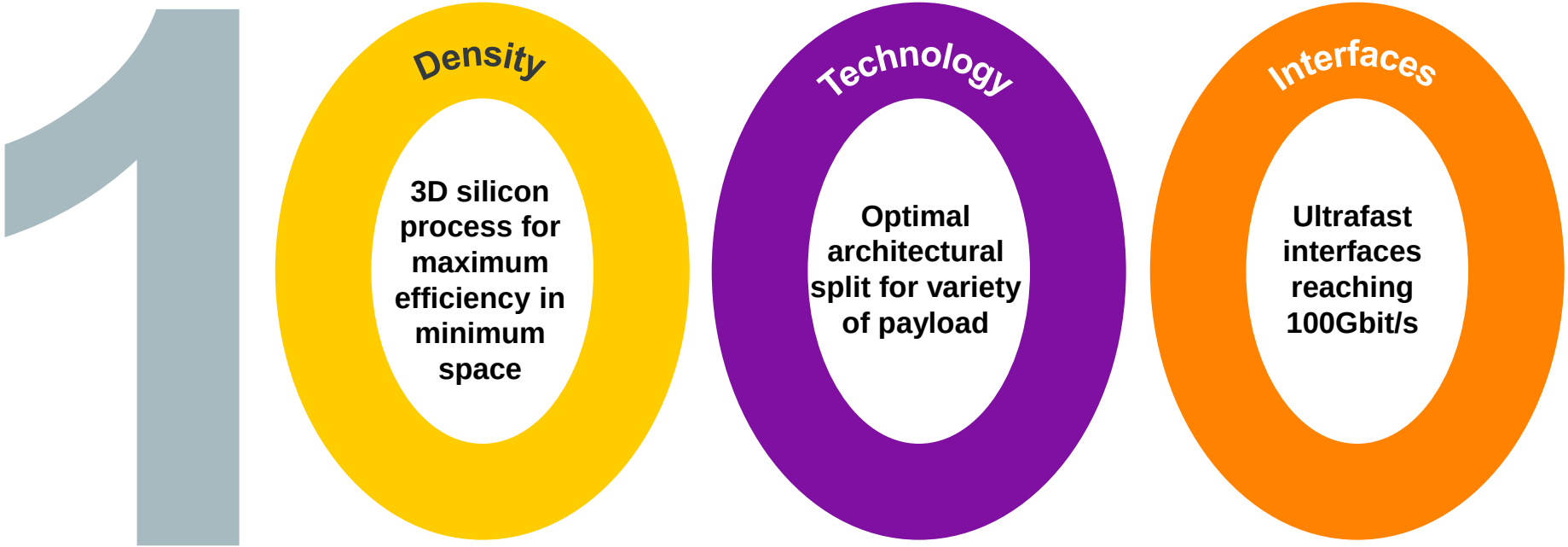
High end telecom networking – deep dive

LCU 2013

1000x Packet Traffic Technologies are Maturing

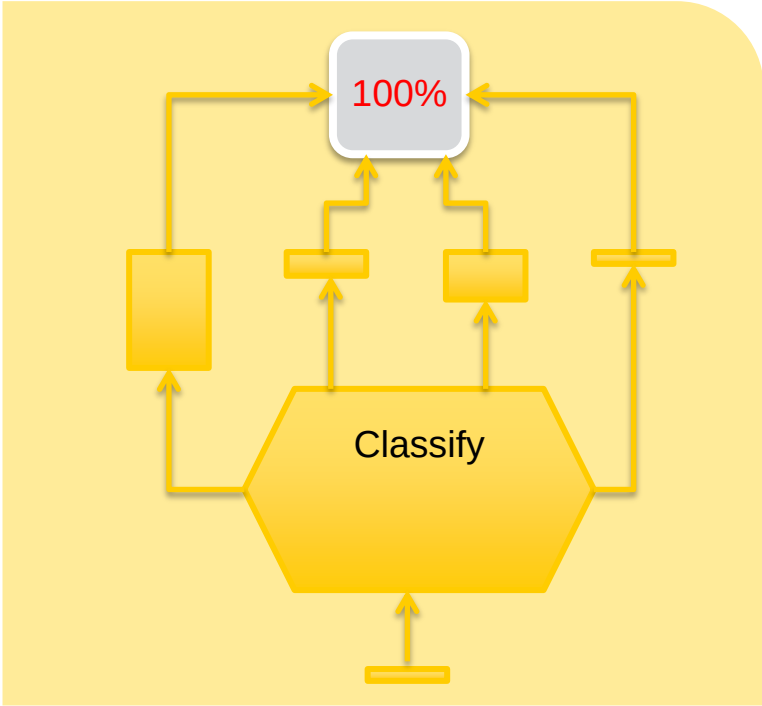


… times more capacity

Single core networking

Only priority queuing doable with single core

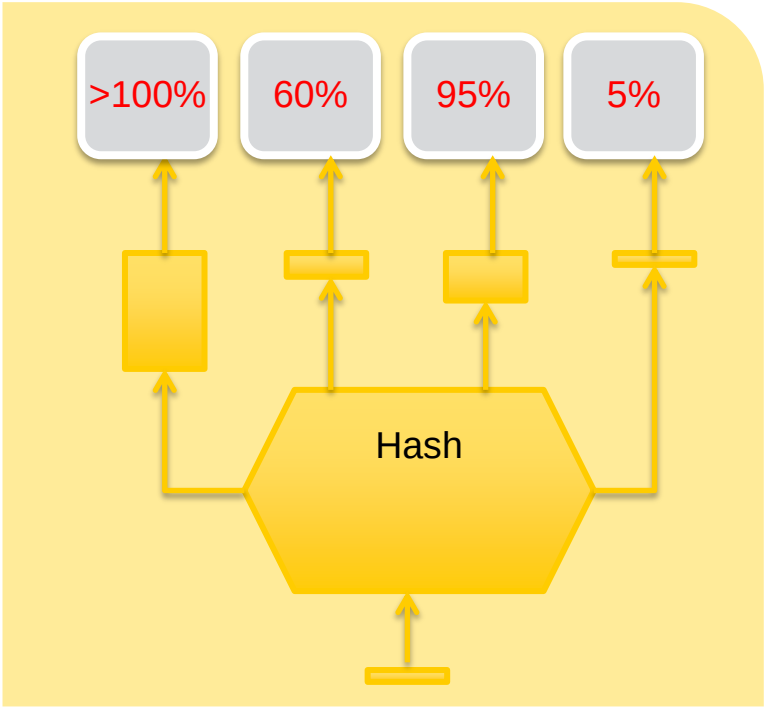
- Static priorities
- Highest priority first with starving prevention logic
- Packet order not an issue
- Does not scale



Multicore SoC Load Sharing

Multicore SoC great for load sharing

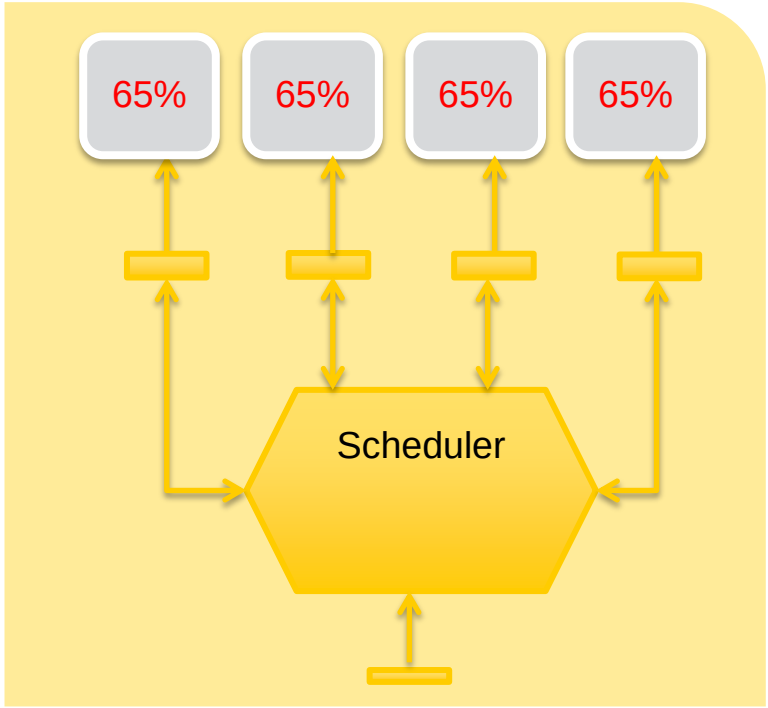
- Static resource allocation
- Queue selection typically by hash of incoming packet header or round robin
- HW aware applications
- Unpredictable, uneven core load



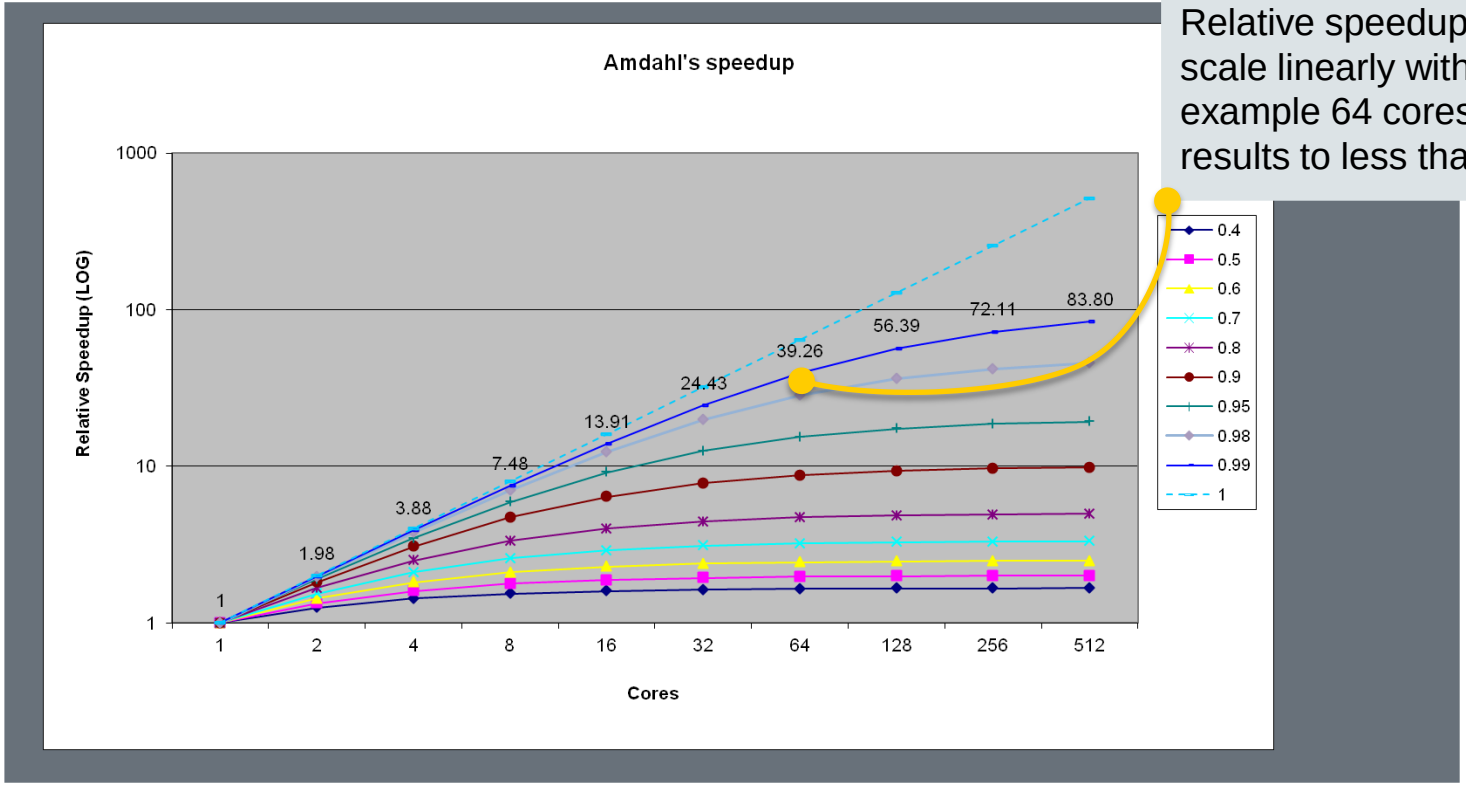
Multicore SoC Load Balancing

Multicore SoC great for load balancing

- Dynamic resource allocation with rules
- Fast per packet decision
- Event Machine model: each thread asks for new job after current is finished
- Automatic scaling – “single thread” programming model



Hickups with multicore scaling



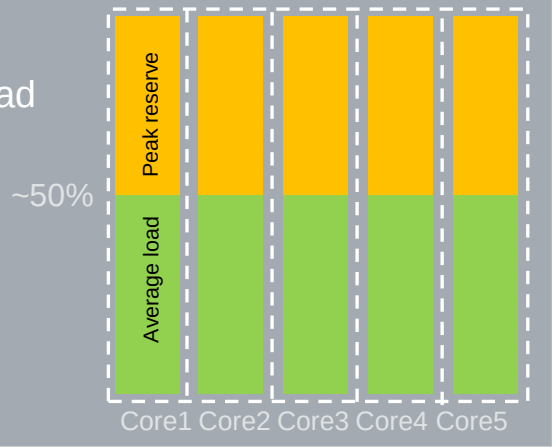
Relative speedup does not scale linearly with cores, for example 64 cores, 1% serial results to less than 40x speedup

Case study: WCDMA HSPA

- Mobile traffic for large amount of cells
- Find max throughput - QoS threshold for adding more users

Static core allocation

- Users allocated per core based on resource availability to maintain enough peak reserves
- High headroom required
- Low average load

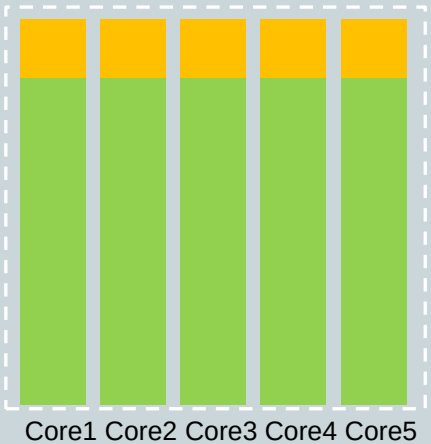


Dynamic core allocation

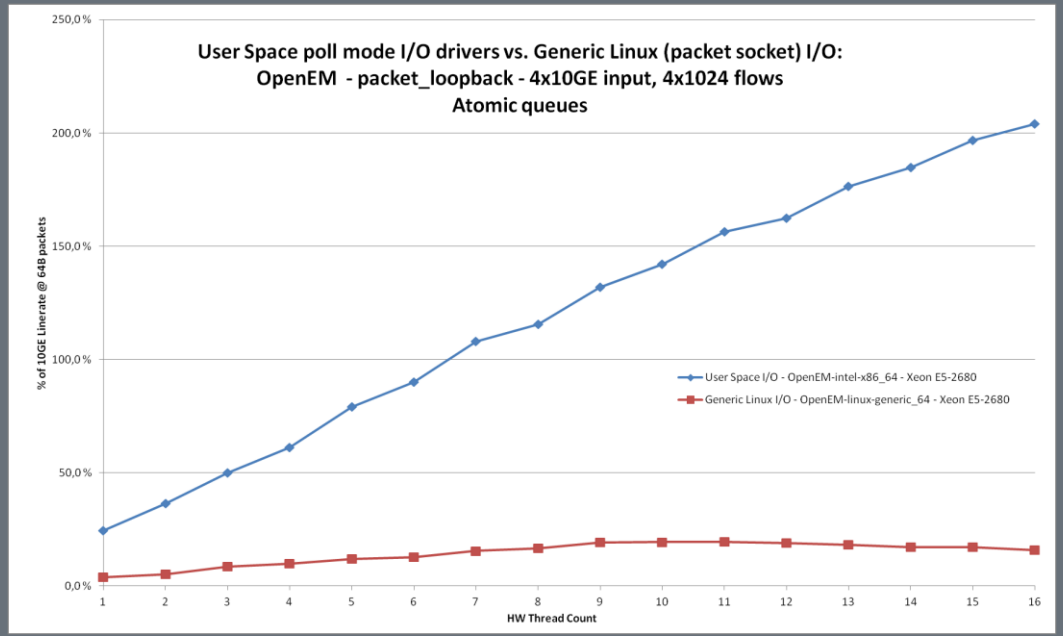
- Allocation of users to a resource pool of cores
- Statistical multiplexing gain
- Low headroom
- High average load

~90%

80% gain



Case study: Linux userspace PMD vs. socket packet test

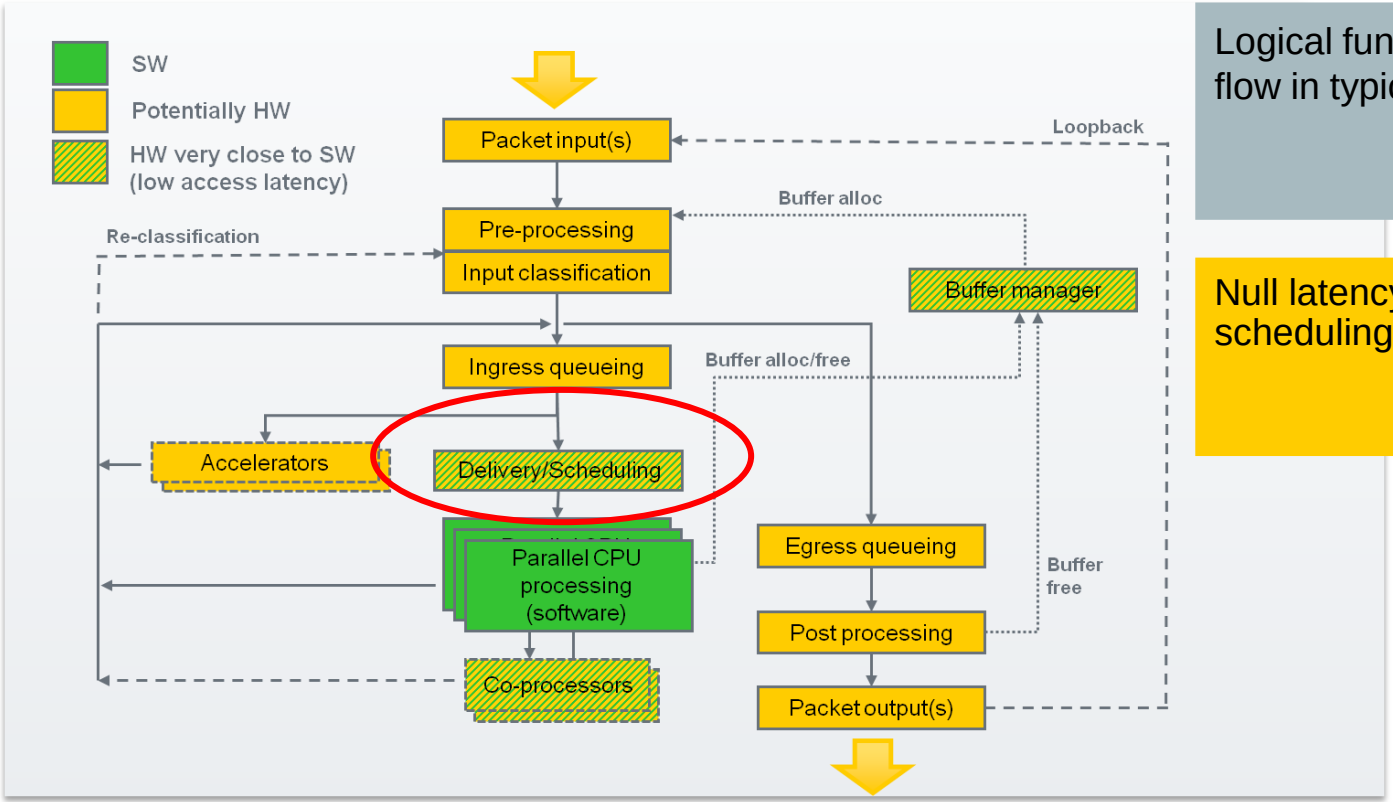


Kernel approach saturates at ten cores

Dataplane in userspace scales linearly

HW load balancer further improves scaling with number of cores

HW abstraction of networking SoC



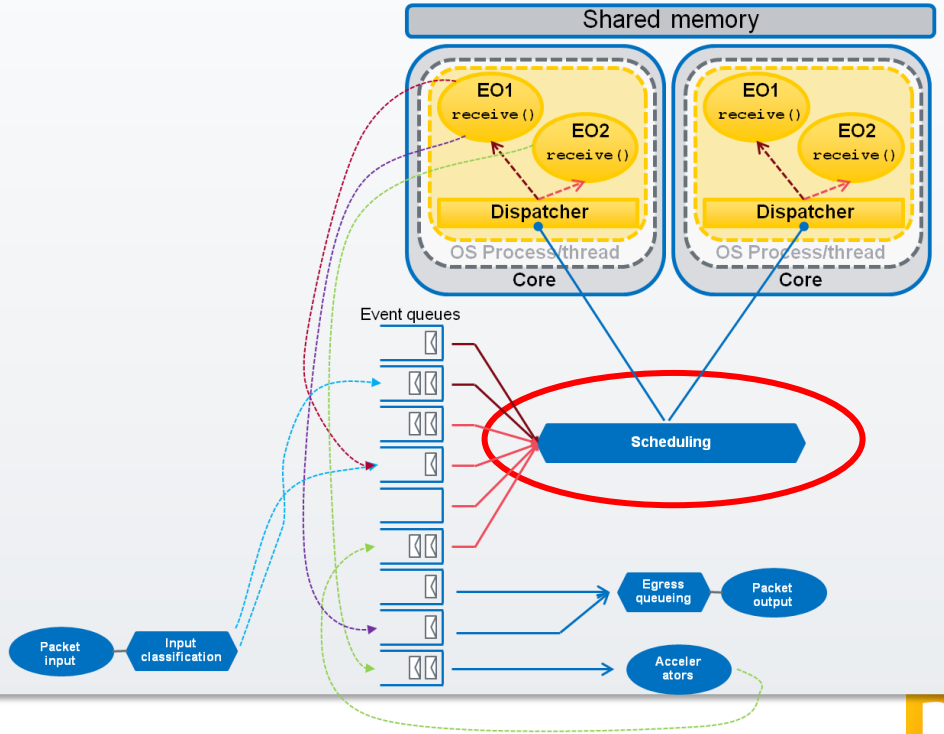
Logical functionality and packet flow in typical networking SoC

Null latency intelligent packet scheduling to the cores

SW abstraction of networking SoC

Open Event Machine SW architecture

Term	Description
EO 	Execution Object, an OpenEM term. Contains a receive function / handler .Part of application
Dispatcher 	SW running the main loop of each thread. Interfaces with the scheduler, maps queues to Eos. Part of EM implementation
OS Process/thread 	One per core (or HW thread) with single core affinity. Non-OS / bare metal also possible
Scheduling 	HW block scheduling events from event queues towards SW on cores
Event queues 	FIFOs containing events to be scheduled
Events 	Packets, messages, descriptors describing work



What do we need

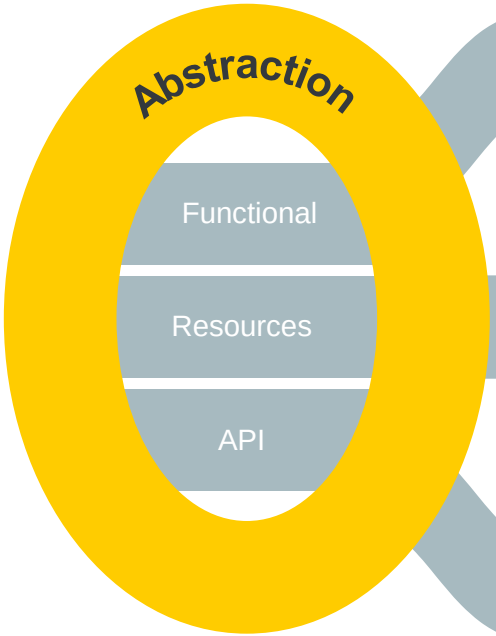
1000x

Abstraction

Scalability

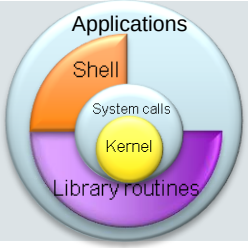
Efficiency

Abstraction is the key for portability

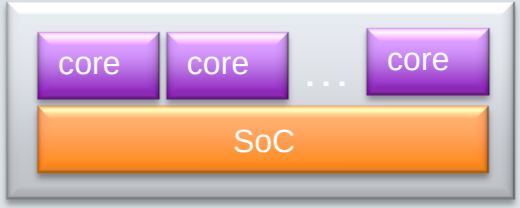


Maintain same architectural split across platforms

Agnostic to programming model



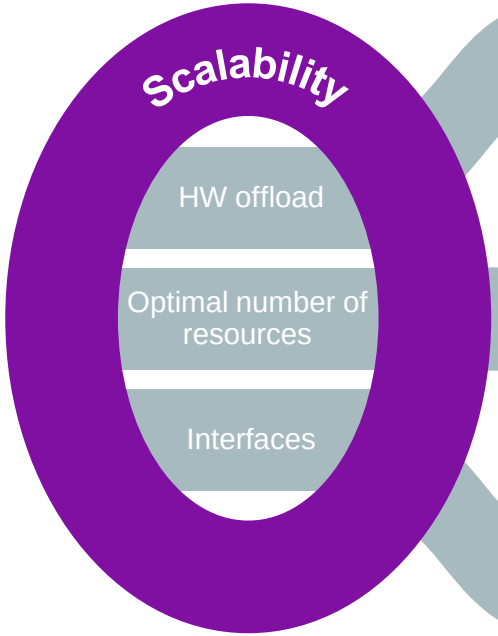
No need to (re)program as SoC or core resources increase. “Single core” programming model



Application portability

Common terminology

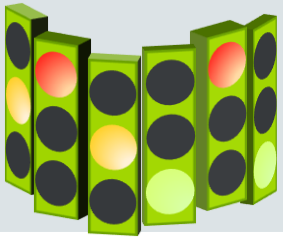
Scalability for optimal capacity



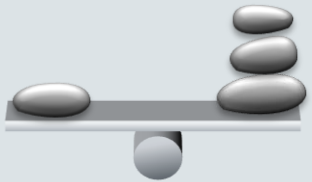
Eliminates single core bottleneck

Allows dynamic load balancing

Minimize the serialization

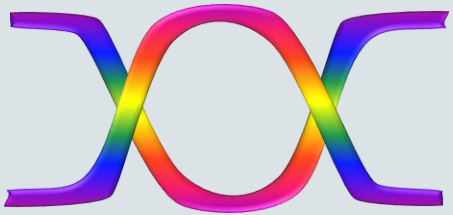


Optimal balance of cores, accelerators and interfaces

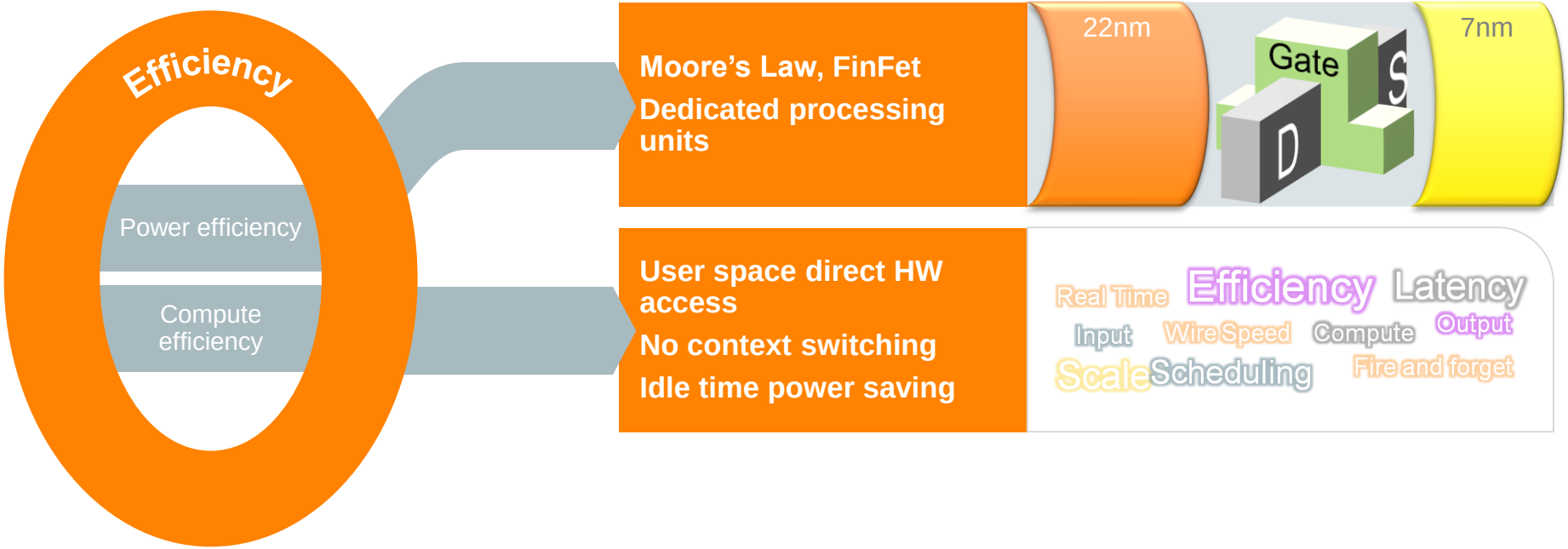


Ready for ultrafast interface technologies

No packet drop before classification



Efficiency for maximum density



Open Data Plane

ODP will be the de-facto data plane programming model

1

Common API for networking SoC

Abstraction with performance

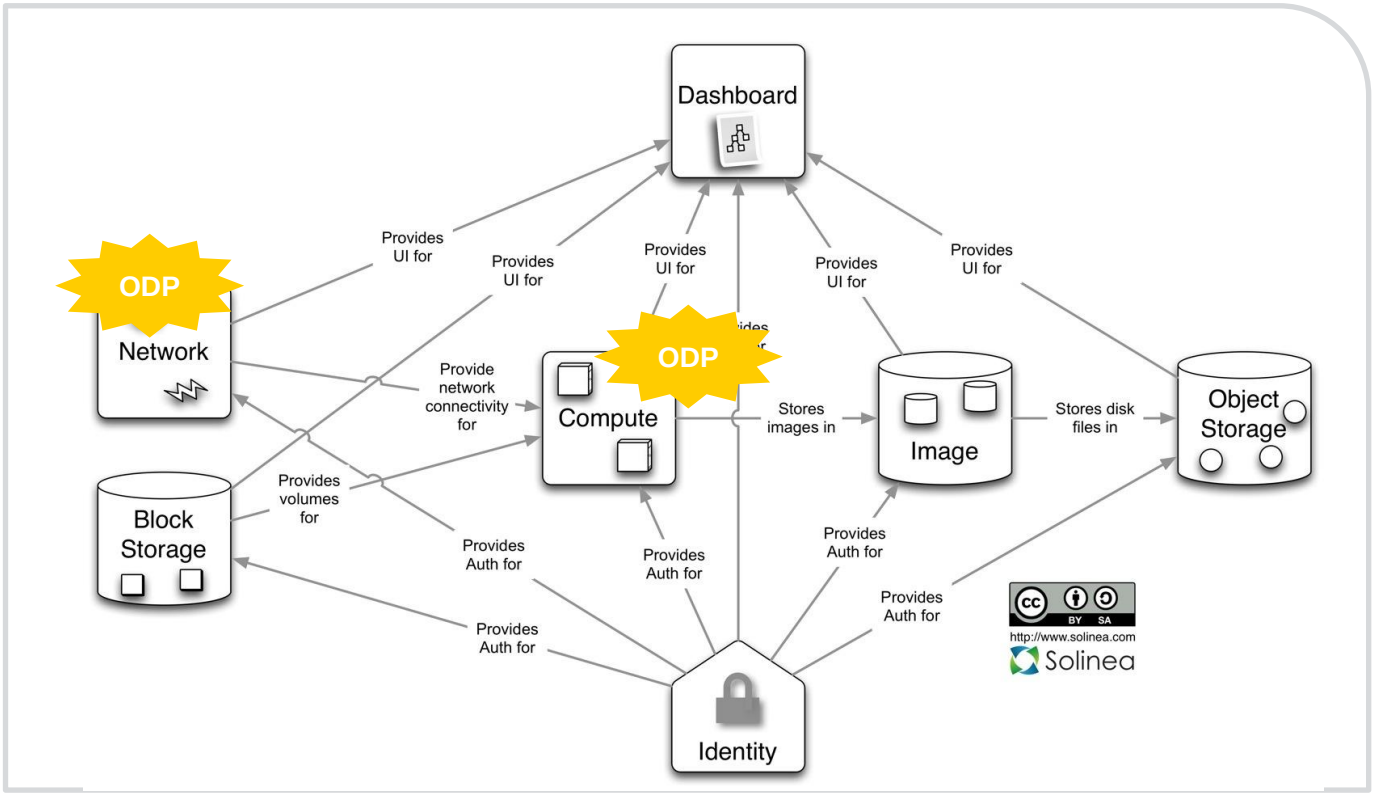
Application portability across platforms, time to market

Scaling to any number of cores and accelerators

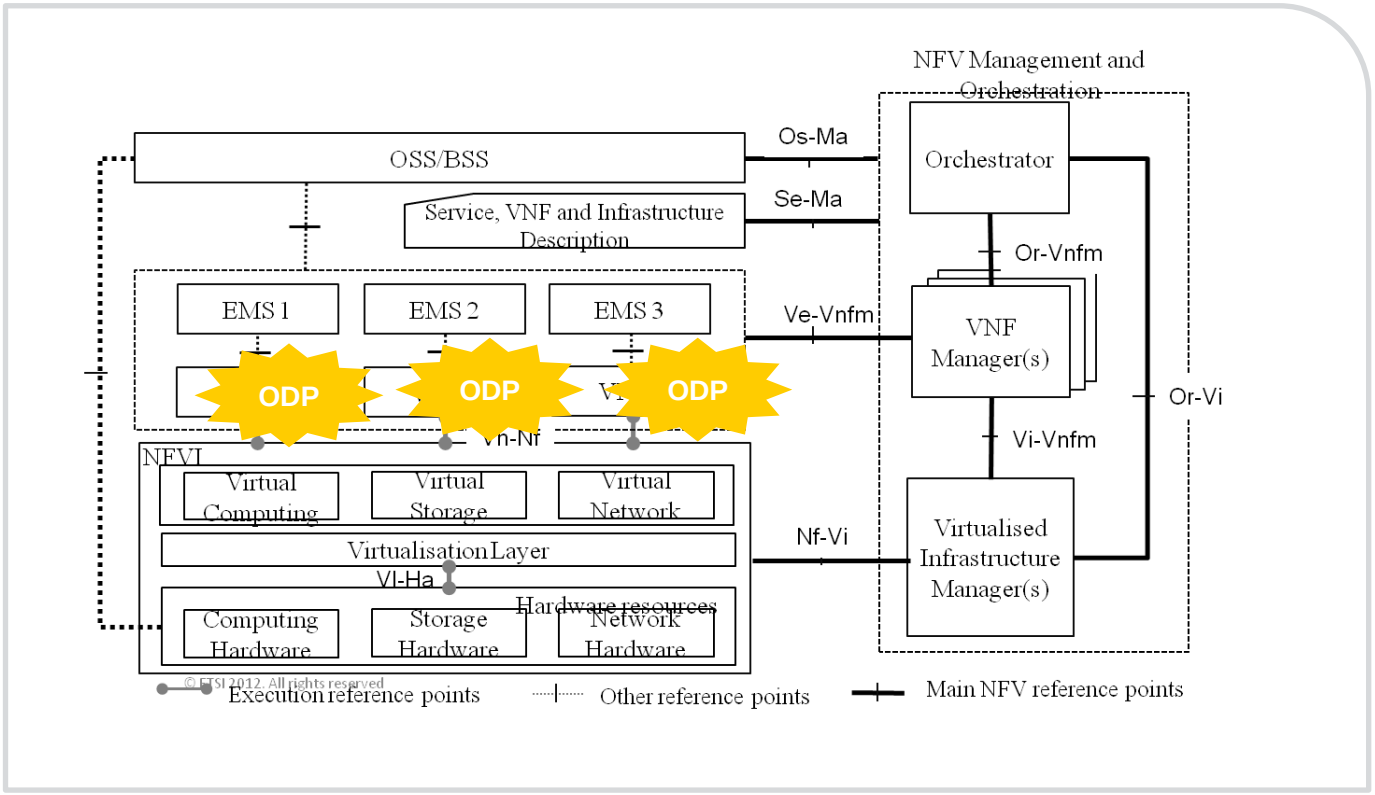
Efficiency in mind

OpenDataPlane™

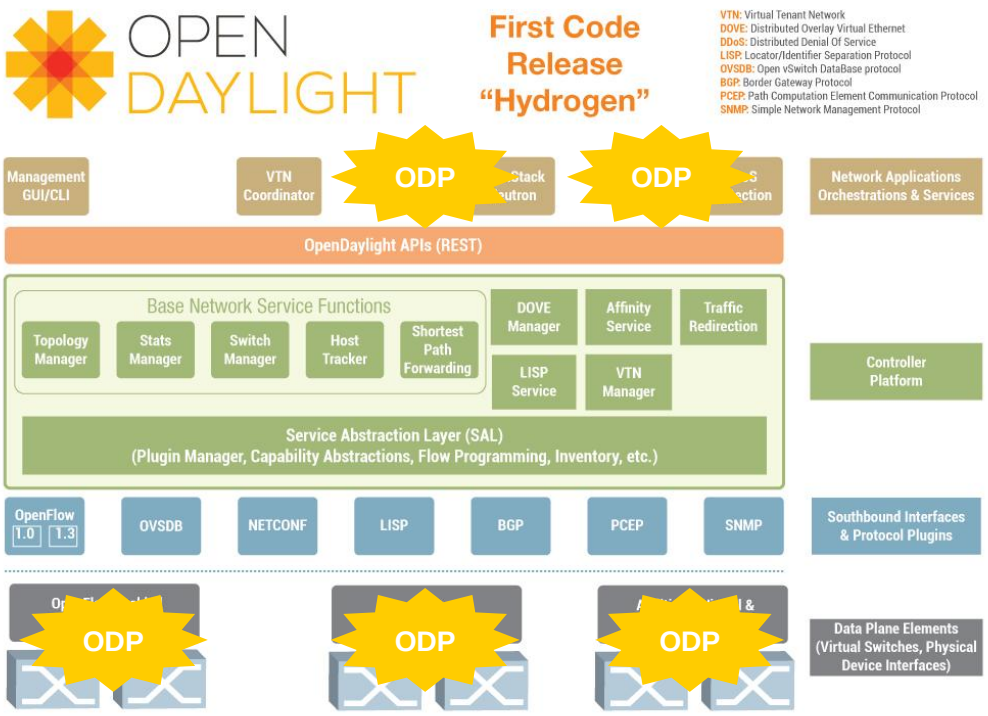
How does ODP map to Openstack



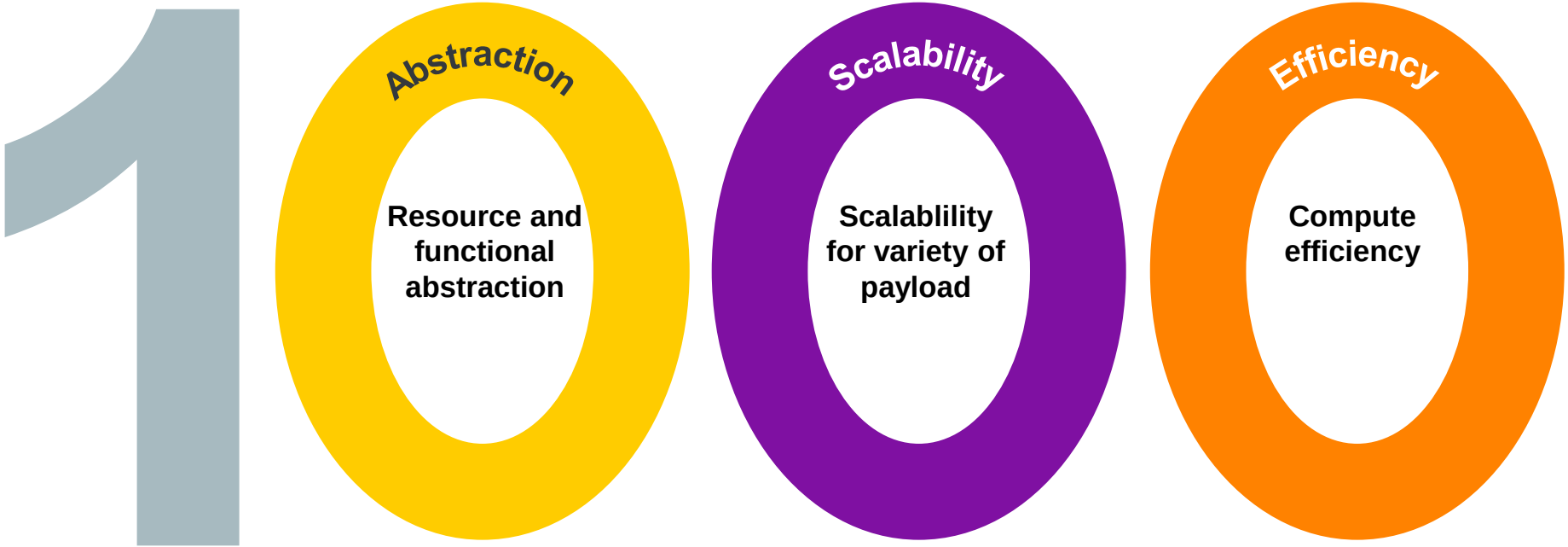
How does ODP map to NFV



How does ODP map to Open Daylight



Summary



… get ready for 1000x packet compute

Questions?

1GB of personalized data per user per day means 1000x capacity compared today.