordi Torres (torres@ac.upc.edu),  
Javier Alonso (alonso@ac.upc.edu)