A Multi-Kernel Survey for High-Performance Computing

Balazs Gerofi[†], Yutaka Ishikawa[†], Rolf Riesen[‡], Robert W. Wisniewski[‡], Yoonho Park[§], Bryan Rosenburg[§]

[†] RIKEN Advanced Institute for Computational Science, JAPAN
[‡] Intel Corporation, US
[§] IBM T.J. Watson Research Center, US

2016/06 ROSS’16 Kyoto, JAPAN

Background

- **Requirements of OS Kernel targeting high-end HPC**
  - Noiseless execution environment for bulk-synchronous applications
  - Ability to easily adapt to new/future system architectures
    - E.g.: manycore CPUs, heterogenous core architectures, deep memory hierarchy, etc.
      - New process/thread management, memory management, …
  - Ability to adapt to new/future application demand
    - Big-Data, in-situ applications
      - Support data flow from Internet devices to compute nodes
      - Optimize data movement

	Approach	Pros.	Cons.
Full-Weight Kernel (FWK) e.g. Linux	Disabling, removing, tuning, reimplementation, and adding new features	Large community support results in rapid new hardware adaptation	• Hard to implement a new feature if the original mechanism is conflicted with the new feature • Hard to follow the latest kernel distribution due to local large modifications
Light-Weight Kernel (LWK)	Implementation from scratch and adding new features	Easy to extend it because of small in terms of logic and code size	• Applications, running on FWK, cannot run always in LWK • Small community maintenance limits rapid growth • Lack of device drivers

Background

- **Requirements of OS Kernel targeting high-end HPC**
  - Noiseless execution environment for bulk-synchronous applications
  - Ability to easily adapt to new/future system architectures
    - E.g.: manycore CPUs, heterogenous core architectures, deep memory hierarchy, etc.
      - New p
  - Ability to
    - Big-Dat
      - Suppo
      - Optimize data movement

Lightweight multi-kernels (also referred to as hybrid kernels) in HPC have received significant attention recently

	Approach	Pros.	Cons.
Full-Weight Kernel (FWK) e.g. Linux	Disabling, removing, tuning, reimplementation, and adding new features	Large community support results in rapid new hardware adaptation	• Hard to implement a new feature if the original mechanism is conflicted with the new feature • Hard to follow the latest kernel distribution due to local large modifications
Light-Weight Kernel (LWK)	Implementation from scratch and adding new features	Easy to extend it because of small in terms of logic and code size	• Applications, running on FWK, cannot run always in LWK • Small community maintenance limits rapid growth • Lack of device drivers

Motivation

- **Lightweight multi-kernels (also referred to as hybrid kernels) in HPC have received significant attention recently**
- **Several research projects are exploring this direction:**
  - FusedOS @ IBM
  - IHK/McKernel led by RIKEN
  - mOS @ Intel
  - Hobbes (i.e., Pisces/Kitten, Kitten/Palacios) led by Sandia
  - Fast and Fault-tolerant Microkernel-based System for Exascale Computing (FFMK) led by TU Dresden

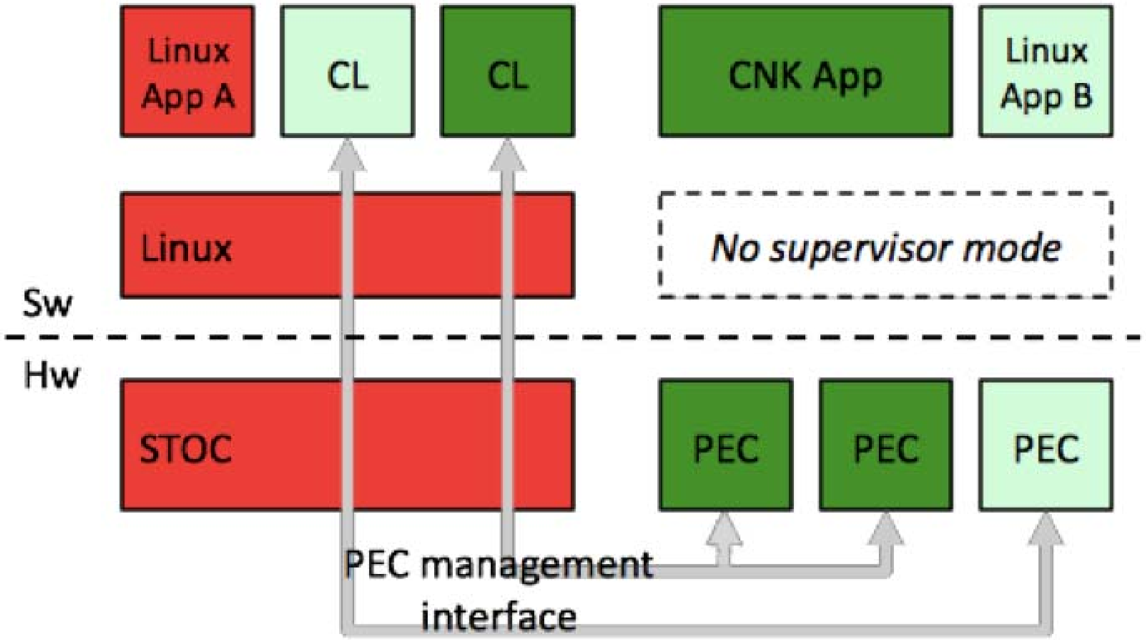
L4 + L^{4}Linux

- **What are the differences?**
- **Is there a common set of criteria?**
- **Can we classify them accordingly?**

Outline

- **Overview of Projects**
  - FusedOS, IHK/McKernel, mOS, FFMK, Hobbes
- **Characteristics, Comparison and Classification**
  - System Administrator Perspective
  - Application Perspective
  - Linux Perspective
  - Lightweight-kernel Perspective
- **Conclusion**

FusedOS @ IBM

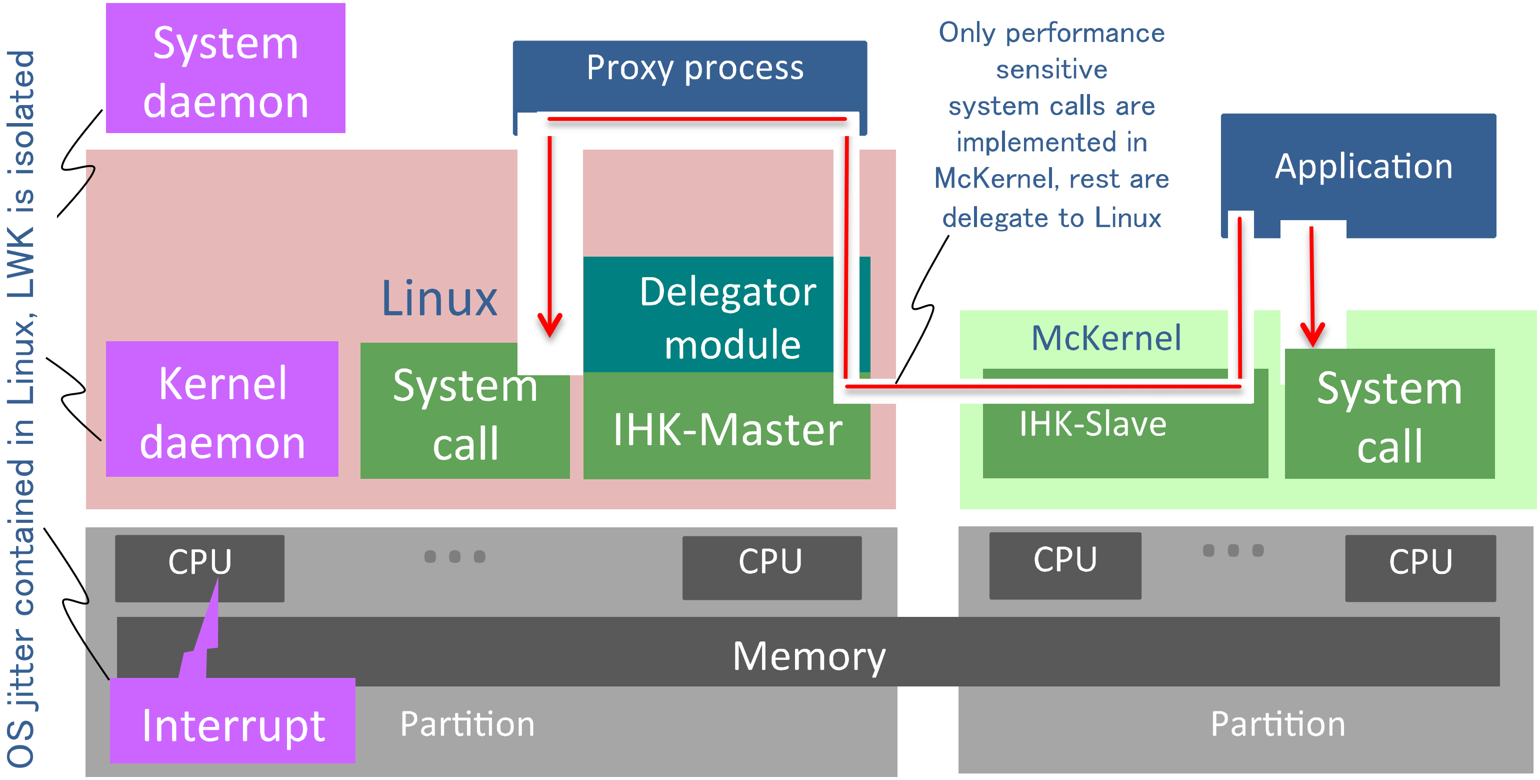


- **First proposal to run Linux and LWK side-by-side**
- **Linux runs the CNK Library (CL) in user-space**
  - a.k.a., proxy process in hybrid context
- **Traditional LWK component exists only in user-space on PEC**
  - All system calls are offloaded and handled by CL on Linux

- **STOC = Single-thread optimized core [1]**
- **PEC = Power-efficient core**

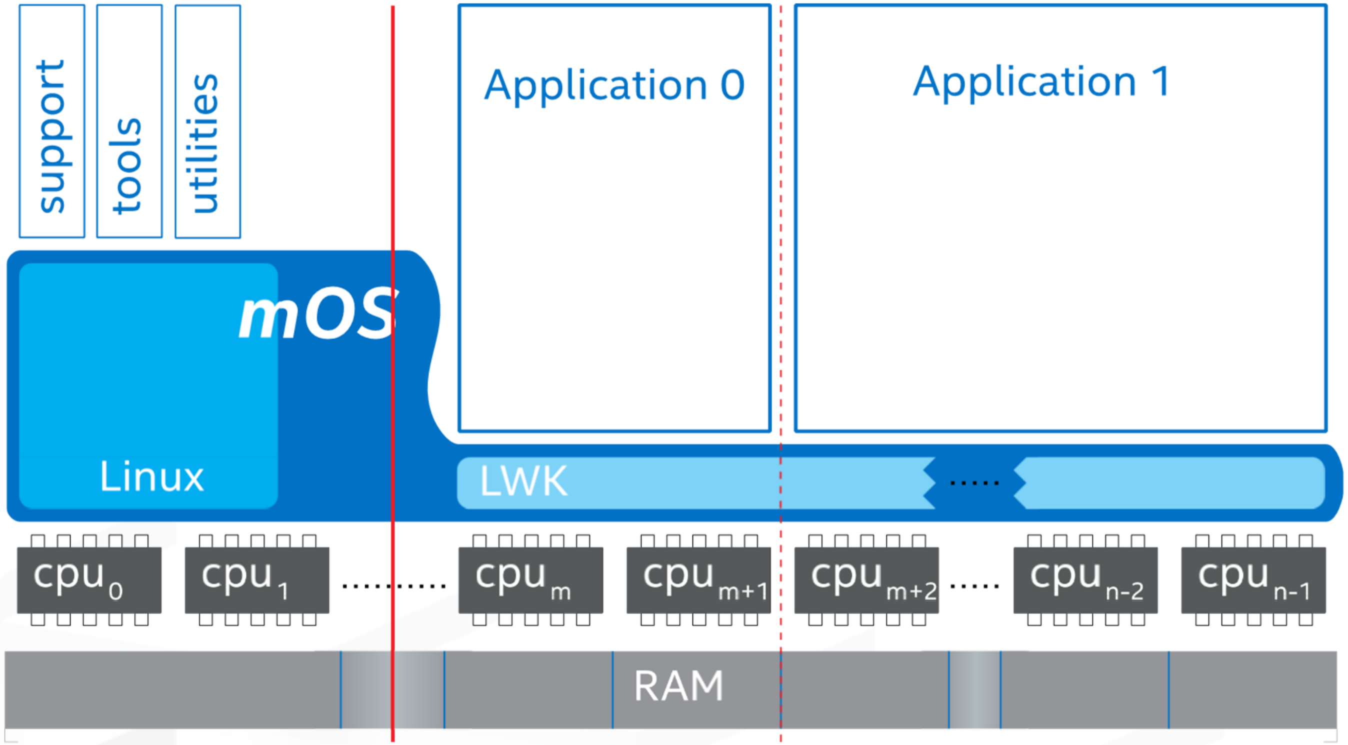
[1] http://hpc.mju.ac.kr/SIG_HPC/2013_Fall_Workshop/documents/2.%20FusedOS%20KISTI%20invited%20talk.pdf

IHK/McKernel led by RIKEN



- **Interface for Heterogeneous Kernels (IHK)**
  - Partitions system resources (CPU cores, memory)
  - Manages LWK instances
  - Provides communication between Linux and LWKs
- **McKernel**
  - LWK developed from scratch, relies on IHK
  - Standalone code-base
  - Proxy process offload model – only performance critical syscalls implemented in LWK

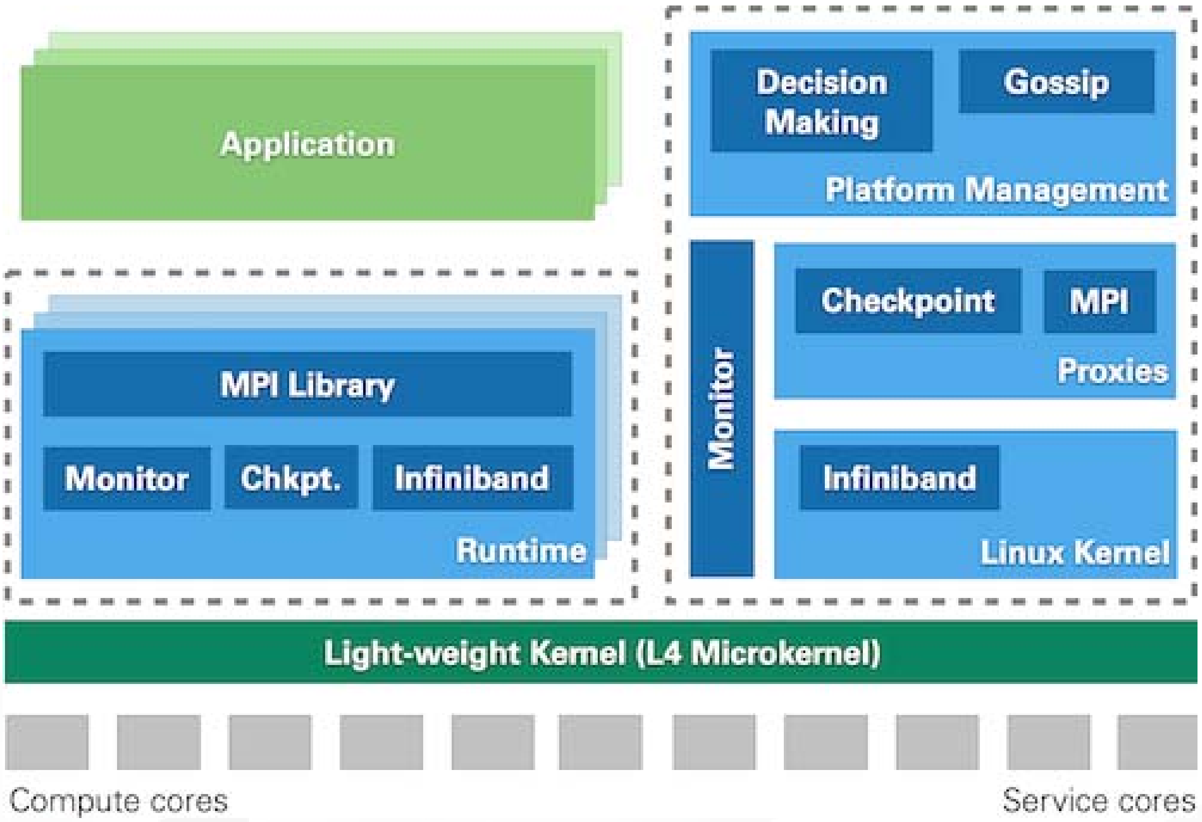
mOS @ Intel Corporation



- **mOS compiles the LWK code into Linux**
  - Restricts LWK dedicated cores to the LWK code-base
  - Provides its own memory management and simplified scheduling
- **Non-critical system calls are shipped to Linux by re-affinitizing (i.e., migrating) threads to Linux cores**
- **LWK data structures are/need to be Linux compatible**
- **LWK processes are visible in Linux**
  - Tools, pseudo file systems work

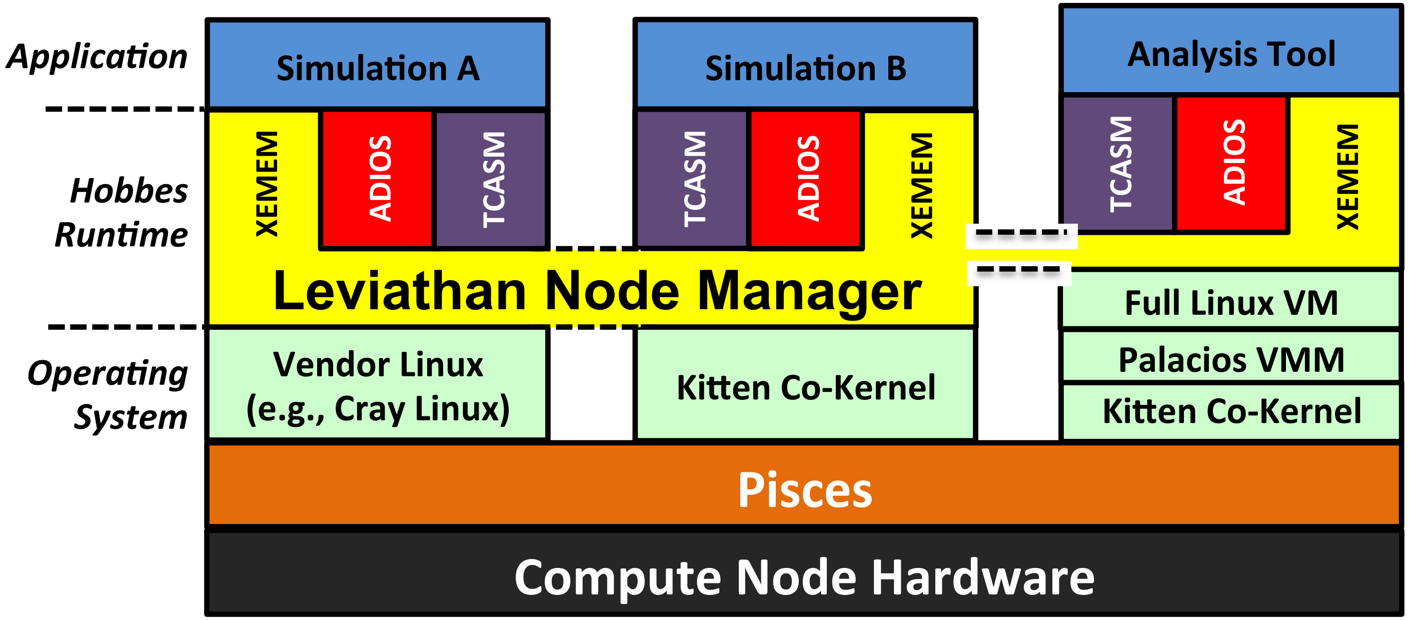
FFMK led by TU Dresden

L4 + L^4Linux



- **L4 microkernel boots node and Linux is run paravirtualized**
- **Performance-critical parts of application run directly on L4**
- **Non-critical parts reuse Linux**
  - Threads are attached/detached from/to Linux for system call execution
- **Currently all POSIX system calls executed in Linux**

Hobbes led by Sandia National Labs



- **Hobbes central concept: application composition**
- **Node OS has three main components:**
  - Kitten light-weight kernel, Pisces resource manager and Palacios VM monitor
- **Two configurations considered in this study:**
  - **Pisces/Kitten**: Linux boots node and Kitten runs in a resource partition
    - Similar to IHK/McKernel, but no system call offloading
  - **Kitten/Palacios**: Kitten boots the node and Linux is run in VM
    - Similar to FFMK, but VM relies on hardware virtualization support

Defining Characteristics and Criteria

Property		Short Description	Impact
System Administrator Perspective	Standalone LWK	Is the LWK a separate binary from Linux, and does it boot the cores it runs on?	
	Node boot	Which kernel is booted by the BIOS/Firmware of the node?	
	Resource partitioning	How and when are node resources partitioned?	Dynamic LWK image selection during operation
Application Perspective	POSIX compatibility	What is the level of POSIX support on the LWK?	Wide range applications support
	Linux pseudo file system support	Is the Linux pseudo file system visible and fully supported on the LWK side?	
	Access method to Linux functionality	How does an application access Linux functionality?	Execution time of Linux-based applications
	Syscall overhead	What is the system call overhead?	
	Shared memory between the two kernels	Can an LWK and a Linux process share memory?	
	Multi-kernel processes	Can a single process with multiple threads span Linux and the LWK?	
	NUMA support	Does the LWK support NUMA architectures?	Manycore support
	Performance isolation	How is Linux limited from interfering with the LWK	Reproducable high performance environment

Defining Characteristics and Criteria

	Property	Short Description	Impact
System Administrator Perspective	Standalone LWK	Is the LWK a separate binary from Linux, and does it boot the cores it runs on?	
	Node boot	Which kernel is booted by the BIOS/Firmware of the node?	
	Resource partitioning	**How and when are node resources partitioned?**	**Dynamic LWK image selection during operation**
Application Perspective	POSIX	What is the level of POSIX	
	Linux syster		
	Acces functic		
	Syscal		
	Shared memory between the two kernels	Can an LWK and a Linux process share memory?	
	Multi-kernel processes	Can a single process with multiple threads span Linux and the LWK?	
	NUMA support	Does the LWK support NUMA architectures?	Manycore support
	Performance isolation	How is Linux limited from interfering with the LWK	Reproducable high performance environment

- Through five or six year supercomputer operation, several LWKs will be available
- Some users want to use the latest one or a special version of LWK, but some other users want to use the original LWK
- Dynamic LWK image selection enables the users to select one of LWKs without rebooting compute nodes

Defining Characteristics and Criteria

	FusedOS	IHK/McKernel	mOS	Pisces/Kitten	Kitten/Palacios	FFMK (L4)
Resource partitioning	Static (Late)	Dynamic (Late)	Static (Early)	Dynamic (Late)	Dynamic (Late)	Dynamic (Late)

System Administrator Perspective	Node boot	Which kernel is booted by the BIOS/Firmware of the node?	
	Resource partitioning	How and when are node resources partitioned?	Dynamic LWK image selection during operation
Application Perspective	POSIX	What is the level of POSIX	
	Linux system		
	Acces functio		
	Syscal		
	Shared memory between the two kernels	Can an LWK and a Linux process share memory?	
	Multi-kernel processes	Can a single process with multiple threads span Linux and the LWK?	
	NUMA support	Does the LWK support NUMA architectures?	Manycore support
	Performance isolation	How is Linux limited from interfering with the LWK	Reproducable high performance environment

- Through five or six year supercomputer operation, several LWKs will be available
- Some users want to use the latest one or a special version of LWK, but some other users want to use the original LWK
- Dynamic LWK image selection enables the users to select one of LWKs without rebooting compute nodes

Defining Characteristics and Criteria

	Property	Short Description	Impact
System Administrator Perspective	Standalone LWK	Is the LWK a separate binary from Linux, and does it boot the cores it runs on?	
	Node boot	Which kernel is booted by the BIOS/Firmware of the node?	
	Resource partitioning	How and when are node resources partitioned?	Dynamic LWK image selection during operation
	POSIX compatibility	What is the level of POSIX support on the LWK?	Wide range applications support
	Linux pseudo file system support	Is the Linux pseudo file system visible and fully supported on the LWK side?	
Perspective	Shared memory between the two kernels	Can an LWK and a Linux process share memory?	
	Multi-kernel processes	Can a single process with multiple threads span Linux and the LWK?	
	NUMA support	Does the LWK support NUMA architectures?	Manycore support
	Performance isolation	How is Linux limited from interfering with the LWK	Reproducable high performance environment

	FusedOS	IHK/McKernel	mOS	Pisces/Kitten	Kitten/Palacios	FFMK (L4)
POSIX compatibility on LWK	Yes	Yes	Yes	No	No	Yes
Pseudo file system	No	Mostly	Yes	No	No	No

Defining Characteristics and Criteria

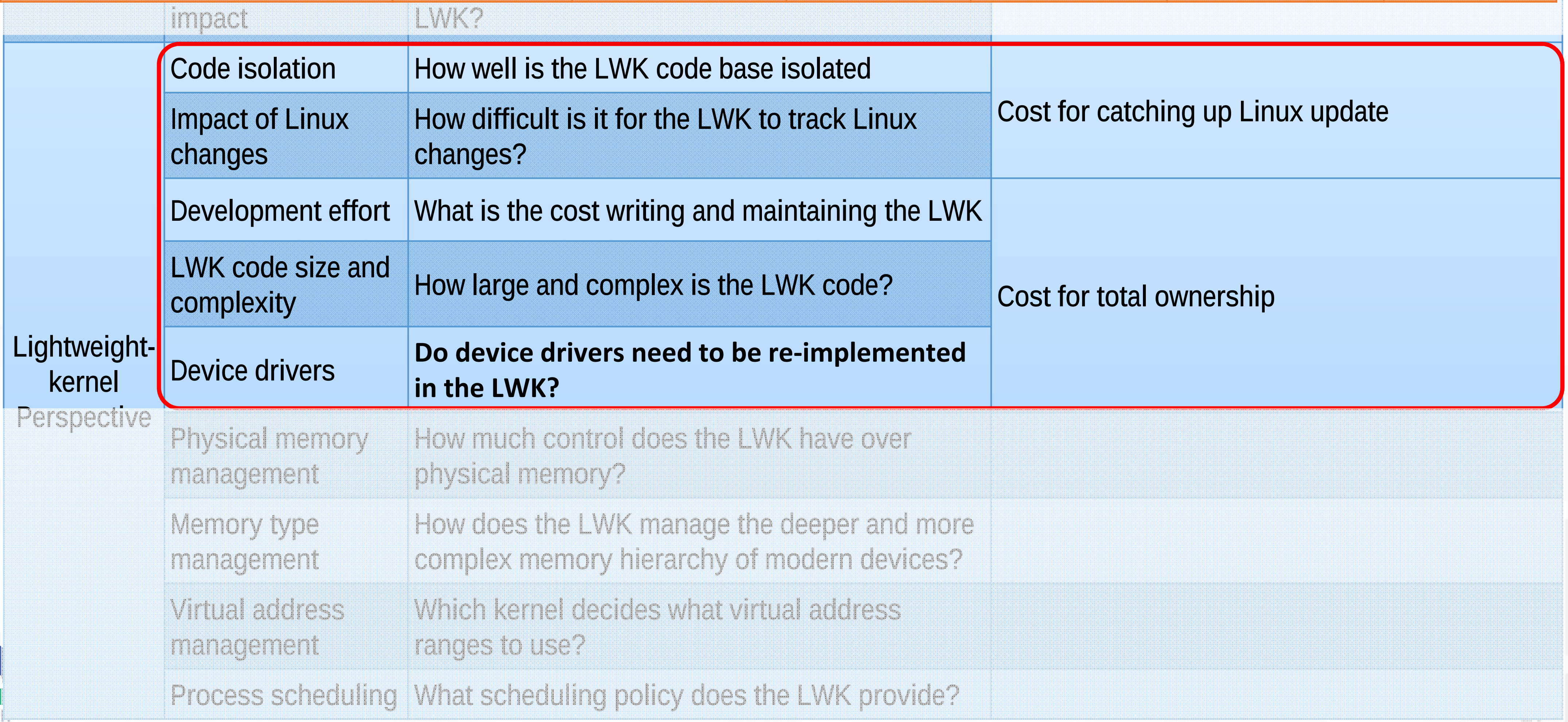
Property		Short Description	Impact
System Administrator Perspective	Standalone LWK	Is the LWK a separate binary from Linux, and does it boot the cores it runs on?	
	Node boot	Which kernel is booted by the BIOS/Firmware of the node?	
	Resource partitioning	How and when are node resources partitioned?	Dynamic LWK image selection during operation
Application Perspective	POSIX compatibility	What is the level of POSIX support on the LWK?	Wide range applications support
	Linux pseudo file system support	Is the Linux pseudo file system visible and fully supported on the LWK side?	
	Access method to Linux functionality	How does an application access Linux functionality?	Execution time of Linux-based applications
	Syscall overhead	What is the system call overhead?	
	Shared memory between the two kernels	Can an LWK and a Linux process share memory?	
	Multi-kernel processes	Can a single process with multiple threads span Linux and the LWK?	
	NUMA support	Does the LWK support NUMA architectures?	Manycore support
	Performance isolation	How is Linux limited from interfering with the LWK	Reproducable high performance environment

Defining Characteristics and Criteria

Property		Short Description	Impact
System Administrator Perspective	Standalone LWK	Is the LWK a separate binary from Linux, and does it boot the cores it runs on?	
	Node boot	Which kernel is booted by the BIOS/Firmware of the node?	
	Resource partitioning	How and when are node resources	Dynamic LWK image selection during
	system support	and fully supported on the LWK side?	
Application Perspective	Access method to Linux functionality	How does an application access Linux functionality?	Execution time of Linux-based applications
	Syscall overhead	What is the system call overhead?	
	Shared memory between the two kernels	Can an LWK and a Linux process share memory?	
	Multi-kernel processes	Can a single process with multiple threads span Linux and the LWK?	
	NUMA support	Does the LWK support NUMA architectures?	Manycore support
	Performance isolation	How is Linux limited from interfering with the LWK	Reproducable high performance environment

	FusedOS	IHK/McKernel	mOS	Pisces/Kitten	Kitten/Palacios	FFMK (L4)
Access method to Linux features	**Proxy**	**Proxy**	**Migrate**	**No**	**No**	**Migrate**
Linux sys call overhead	**High**	**High**	**High**	**-**	**-**	**High**

	FusedOS	IHK/McKernel	mOS	Pisces/Kitten	Kitten/Palacios	FFMK (L4)
Isolated LWK code base	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**
Impact of Linux changes	**Minimal**	**Minimal**	**Code merge**	**Minimal**	**Minimal**	**L4Linux port**
Development effort	**Small**	**Significant**	**Ideally small**	**Significant**	**Significant**	**Significant**
Code size (kLOC) *	**150**	**65**	**12**	**213 (Kitten+Pisces+Palacios)**		**32**
Device driver transparency	**No**	**Yes**	**Yes**	**No**	**No**	**No**



	impact	LWK?	
Lightweight-kernel Perspective	Code isolation	How well is the LWK code base isolated	Cost for catching up Linux update
	Impact of Linux changes	How difficult is it for the LWK to track Linux changes?	
	Development effort	What is the cost writing and maintaining the LWK	Cost for total ownership
	LWK code size and complexity	How large and complex is the LWK code?	
	Device drivers	**Do device drivers need to be re-implemented in the LWK?**	
	Physical memory management	How much control does the LWK have over physical memory?	
	Memory type management	How does the LWK manage the deeper and more complex memory hierarchy of modern devices?	
	Virtual address management	Which kernel decides what virtual address ranges to use?	
	Process scheduling	What scheduling policy does the LWK provide?	

Summary

- The multi-kernel OS approach is promising for addressing challenges at extreme scale HPC
- Multiple projects exploring the field
- We compiled their fundamental properties and defining characteristics
- Established a set of criteria
- Mapped each project onto these criteria and provided a comparison among them