Container-Virtualization within the Space Industry

Jens Pfau, Pavol Safarik & Sascha Schünemann
Outline

• Containers vs. Virtual Machines
  • Efficiency
  • Portability
  • Scalability
• Tool Support
• Benefits of using container technology
  • In development
  • In production
Containers vs. Virtual Machines

**Virtual Machine**

- App 1
  - Bins/Libs
  - Guest OS
- Hypervisor
- Host Operating System
- Infrastructure

**Container**

- App 1
  - Bins/Libs
  - Guest OS
- Docker Engine
- Operating System
- Infrastructure

**Benefits**

- Efficient resource usage
- Portable
- Simplified automated scaling

Source: https://www.docker.com/what-docker/
What can run inside a container?

- Originally only Linux applications,
- But others are catching up (e.g. Microsoft)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Repository</th>
<th>Stars</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ubuntu</td>
<td>2,007</td>
</tr>
<tr>
<td>2</td>
<td>centos</td>
<td>1,164</td>
</tr>
<tr>
<td>3</td>
<td>nginx</td>
<td>1,163</td>
</tr>
<tr>
<td>4</td>
<td>redis</td>
<td>957</td>
</tr>
<tr>
<td>5</td>
<td>node</td>
<td>891</td>
</tr>
<tr>
<td>6</td>
<td>postgres</td>
<td>889</td>
</tr>
<tr>
<td>7</td>
<td>mysql</td>
<td>885</td>
</tr>
<tr>
<td>8</td>
<td>mongo</td>
<td>796</td>
</tr>
<tr>
<td>9</td>
<td>debian</td>
<td>573</td>
</tr>
<tr>
<td>10</td>
<td>jenkins</td>
<td>508</td>
</tr>
</tbody>
</table>
Efficiency

Memory / CPU Consumption

All containers on a host share the same kernel.

- Little overhead in terms of memory and CPU consumption.
- Containers do not allocate more memory or CPU than required.
- Fast starting of containers (almost instantaneous).

Disk Usage

Original App
App A
Bins/Libs

App B
App B using same binaries and libraries

Modified App A'

Source: https://www.docker.com/whatisdocker/
Portability

Off-premise / public cloud

- Windows Azure
- Google Cloud Platform
- Amazon Web Services

On-premise / private cloud

- openstack
- VMware
- Microsoft Hyper-V

Bare metal
Scaling using containers

Source: http://martinfowler.com/articles/microservices.html
Tool Support

- Container Operating Systems
- Image Registries
- Scheduling & Orchestration
- Security
- Developer Tools
- Monitoring & Logging
- Software Defined Networking

Container Ecosystem
Container Usage in Development

IDPF

Processing infrastructure for EUMETSAT's Meteosat Third Generation Ground Segment Level 1 data facility

- Receives high volume raw instrument data
- Performs spectral, radiometric and geometric calibrations and corrections.
- Scheduling over a cluster of nodes
Container Usage in Development

Single Docker host running
- IDPF server container
- Multiple IDPF processing containers

Docker host can be
- Developer workstation
- Virtual machine
Benefits of Containers in Development Phase

Portability:
• To run the test system, only a Docker engine is needed

Scalability:
• Due to autoscaling, only necessary amount of resources is used
• Inactive nodes are automatically removed freeing memory, CPU, and storage resources

Efficiency:
• Reduced disk, memory, and CPU consumption during testing
• Increased automated test coverage by allowing usage of distributed test scenarios during nightly automated testing
Benefits of Containers in Production

Commit
Static Code Analysis
Code Review
Merge & Build & Container Build
Unit Testing
Deployment
Integration Testing
Delivery

Development

Customer

Server
Replace container
Summary

• Containers are efficient, portable, and scalable.
  • Reduced operating costs
  • Increased flexibility
• Container technology complements virtualization.
• Particularly suited for container deployment:
  • Systems with loose coupling between components.
  • Systems that require the scheduling and scaling of tasks.
Thank you.

Jens Pfau  
System Architect  
CGI Deutschland Ltd & Co KG  
Tel: +49 6151 36860 191  
jens.pfau@cgi.com

Pavol Safarik  
Software Engineer  
CGI Deutschland Ltd & Co KG  
Tel: +49 6151 36860 197  
pavol.safarik@cgi.com

Sascha Schuenemann  
Associate Consultant  
CGI Deutschland Ltd & Co KG  
Tel: +49 6151 36860 208  
sascha.schuenemann@cgi.com