The path to data plane microservices

Ray Kinsella

DPDK Summit USA - San Jose - 2017
Legal Disclaimers

- Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.
- No computer system can be absolutely secure.
- Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.
- Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.
- All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.
- Intel, the Intel logo, and other Intel product and solution names in this presentation are trademarks of Intel.
- *Other names and brands may be claimed as the property of others.
- © 2017 Intel Corporation.
The 12 Factor APP

<table>
<thead>
<tr>
<th>12 factors (solid principle for Cloud Software Architecture)</th>
<th>Codebase</th>
<th>Dependencies</th>
<th>Config</th>
<th>Backing Services</th>
<th>Build, release, run</th>
<th>Processes</th>
<th>Port binding</th>
<th>Concurrency</th>
<th>Disposability</th>
<th>Dev/prod parity</th>
<th>Logs</th>
<th>Admin processes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One codebase tracked in revision control, many deploys</td>
<td>Explicitly declare and isolate dependencies</td>
<td>Store configuration in the environment</td>
<td>Treat backing services as attached resources</td>
<td>Strictly separate build and run stages</td>
<td>Execute the app as one or more stateless processes</td>
<td>Export services via port binding</td>
<td>Scale out via the process model</td>
<td>Maximize robustness with fast startup and graceful shutdown</td>
<td>Keep development, staging, and production as similar as possible</td>
<td>Treat logs as event streams</td>
<td>Run admin/management tasks as one-off processes</td>
</tr>
</tbody>
</table>

Priority Principles

https://12factor.net/
Adam Wiggins 2017
Microservices environment

Microservices Enabling

Container Enabling

Resource sharing API

Data plane Microservices

CPU Sharing

I/O Sharing

Memory Sharing

Consistent APIs across deployment models
Monolithic

Dynamic Allocation

4K page allocation

Also
- CRIU - check-point and restore in userspace
- State synchronization ....
Scalable I/O for decomposition

Monolithic

Virtual Switch

HW Accelerated
Lightweight threading models

Monolithic

Polling

Polling

In-Process Scheduler

DPDK Scheduler

Core

Core

Core

Core

Multi-Process Scheduler

Linux Scheduler

DPDK Scheduler

Core

Core

Core

Core

uService1

uService2

uService3

uService4

uService5

(rx/tx)

Logging

uService6

DPDK

DATA PLANE DEVELOPMENT KIT

Polling

DPDK

DATA PLANE DEVELOPMENT KIT

New instance

New instance

uService2

uService2
# Data plane microservice models

<table>
<thead>
<tr>
<th>Model</th>
<th>In-process Microservices</th>
<th>Multi-process Microservices</th>
<th>Multi-node Microservices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why?</td>
<td>Highest Performance</td>
<td>Multi-process scaling</td>
<td>Multi-node scaling</td>
</tr>
<tr>
<td>Scheduling</td>
<td>DPDK Scheduler</td>
<td>Cooperative OS</td>
<td>Cooperative OS</td>
</tr>
<tr>
<td>Memory</td>
<td>Monolithic</td>
<td>Dynamic</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Transport</td>
<td>Mem Ring</td>
<td>Mem Ring, vSwitch, HW</td>
<td>RoCE, RDMA, TCP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accelerated</td>
<td></td>
</tr>
<tr>
<td>Failure Protection</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Live Migration</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Data plane microservice evolution
Questions?

Ray Kinsella
ray.kinsella@intel.com
irc: mortderire