An Overview of Post-K Development

Yutaka Ishikawa
RIKEN AICS

9:00– 9:50 31st of January, 2018

28th-31st, January, 2018
HPC Asia 2018, Tokyo, Japan

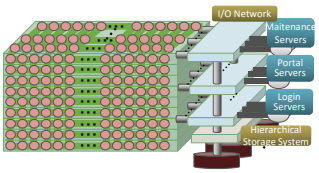
Outline

- **FLAGSHIP2020 Project**
- **Co-Design and R&D organization**
- **CPU Architecture and Post-K System Software**
- **Light-Weight OS Kernel: IHK/McKernel**

FLAGSHIP2020 Project

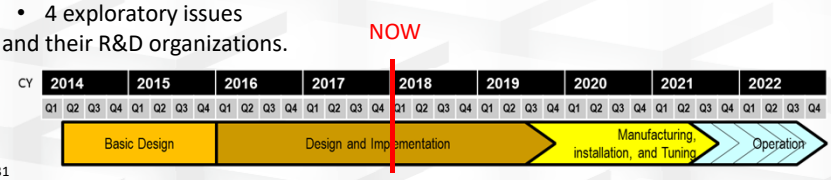
AICS

- Missions
  - Building the Japanese national flagship supercomputer, post K, and
  - Developing wide range of HPC applications, running on post K, in order to solve social and science issues in Japan



- Project organization
  - Post K Computer development
    - RIKEN AICS is in charge of development
    - Fujitsu is vendor partner.
    - International collaborations: DOE, JLESC, CEA (NCSA, ANL, UTK, JSC, BSC, INRIA, RIKEN)
  - Applications
    - The government selected
      - 9 social & scientific priority issues
      - 4 exploratory issues

      and their R&D organizations.
- Current Status
  - In the middle of "Design and Implementation" phase.
  - CPU has been designed and is being implemented
    - now evaluated by simulators and compilers
  - System software stack has been designed and is being implemented

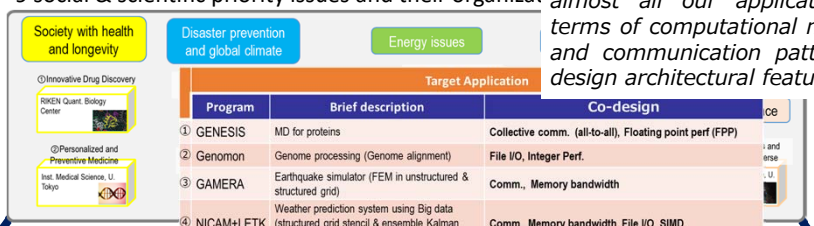


Co-design: *How we are running under the concept*

AICS

9 social & scientific priority issues and their organizat

Those are representatives of almost all our applications in terms of computational methods and communication patterns to design architectural features.



Target Application

	Program	Brief description	Co-design
①	GENESIS	MD for proteins	Collective comm. (all-to-all), Floating point perf (FPP)
②	Genomon	Genome processing (Genome alignment)	File I/O, Integer Perf.
③	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)	Comm., Memory bandwidth
④	NICAM+LETK	Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter)	Comm., Memory bandwidth, File I/O, SIMD
⑤	NTChem	molecular electronic (structure calculation)	Collective comm. (all-to-all, allreduce), FPP, SIMD,
⑥	FFB	Large Eddy Simulation (unstructured grid)	Comm., Memory bandwidth,
⑦	RSDFT	an ab-initio program (density functional theory)	Collective comm. (bcast), FFP
⑧	Adventure	Computational Mechanics System for Large Scale Analysis and Design (unstructured grid)	Comm., Memory bandwidth, SIMD
⑨	CCS-QCD	Lattice QCD simulation (structured grid Monte Carlo)	Comm., Memory bandwidth, Collective comm. (allreduce)

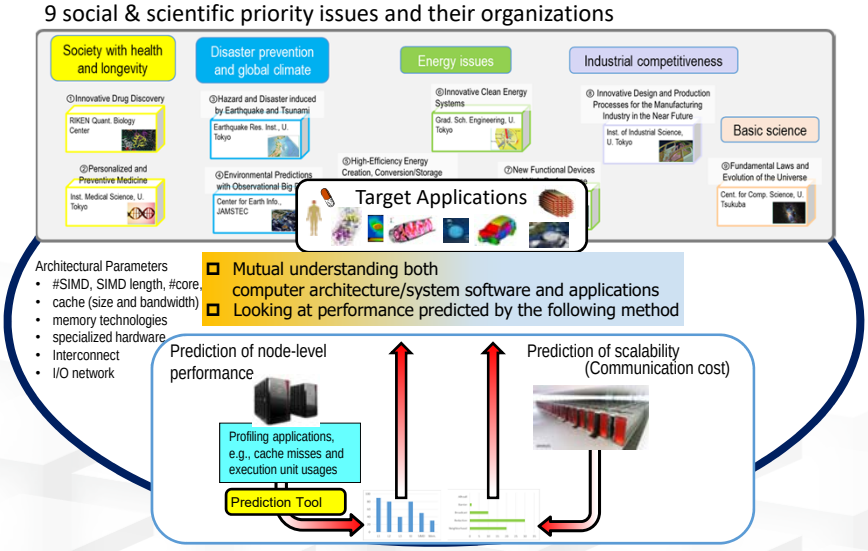
Architectural Parameters
- #SIMD, SIMD length, #core, #NUMA node
- cache (size and bandwidth)
- memory technologies
- specialized hardware
- Interconnect
- I/O network

FUJITSU

RIKEN AICS

Co-design: *How we are running under the concept*

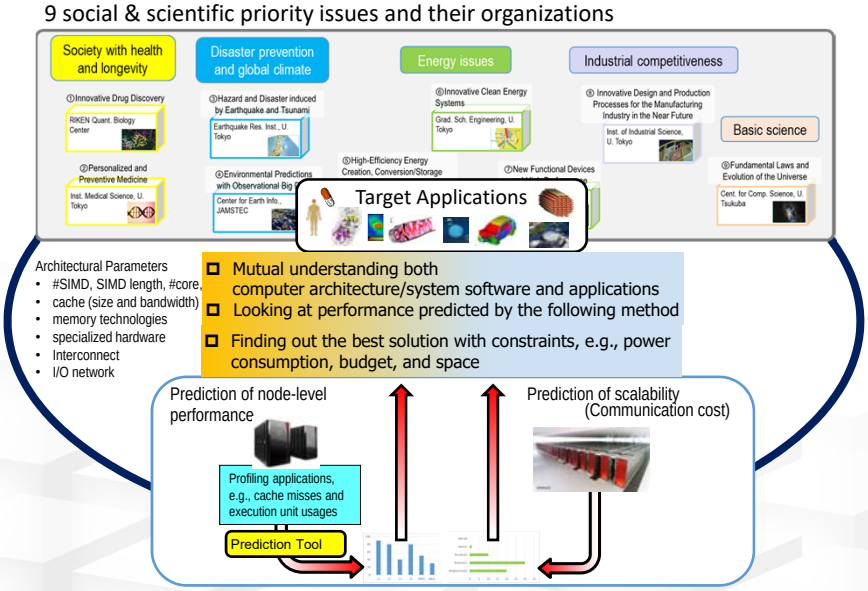
9 social & scientific priority issues and their organizations



5

Co-design: *How we are running under the concept*

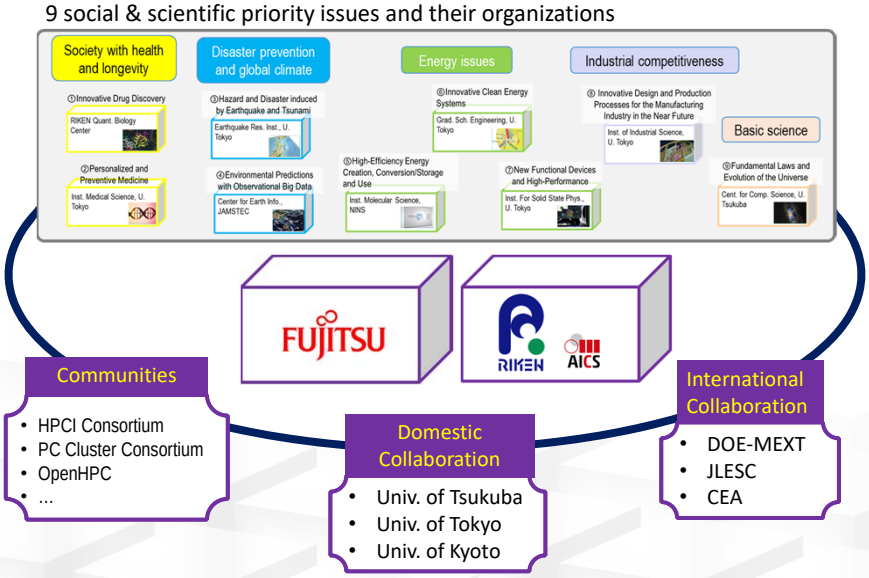
9 social & scientific priority issues and their organizations



6

R&D Organization

9 social & scientific priority issues and their organizations



Communities

- HPCI Consortium
- PC Cluster Consortium
- OpenHPC
- …

Domestic Collaboration

- Univ. of Tsukuba
- Univ. of Tokyo
- Univ. of Kyoto

International Collaboration

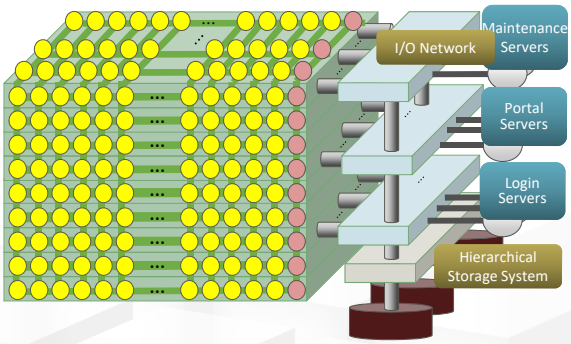
- DOE-MEXT
- JLESC
- CEA

International Collaborations

- **DOE-MEXT**
  - Optimized Memory Management, Efficient MPI for exascale, Dynamic Execution Runtime, Storage Architectures, Metadata and active storage, Storage as a Service, Parallel I/O Libraries, MiniApps for Exascale CoDesign, Performance Models for Proxy Apps, OpenMP/XMP Runtime, Programming Models for Heterogeneity, LLVM for vectorization, Power Monitoring and Control, Power Steering, Resilience API, Shared Fault Data, etc.
- **JLESC: Joint Laboratory on Extreme Scale Computing (NCSA, ANL, UTK, JSC, BSC, INRIA, RIKEN)**
  - Simplified Sustained System performance benchmark , Developer tools for porting and tuning parallel applications on extreme-scale parallel systems,. HPC libraries for solving dense symmetric eigenvalue problems,.DTF+SZ: Using lossy data compression for direct data transfer in multi-component applications, Process-in-Process: Techniques for Practical Address-Space Sharing
- **CEA**
  - Programming Language, Runtime Environment, Energy-aware batch job scheduler, Large DFT calculations and QM/MM, Application of High Performance Computing to Earthquake Related Issues of Nuclear Power Plant Facilities, KPIs (Key Performance Indicators)

An Overview of Post K

- **Compute Node, Compute + I/O Node**
  - Armv8-A SVE + Fujitsu Extension
- **6D mesh/torus Interconnect**
- **3-level hierarchical storage system**
  - Silicon Disk
  - Magnetic Disk
  - Storage for archive

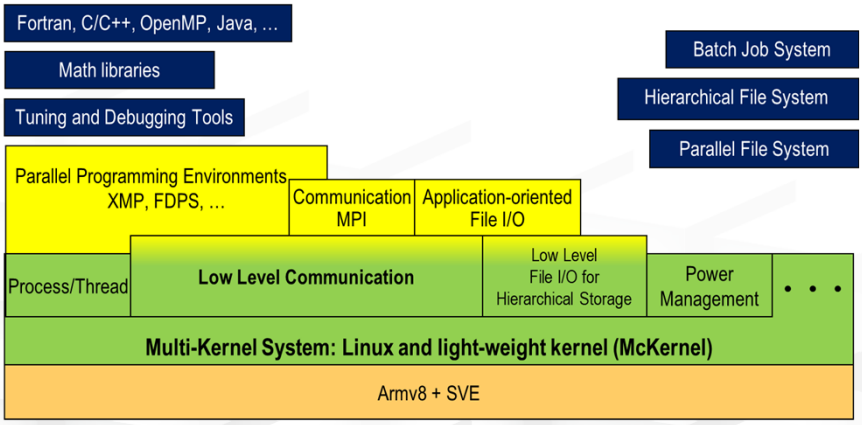


An Overview of System Software Stack

Easy of use is one of our KPIs (Key Performance Indicators)

Providing wide range of applications/tools/libraries/compilers

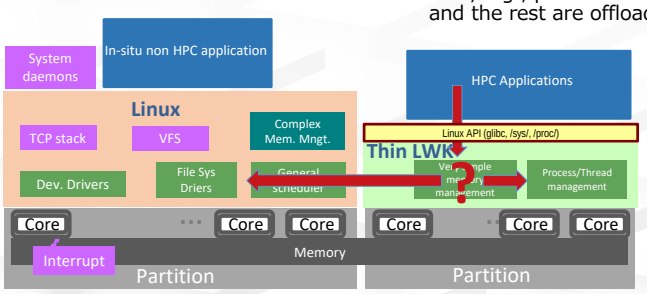
Linux Distribution



IHK/McKernel developed at RIKEN

- **Partition resources (CPU cores, memory)**
- **Full Linux kernel on some cores**
  - System daemons and in-situ non HPC applications
  - Device drivers
- **Light-weight kernel(LWK), McKernel on other cores**
  - HPC applications

- **IHK: Linux kernel module**
  Interface for Heterogeneous Kernels
  - Allows dynamically partitioning of node resources: CPU cores, physical memory, …
  - Enables management of LWKs (assign resources, load, boot, destroy, etc..)
  - Provides inter-kernel communication, messaging and notification
- **McKernel: Light-weight kernel**
  - Is designed for HPC, noiseless, simple
  - Implements only performance sensitive system calls, e.g., process and memory management, and the rest are offloaded to Linux

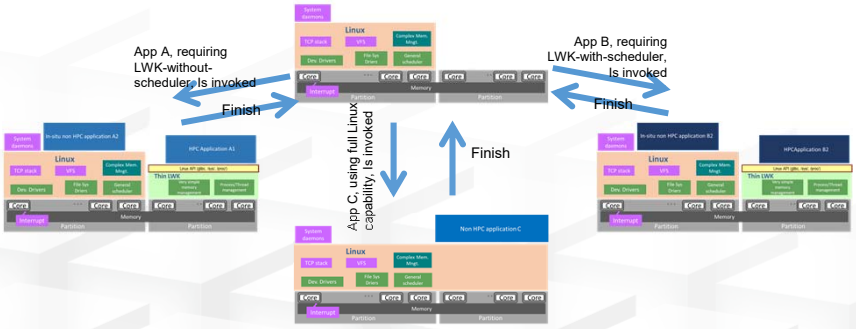


- Executes the same binary of Linux without any recompilation

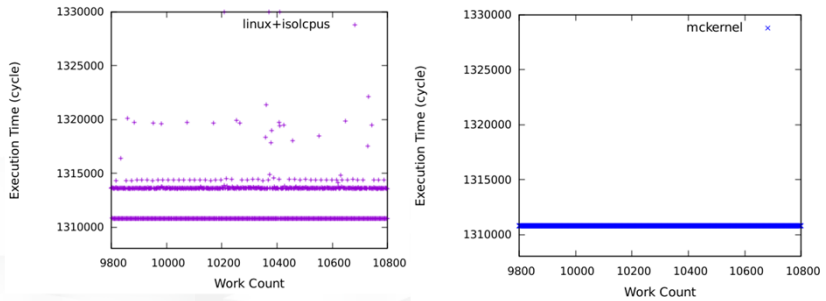
- IHK/McKernel runs on
  - Intel Xeon and Xeon phi
  - Fujitsu FX10 and FX100 (Experiments)

How to deploy IHK/McKernel

- Linux Kernel with IHK kernel module is resident
  - daemons for job scheduler and etc. run on Linux
- McKernel is dynamically reloaded (rebooted) by IHK for each application
  - No hardware reboot



FWQ (Fixed Work Quanta) Benchmark

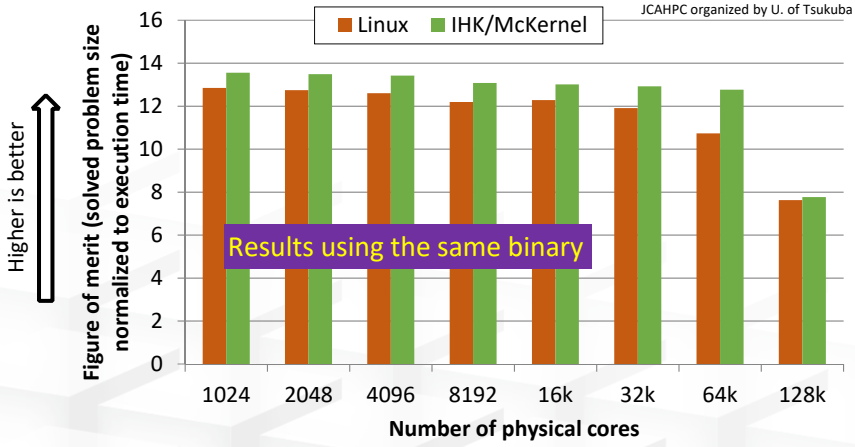


https://asc.llnl.gov/sequoia/benchmarks

GeoFEM (University of Tokyo)

- **ICCG with Additive Schwartz Domain Decomposition - weak scaling**
- **Up to 18% improvement**

Acknowledgement: Kengo Nakajima, University of Tokyo, for providing GeoFEM. This result is on Oakforest-PACS supercomputer, 25 PF in peak, at JCAHPC organized by U. of Tsukuba and U. of Tokyo

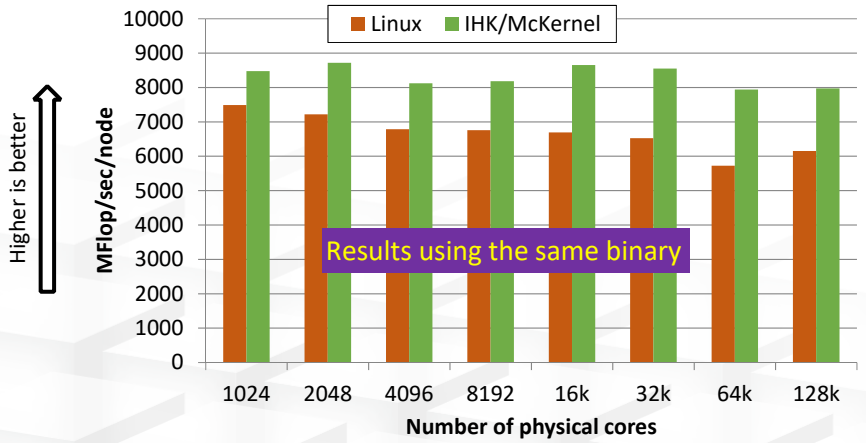


Latest results will be presented at IPDPS'18 by Balazs Gerofi, et al.,
Performance and Scalability of Lightweight Multi-Kernel based Operating Systems

CCS-QCD (University of Tsukuba)

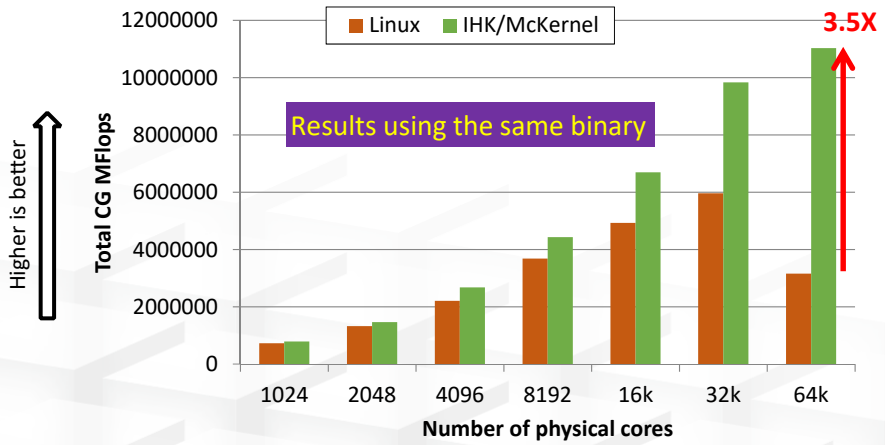
- Lattice quantum chromodynamics code - weak scaling
- Up to 38% improvement

Acknowledgement: Ken'ichi Ishikawa, Hiroshima University, providing CCS-QCD. This result is on Oakforest-PACS supercomputer, 25 PF in peak, at JCAHPC organized by U. of Tsukuba and U. of Tokyo



miniFE (CORAL benchmark suite)

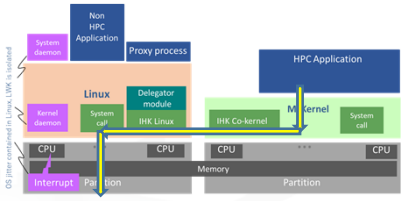
- Conjugate gradient - strong scaling
- Up to 3.5X improvement (Linux falls over..)



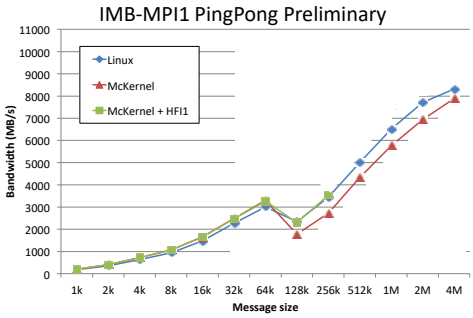
Oakforest-PACS supercomputer, 25 PF in peak, at JCAHPC organized by U. of Tsukuba and U. of Tokyo

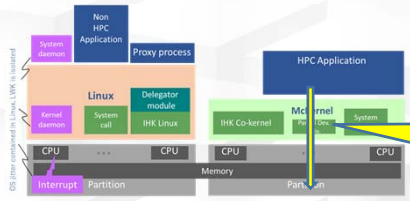
Ongoing R&D: Device driver integration

- In the current Omnipath, the writev systemcall is issued in large message sizes in order to utilize the RDMA capability.



This causes extra overhead, and thus communication latency and bandwidth is worse than Linux in the large message sizes.

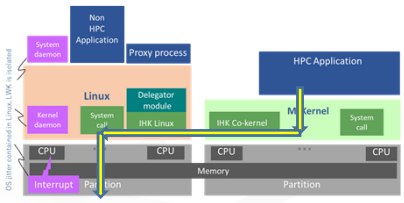




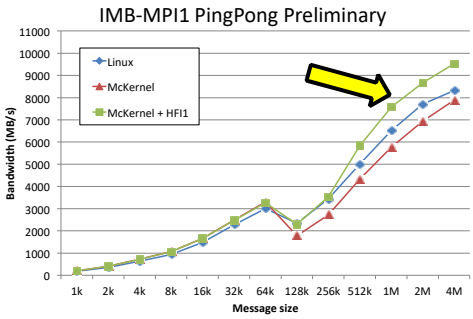
- A partial Linux driver is ported to McKernel with less effort
  - ✓ Making McKernel access the same kernel virtual address of Linux
  - ✓ Providing the same Linux internal functions used by a device driver, such as kmalloc and synchronization primitives

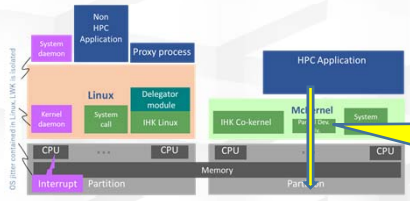
Ongoing R&D: Device driver integration

- In the current Omnipath, the writev systemcall is issued in large message sizes in order to utilize the RDMA capability.



This causes extra overhead, and thus communication latency and bandwidth is worse than Linux in the large message sizes.





- A partial Linux driver is ported to McKernel with less effort
  - ✓ Making McKernel access the same kernel virtual address of Linux
  - ✓ Providing the same Linux internal functions used by a device driver, such as kmalloc and synchronization primitives
  - ✓ …

Concluding Remarks

- The system software stack for post-K is being designed and implemented with the leverage of international collaborations, CEA, DOE Labs, and JLESC (NCSA, INRIA, ANL, BSC, JSC, RIKEN)

- The software stack developed at RIKEN is open source
- It also runs on Intel Xeon and Xeon phi