Predicting the Performance of Virtual Machine Migration

Sherif Akoush, Ripduman Sohan, Andrew Rice, Andrew W Moore and Andy Hopper

17-Aug-2010
Live migration moves a running virtual guest to a new physical host

http://picasaweb.google.com/lh/photo/uAPf5hbH9pHE_L2KvjcGpw
Downtime is the period of time for which the domain is stopped.

Migration time is total duration of the movement process.
The migration algorithm causes times to vary with workload.

- **Initialisation**
- **Pre-copy**
- **Stop and copy**
- **Finalisation**
The migration algorithm causes times to vary with workload

Initialisation

Pre-copy

Stop and copy

Finalisation

Downtime

Total migration time
Administrators need advance knowledge of **downtime**

How frequently can I migrate this domain without violating the Service Level Agreement?
Administrators need advance knowledge of **migration time**

Is it worth consolidating workloads and switching off hardware?
Heuristic stop conditions are used to terminate the pre-copy stage

- Initialisation
- Pre-copy
- Stop and copy
- Finalisation

**Stop if:**

- Fewer than 50 pages dirtied during last iteration
- More than 3 times total RAM has been copied
- 29 iterations have been carried out
Key parameter 1: **page dirty rate** has a non-linear effect

Downtime

Migration time

Page dirty microbenchmark, 1Gbps link

314ms
Key parameter 2: **Link bandwidth has a non-linear effect**

**Downtime**

![Graph showing downtime for 1 Gbps and 10 Gbps link bandwidths.]

**Migration time**

![Graph showing migration time for 1 Gbps and 10 Gbps link bandwidths.]

*Note: The graphs illustrate the relationship between page dirty rate and downtime/migration time for different link bandwidths.*
AVG model suitable for guests with a constant page dirty rate
HIST model for guests with cyclic behaviour

1) Keep a recorded history of page dirtying
2) Simulate the Xen migration process

<table>
<thead>
<tr>
<th></th>
<th>MT_A</th>
<th>MT_P</th>
<th>Err</th>
<th>DT_A</th>
<th>DT_P</th>
<th>Err</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>5.8 s</td>
<td>5.7 s</td>
<td>2.4%</td>
<td>317.3 ms</td>
<td>314.1 ms</td>
<td>2.4%</td>
</tr>
<tr>
<td>WEB</td>
<td>7.5 s</td>
<td>7.4 s</td>
<td>2.0%</td>
<td>449.5 ms</td>
<td>420.4 ms</td>
<td>6.4%</td>
</tr>
<tr>
<td>SFS</td>
<td>14.8 s</td>
<td>14.9 s</td>
<td>1.5%</td>
<td>217.6 ms</td>
<td>217.7 ms</td>
<td>0.1%</td>
</tr>
<tr>
<td>MR</td>
<td>14.9 s</td>
<td>15.13 s</td>
<td>1.4%</td>
<td>348.9 ms</td>
<td>348.1 ms</td>
<td>0.2%</td>
</tr>
</tbody>
</table>
Future work

How can we change the stopping condition heuristics to provide some guarantees about migration time?

We've so far only considered migration of RAM. What happens when we need to move storage too?

What is the most effective means for migrating a set of VMs simultaneously?
Conclusion

It is important to know the expected interruption due to migration in advance.

It's possible to predict migration times and service interruption.

Thanks to: Sherif Akoush, Ripduman Sohan, Andrew Moore, Andy Hopper, Kieran Mansley.

http://www.cl.cam.ac.uk/research/dtg/planet