Staged Deployment in Mirage, an Integrated Software Upgrade Testing and Distribution System

Olivier Crameri, Nikola Knezevic, Dejan Kostic, Ricardo Bianchini, Willy Zwaenepoel
Software upgrade deployment

OUTSIDE WORLD

VENDOR
Software upgrade deployment

OUTSIDE WORLD

VENDOR
Software upgrade deployment

OUTSIDE WORLD

β

β

β

β

β

β

β

β

VENDOR

β: Beta-testers
Software upgrade deployment

β: Beta-testers

OUTSIDE WORLD

VENDOR
Software upgrade deployment

\(\beta\): Beta-testers

OUTSIDE WORLD

\(\beta\) \(\checkmark\) \(\beta\) \(\times\) \(\beta\) \(\times\)

\(\beta\) \(\times\) \(\beta\) \(\beta\) \(\beta\)

\(\beta\) \(\beta\) \(\times\)

VENDOR

\(\beta\) \(\beta\)
Software upgrade deployment

\[ \beta: \text{Beta-testers} \]

OUTSIDE WORLD

\[ \beta \]

\[ x \]

\[ \beta \]

\[ x \]

\[ \beta \]

\[ \beta \]

\[ \beta \]

\[ x \]

\[ \beta \]

\[ \beta \]

\[ \beta \]

VENDOR

\[ \text{EPFL} \]

\[ \text{Rutgers} \]
Software upgrade deployment

β: Beta-testers

OUTSIDE WORLD

VENDOR
Staged deployment in Mirage

OUTSIDE WORLD

VENDOR
Staged deployment in Mirage

OUTSIDE WORLD

VENDOR
Staged deployment in Mirage

OUTSIDE WORLD

R: representatives

VENDOR

RUTGERS
Staged deployment in Mirage

OUTSIDE WORLD

VENDOR

R: representatives
Staged deployment in Mirage

OUTSIDE WORLD

VENDOR

R: representatives
Staged deployment in Mirage

OUTSIDE WORLD

R: representatives

VENDOR
Key idea

- Leverage information about the environment:
  - better testing coverage
  - reduced upgrade overhead
Mirage

Upgrade deployment (vendor)

Reporting (repository)

User-machine testing (user)
Mirage

Upgrade deployment (vendor)

Reporting (repository) → User-machine testing (user)
Challenges

• Clustering machines
  ✦ identify the environment
  ✦ fingerprint environmental resources
  ✦ cluster

• Deployment
Goal of clustering

- All machines in a cluster behave identically with respect to an upgrade
Benefit of clustering

- Testing at a representative provides information about an entire cluster
Extent of the benefit

- Depends on:
  - quality of clustering
  - quality of testing
Clustering machines

Identification  Fingerprinting  Clustering

Apache
Clustering machines

Identification

Fingerprinting

Clustering

Apache

Instrumentation

libc-2.4.so

$PATH

httpd.conf

access_log

/tmp/xyz

index.html

Tomcat
Clustering machines

Identification  Fingerprinting  Clustering

Libraries: libc-2.4.so
Environment variables: $PATH
Configuration files: httpd.conf
Log files: access_log
Temporary files: /tmp/xyz
Data files: index.html
Dependent Applications: Tomcat
Clustering machines

Identification  Fingerprinting  Clustering

Libraries:
Environment variables:
Configuration files:
Log files: access_log
Temporary files: /tmp/xyz
Data files: index.html
Dependent Applications:

libc-2.4.so
$PATH
httpd.conf
Tomcat

heuristic + vendor rules
Clustering machines

Identification - Fingerprinting - Clustering

Libraries:

Environment variables:

Configuration files:

Log files: access_log

Temporary files: /tmp/xyz

Data files: index.html

Dependent Applications:

heuristic + vendor rules
Clustering machines

Identification

Fingerprinting

Clustering

- libc-2.4.so
- $PATH
- httpd.conf
- Tomcat

Parsers
Clustering machines

Identification

Fingerprinting

Clustering

$PATH
httpd.conf
Tomcat

Parsers

libc.2.4.hash
Clustering machines

Identification

Fingerprinting

Clustering

Tomcat

Parsers

libc.2.4.hash

$PATH.hash

httpd.conf.listen.hash

httpd.conf.mod_php.hash

httpd.conf.mod_cgi.hash
Clustering machines

Identification

Fingerprinting

Clustering

Parsers

libc.2.4.hash

$PATH.hash

httpd.conf.listen.hash

httpd.conf.mod_php.hash

httpd.conf.mod_cgi.hash

Tomcat.5.5
Clustering machines

Identification

Fingerprinting

Clustering
Clustering machines

Identification

Fingerprinting

Clustering
Clustering machines

Identification

Fingerprinting

Clustering
Challenges

• Clustering machines
  ✦ identify the environment
  ✦ fingerprint environmental resources
  ✦ cluster

• Deployment
Challenges

• Clustering machines
  ✦ identify the environment
  ✦ fingerprint environmental resources
  ✦ cluster

• Deployment
Deployment goals

• Low upgrade overhead
• Fast deployment
Deployment goals

• Low upgrade overhead
• Fast deployment

There is a tradeoff!
### Controlling the tradeoff

- **Leverage clusters for staging**
- **Deploy in parallel or in sequential stages**

<table>
<thead>
<tr>
<th></th>
<th>Parallel</th>
<th>Sequential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of deployment</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Upgrade overhead</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>
Representatives first, in parallel
Representatives first, in parallel

R: representatives

OUTSIDE WORLD

VENDOR
Representatives first, in parallel

OUTSIDE WORLD

R: representatives

Upgrade overhead: 2
Representatives first, in sequence

R: representatives

OUTSIDE WORLD

VENDOR
Representatives first, in sequence

R: representatives

OUTSIDE WORLD

VENDOR
Representatives first, in sequence

OUTSIDE WORLD

VENDOR

R: representatives
Representatives first, in sequence

R: representatives

OUTSIDE WORLD

VENDOR

Upgrade overhead: 1
Representatives and other machines in parallel

\[ \text{OUTSIDE WORLD} \]

\[ \text{VENDOR} \]

\[ \text{R: representatives} \]
Representatives and other machines in parallel

**OUTSIDE WORLD**

<table>
<thead>
<tr>
<th>R</th>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**VENDOR**

<table>
<thead>
<tr>
<th>R</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Representatives and other machines in parallel

OUTSIDE WORLD

R: representatives

VENDOR

Upgrade overhead: 2
Representatives and other machines in parallel

R: representatives

OUTSIDE WORLD

VENDOR
Representatives and other machines in parallel

\( R: \) representatives

OUTSIDE WORLD

VENDOR
Representatives and other machines in parallel

R: representatives

OUTSIDE WORLD

VENDOR

Upgrade overhead: 4
Representatives in parallel, other machines in sequence

OUTSIDE WORLD

R: representatives

VENDOR
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE WORLD

VENDOR
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE WORLD

VENDOR
Representatives in parallel, other machines in sequence

OUTSIDE WORLD

VENDOR

R: representatives
Representatives in parallel, other machines in sequence

OUTSIDE WORLD

R: representatives

VENDOR
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE WORLD

VENDOR
Representatives in parallel, other machines in sequence

OUTSIDE WORLD

R: representatives

VENDOR
Representatives in parallel, other machines in sequence

OUTSIDE WORLD

VENDOR

R: representatives
Representatives in parallel, other machines in sequence

OUTSIDE WORLD

VENDOR

Upgrade overhead: 3

R: representatives
Evaluation

• Quality of clustering:
  ✦ accuracy of identification of env. resources
  ✦ accuracy of the clustering algorithm

• Staged deployment:
  ✦ control upgrade overhead and deployment speed
Can we accurately identify environmental resources?

<table>
<thead>
<tr>
<th>App.</th>
<th># env. resources</th>
<th># vendor rules</th>
<th>#errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>firefox</td>
<td>839</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>apache</td>
<td>251</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>php</td>
<td>206</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mysql</td>
<td>250</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Can we accurately identify environmental resources?

<table>
<thead>
<tr>
<th>App.</th>
<th># env. resources</th>
<th># vendor rules</th>
<th>#errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>firefox</td>
<td>839</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>apache</td>
<td>251</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>php</td>
<td>206</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mysql</td>
<td>250</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Example: Apache

- Files initially misclassified:
  - `index.html`
  - `access_log`

- Vendor rules needed:
  - `exclude *.html`
  - `exclude access_log`
Can we accurately cluster machines?

- 21 MySQL environments
  - 2 distributions of Linux
  - PHP and Apache
  - Various MySQL configurations
- 2 real upgrade problems
Can we accurately cluster machines?

• 0 misplaced machines
  ✦ optimal for reducing upgrade overhead by testing at representatives

• 15 clusters
  ✦ artefact of the experiment
  ✦ number of clusters can be reduced by increasing the fingerprinting granularity
Controlling the tradeoff

• Experimental setup:
  ✦ Event-driven simulator
  ✦ 100’000 machines
  ✦ 3 problems
  ✦ 2 staging protocols, plus NoStaging
Controlling the tradeoff

- Upgrade overhead reduced very significantly
- Deployment completes 25% later than NoStaging in the worst case
Related work

- Package management systems
- Patch management systems
- Strider, Microsoft ACT
- No other work considered clustering and staged deployment
Conclusion

- Staged deployment in Mirage
  - machines can be clustered based on their environment
  - significantly reduce upgrade overhead
Thank you for your attention!

olivier.crameri@epfl.ch

Travel scholarship provided by: Infosys®