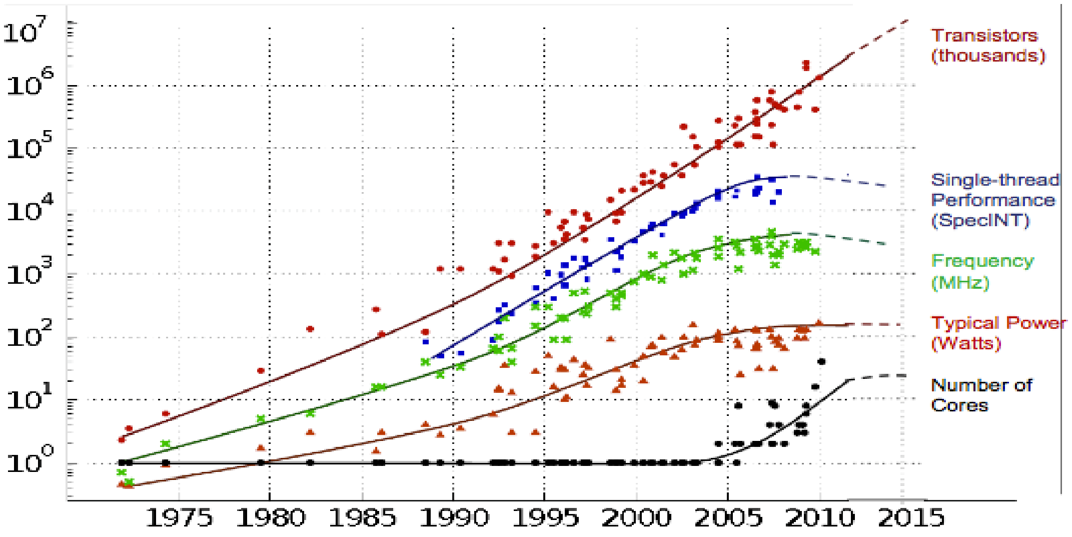
Next Generation Operating Systems

Zeljko Susnjar, Cisco CTG

June 2015

The end of CPU scaling

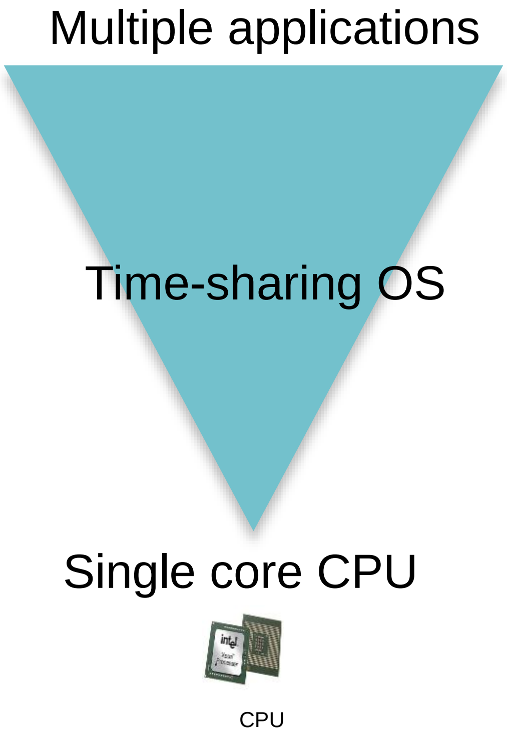


Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

Future computing challenges

- Power efficiency
- Performance == parallelism

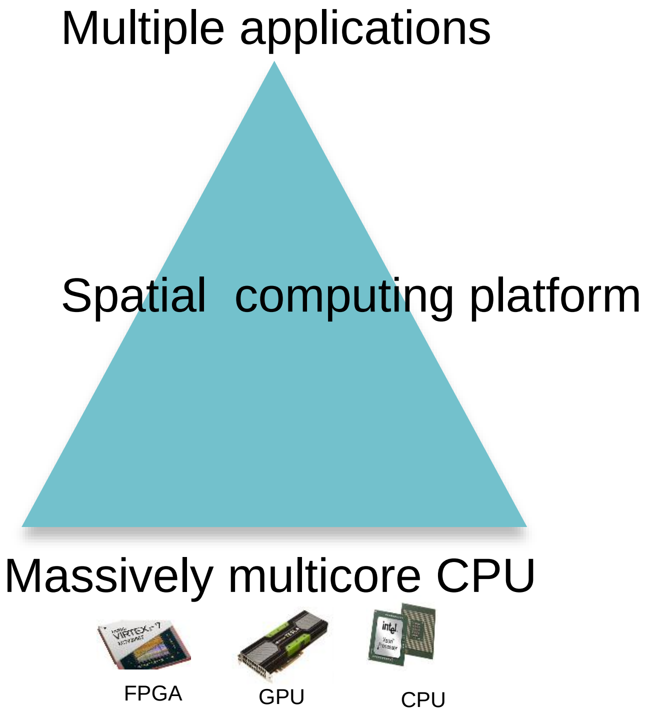
Paradox of the computing industry
System software has not evolved at the same pace as HW



Multiple applications

Time-sharing OS

Single core CPU

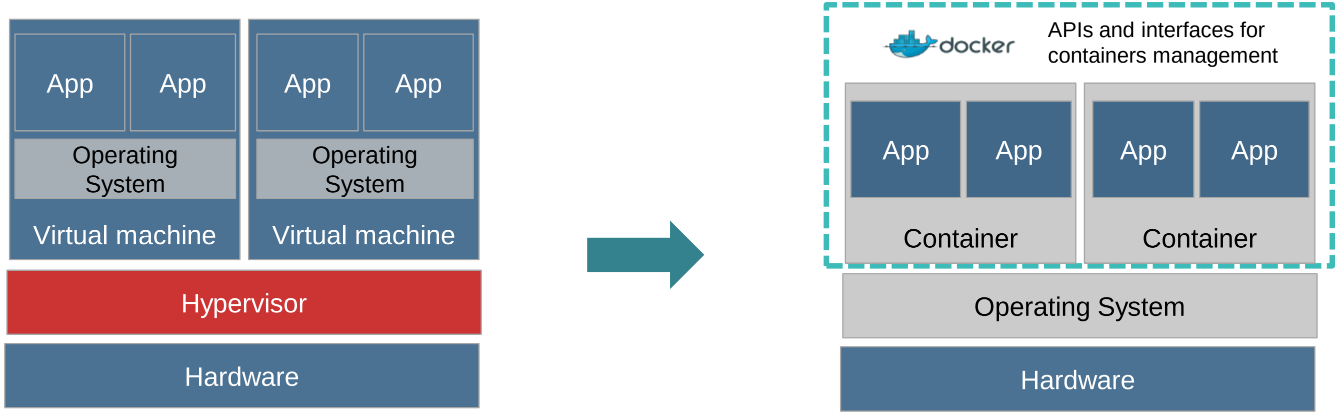


Multiple applications

Spatial computing platform

Massively multicore CPU

Server Virtualization is at a generational shift



App App App App

Operating System Operating System

Virtual machine Virtual machine

Hypervisor

Hardware

docker

APIs and interfaces for containers management

App App App App

Container Container

Operating System

Hardware

Hypervisors are still good, but have pitfalls

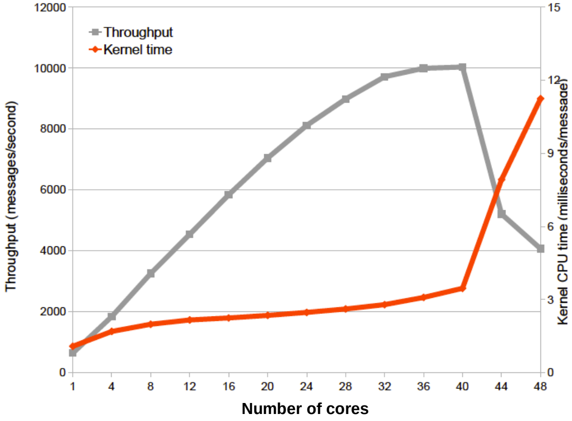
- ⊕ Flexible, multi-OS, application isolation / security
- ⊖ Optimizations, IO handling, expensive license fees

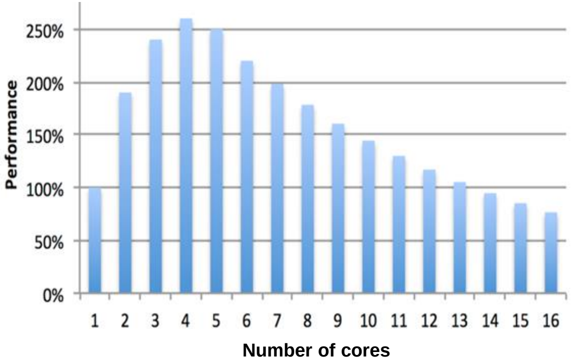
Application containers are the future of virtualization

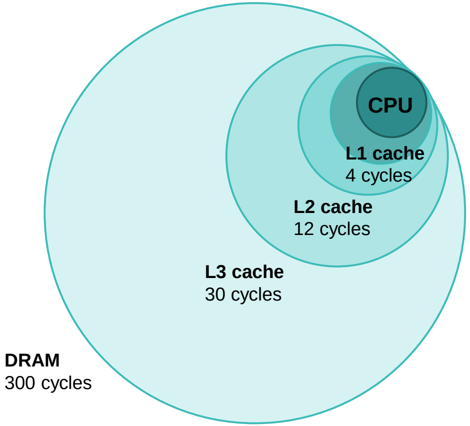
- ⊕ No hypervisor overhead, performance, fine grained resource control
- ⊖ Linux ABI, security and app isolation

OS constraints are hard to overcome with current design

Containers are built on Operating system







CPU

L1 cache
4 cycles

L2 cache
12 cycles

L3 cache
30 cycles

DRAM
300 cycles

Packet scalability

- Kernel stack too complicated
- User mode networking stack like PF_RING/DPDK write your own driver

CPU Cores scalability

- Many task on one core → one task on many cores
- Monolithic architecture based on time sharing

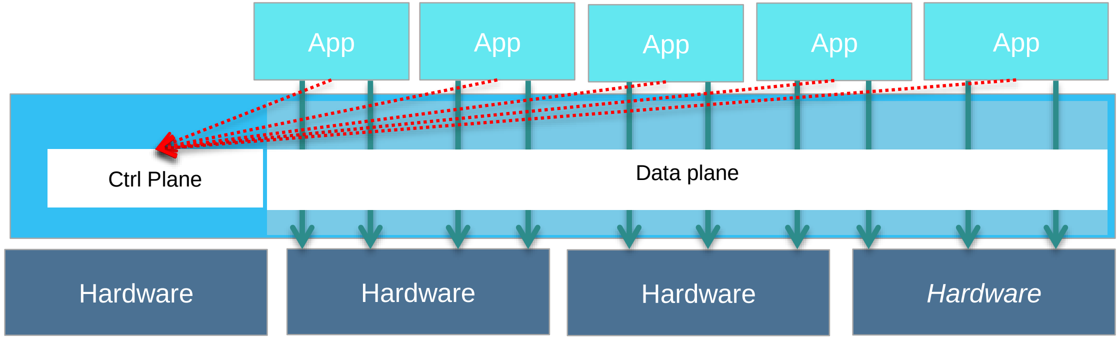
Memory scalability

- CPU cache too small
- Cache misses due to scattered data in the memory

Solving the problem at the lowest level of abstraction

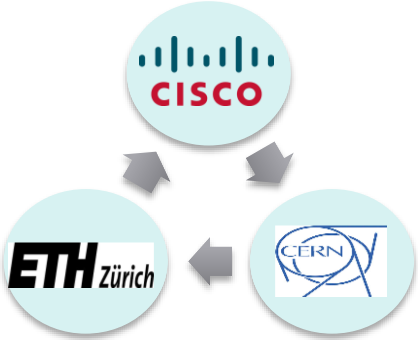
Next-Gen Computing: Redesign of Operating System for energy efficient and linear scalable computing platform
New architectural concept, greatly enhancing application performance in modern datacenters and fundamentally addressing the following challenges:

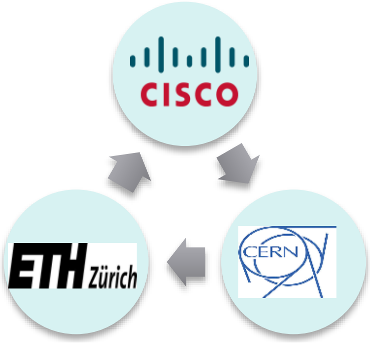
- Application's parallelization driven by rapidly growing number of CPU cores per socket
- Efficient use of heterogeneous resources
- Data center wide system consistency



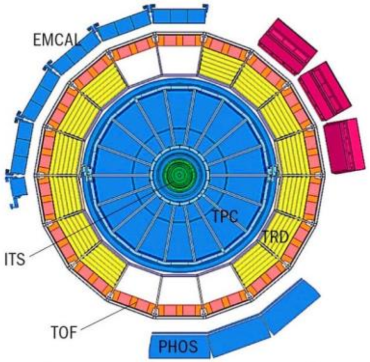
Advantages of OS control/data plane separation

- Greater scale-up bundled with smarter NIC processing
- Reduce CPU kernel overhead
- Higher network throughput by using multiple cores

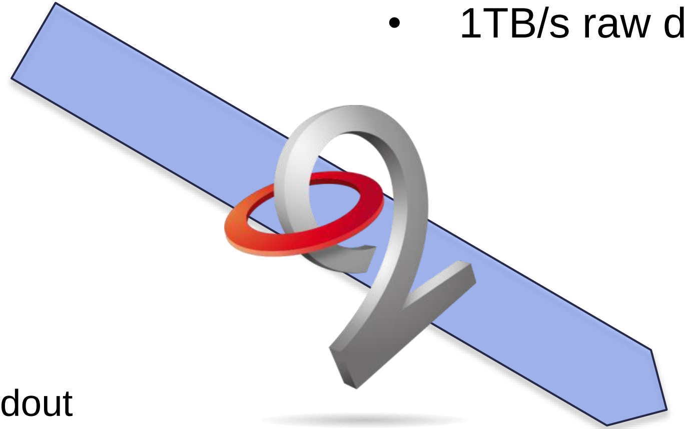



CISCO
CERN
ETH Zürich

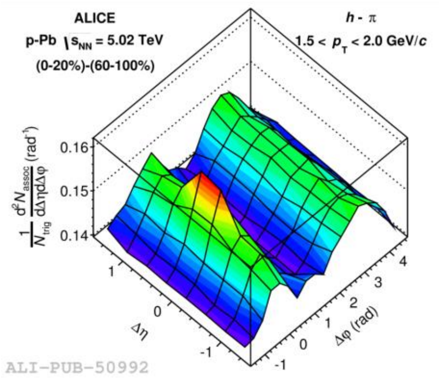
CERN Challenge: The Alice computing requirements



- Detector upgrade for Run 3 (2020)
- 100 increase in event rate
- 1TB/s raw data rate



- From Detector Readout to Analysis:
- What is the “optimal” computing architecture?



O2 facility:
Highly specialized heterogeneous computing platform

+ 463 FPGAs
  - Detector readout and fast cluster finder
+ 100’000 CPU cores
  - To compress 1.1 TB/s data stream by overall factor 14
+ 5000 GPUs
  - To speed up the reconstruction
+ 50 PB of disk

= Considerable computing capacity that will be used for Online and Offline tasks

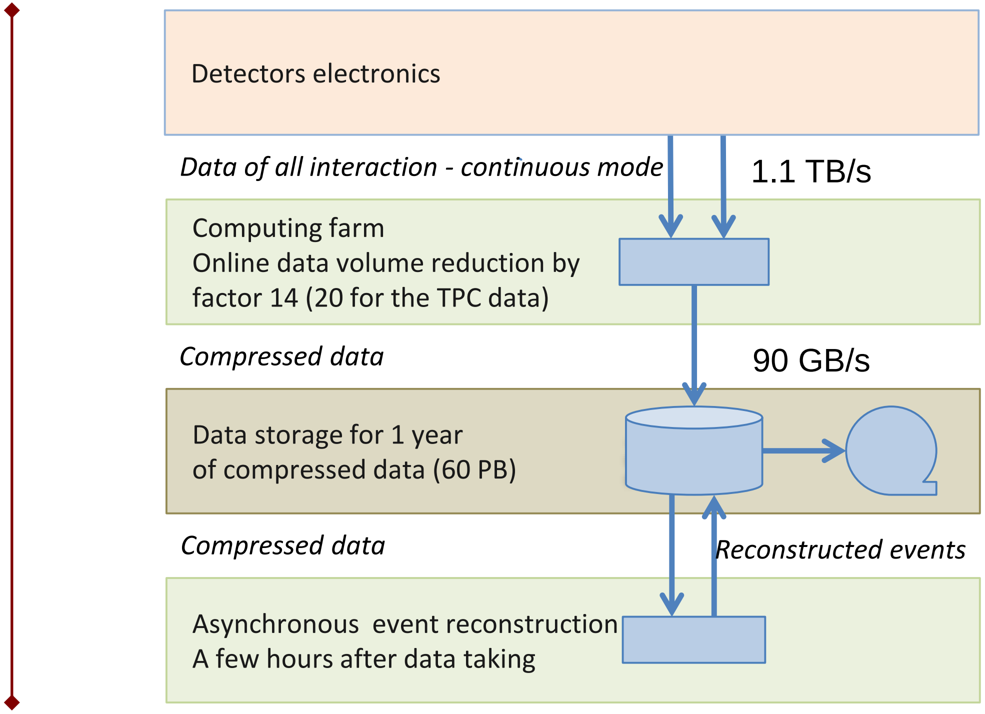
Data Plane Computing System

DPCS: Openlab project investigating applicability of modern OS concepts in the ALICE O2 environment

- Data Plane OS concept
- I/O Virtualization
- Multicore scaling
- Heterogeneous compute

Thank you.

Data flow in O2 facility



Detectors electronics

Data of all interaction - continuous mode

1.1 TB/s

Computing farm
Online data volume reduction by factor 14 (20 for the TPC data)

Compressed data

90 GB/s

Data storage for 1 year
of compressed data (60 PB)

Compressed data

Reconstructed events

Asynchronous event reconstruction
A few hours after data taking

~ 8000 optical links

- Read-out farm: 250 servers with FPGA acceleration
- Processing farm: 1500 servers with GPU acceleration
- Storage system
  68 storage units
  with 34 data servers