



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



12 factors
(solid
principle for
Cloud
Software
Architecture)

Codebase

One codebase tracked in revision control, many deploys

Dependencies

Explicitly declare and isolate dependencies

Config

Store configuration in the environment

Backing Services

Treat backing services as attached resources

Build, release, run

Strictly separate build and run stages

Processes

Execute the app as one or more stateless processes

Port binding

Export services via port binding

Concurrency

Scale out via the process model

Disposability

Maximize robustness with fast startup and graceful shutdown

Dev/prod parity

Keep development, staging, and production as similar as possible

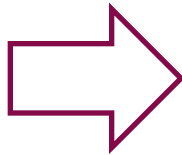
Logs

Treat logs as event streams

Admin processes

Run admin/management tasks as one-off processes

Priority
Principles



<https://12factor.net/>
Adam Wiggins 2017

Microservices environment



Microservices
Enabling

Data plane Microservices

Resource sharing API

Container
Enabling



CPU Sharing



I/O Sharing



Memory Sharing

Consistent APIs across deployment models

Lightweight memory model



2MB

2MB

2MB

2MB

2MB

2MB

Monolithic

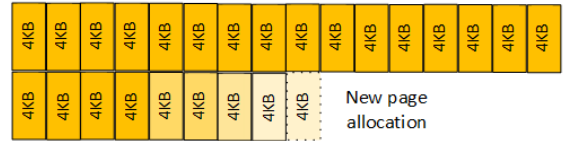


2MB

2MB

New page allocation

Dynamic Allocation

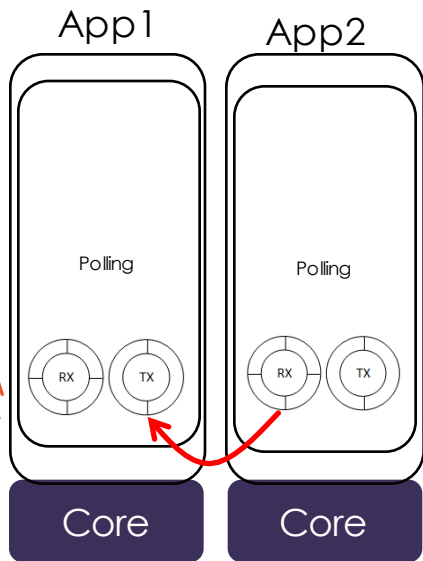


4K page allocation

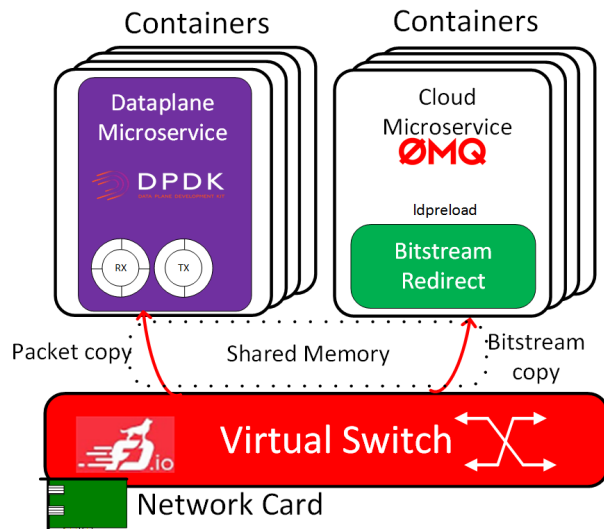
Also

- CRIU - check-point and restore in userspace
- State synchronization

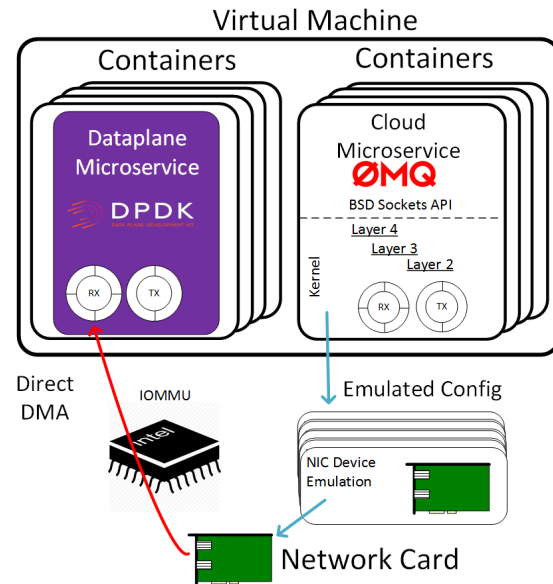
Scalable I/O for decomposition



Monolithic

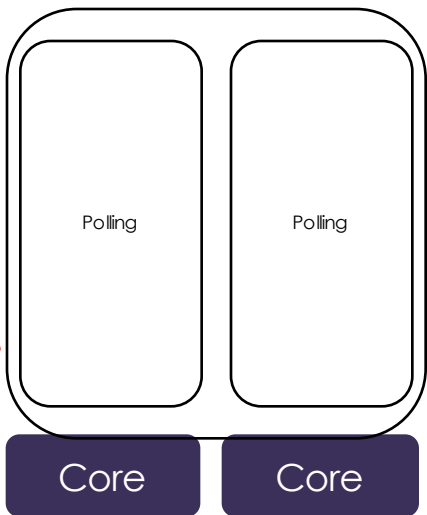


Virtual Switch

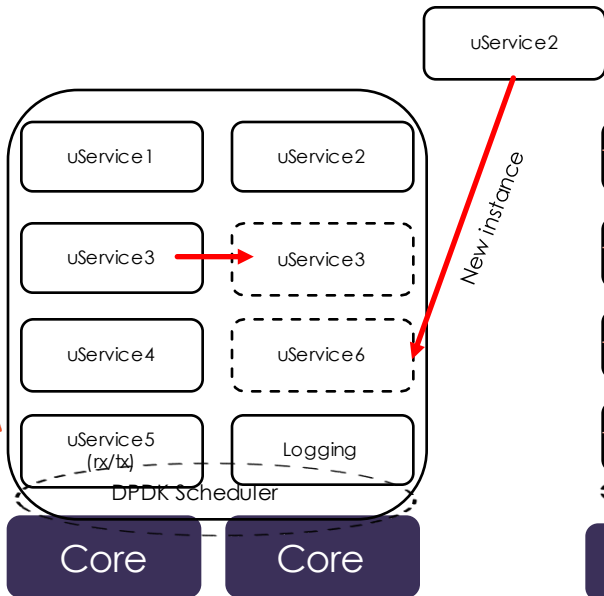


HW Accelerated

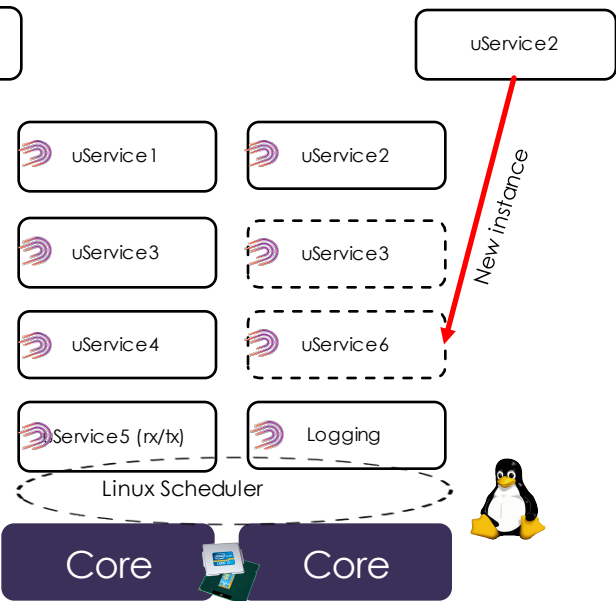
Lightweight threading models



Monolithic



In-Process Scheduler



Multi-Process Scheduler

Data plane microservice models



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

Data plane microservice evolution



Questions?

Ray Kinsella

ray.kinsella@intel.com

irc: mortderire