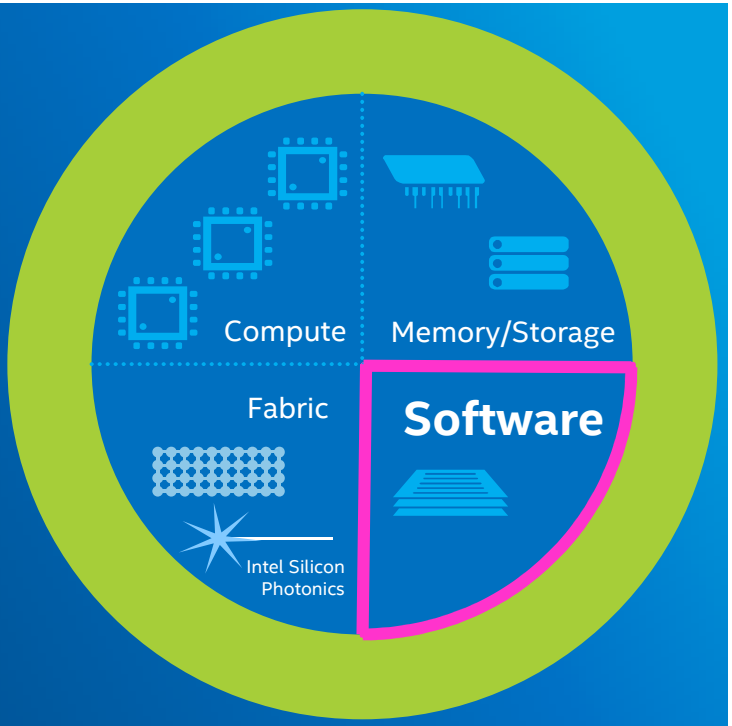


mOS

Robert W. Wisniewski, Chief Software Architect Extreme Scale Computing

On behalf of Rolf Riesen, lead architect for mOS

Supercomputing 2015

Legal Disclaimer

Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel, Intel Xeon, Intel Core microarchitecture, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM 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 http://www.intel.com/performance.

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.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

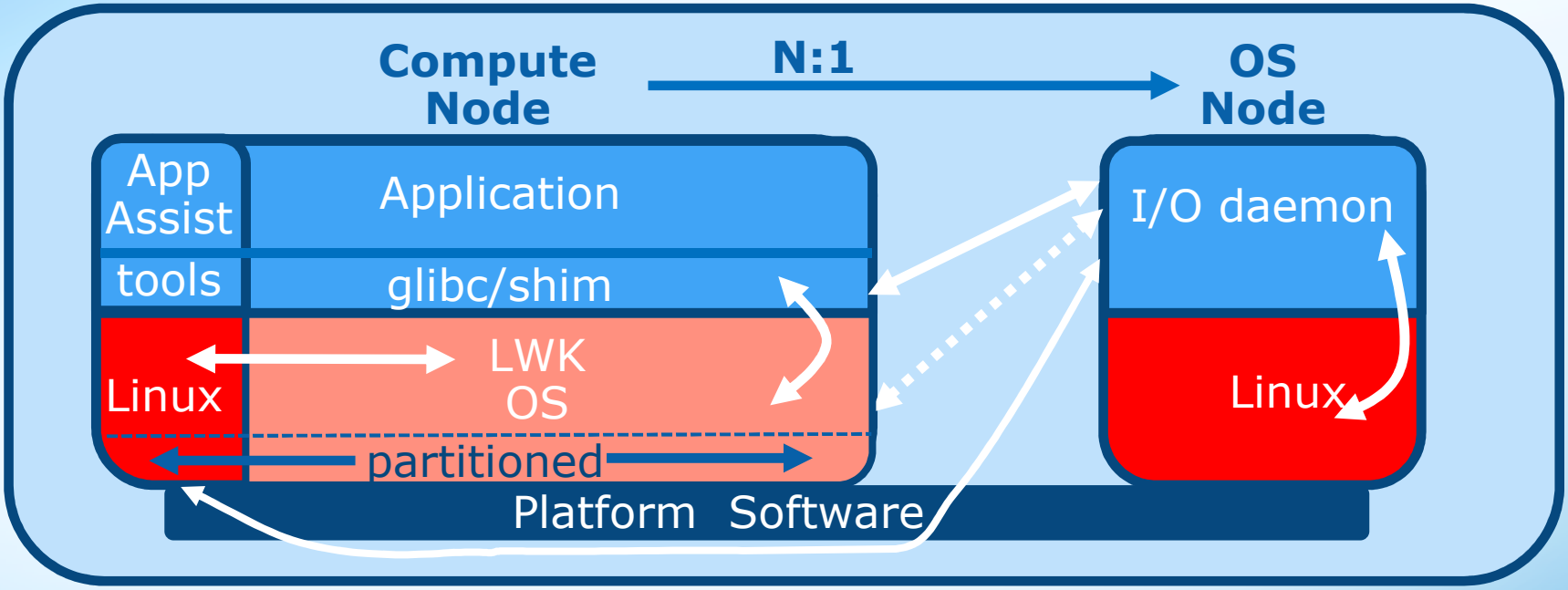
No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

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, Xeon, Xeon Phi, Core, and others are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

*Other names and brands may be claimed as the property of others

mOS: Expanded Compute Node View



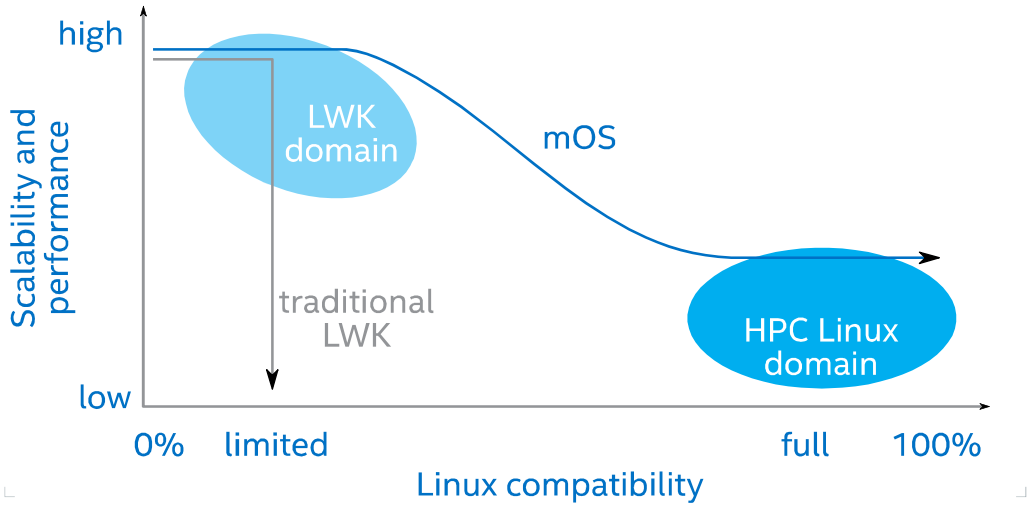
- LWK performance for core HPC application resource requests
- Nimble to adapt to new technology
- Linux compatibility
- Ability to leverage core variability on the compute node

Lightweight kernels and Linux

In the past, it was possible to achieve performance and scalability. Or, one could run Linux. But not both.

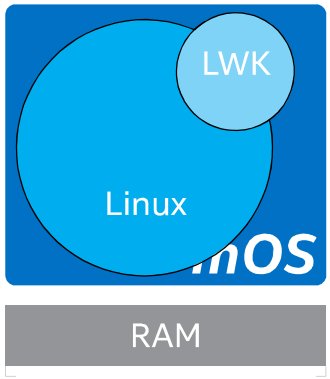
With an architecture like mOS, it is possible to have a more gradual path from the upper left LWK corner to the lower right FWK corner.

An application's choice of which features it wants to use, influences the overall performance and scalability.

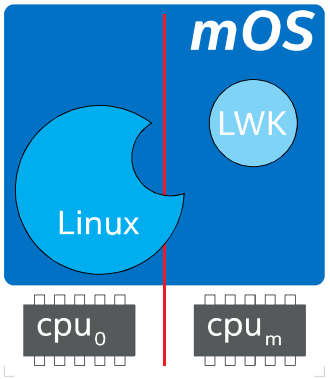


An embedded LWK

- We're neither trimming Linux to an LWK
- Nor are we adding Linux functionality to an LWK
- We are compiling our LWK into the Linux kernel
- Then, for each logical CPU, decide which kernel has control



Code in memory

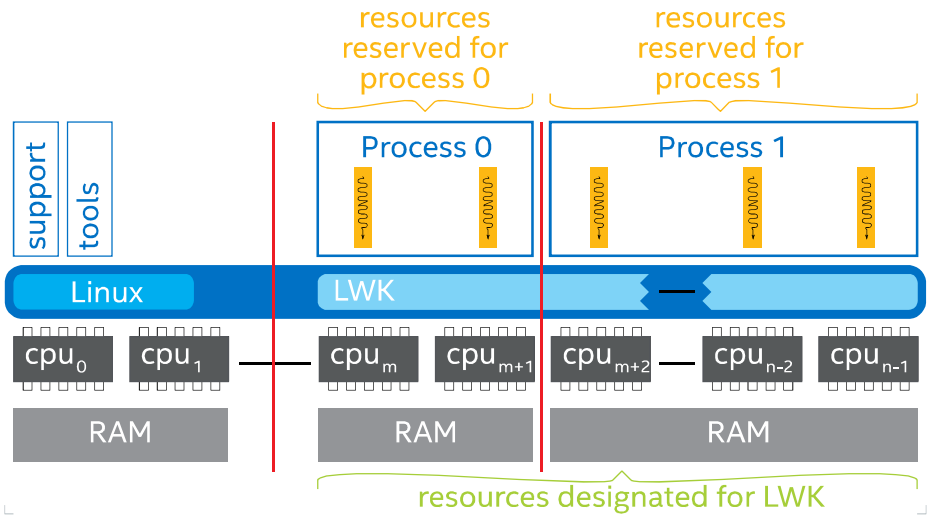


Code running

Partition model

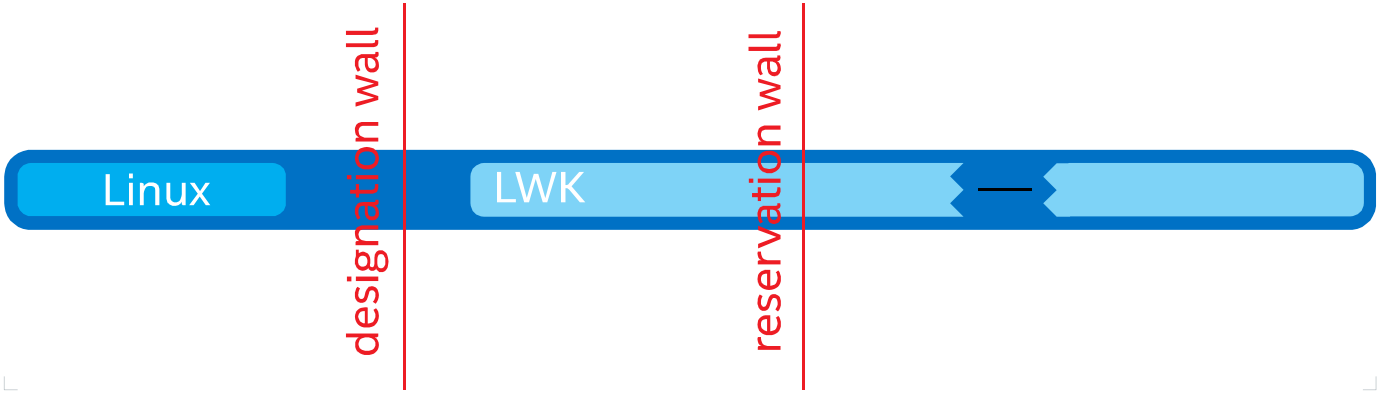
Three stages of partitioning:

1. **Designation** of resources to LWK happens at boot time
2. **Reservation** of a subset of designated resources happens at process launch time
3. **Allocation** of reserved resources happens at runtime



Partition model cont.

- Partition walls are a fundamental principle of mOS
- Important for isolation between Linux and LWK
- Needed to maintain NUMA separation and avoid demand paging
  - No pool of pages in some NUMA domain for later use
- Needed to “hold” resources for later process starts
  - Resource contract is “signed” at yod process launch

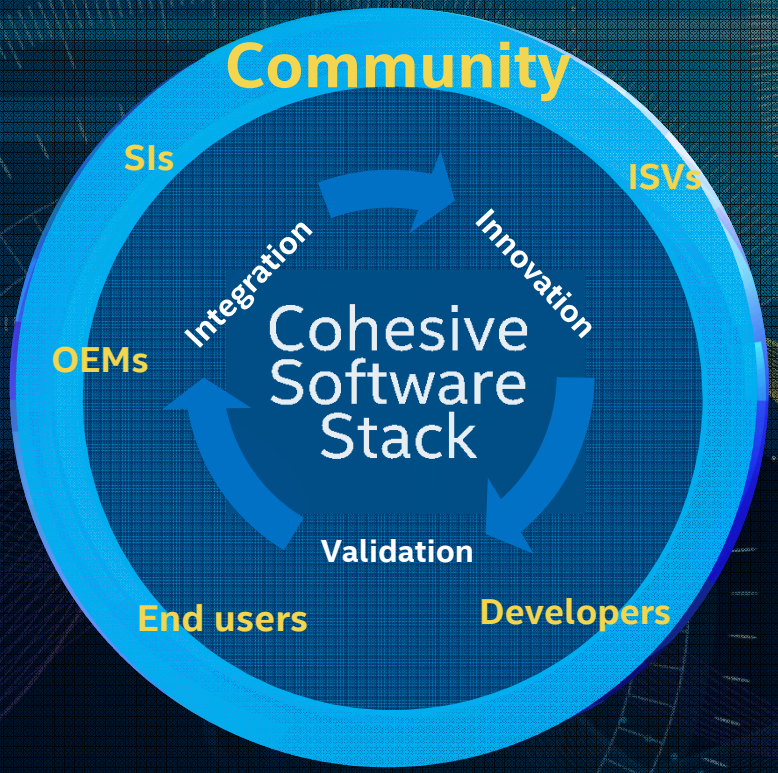


Goals for the HPC software community

Community name: OpenHPC **Web Address:** www.openhpc.community

- Provide a common platform to the HPC community that works across multiple segments and on which end-users can collaborate and innovate.
- Simplify the complexity of installation, configuration, and ongoing maintenance of an HPC software stack
- Receive contributions and feedback from community
- Enable developers to focus on their differentiation and unique area, rather than having to spend effort on developing, testing, and maintaining a whole stack
- Deliver integrated hardware and software innovations to ease the path to extreme scale

Collaborative, Open and Highly Scalable Software Stack



An open community effort

- Intel is engaging with the open source community and key ecosystem partners to enable easy HPC stack building and testing

Benefits the entire HPC ecosystem

- Simplify configuration, management and use
- Accelerate application development
- Turnkey to customizable

Industry collaboration

- Co-developed with community and partners
- Open source availability
- Allows for differentiation
- Innovating system hardware and software[1]

1. Reliability system management, Scalable/low-noise kernels, New data management models

Community Building Blocks for HPC Systems