CCIX
SEAMLESS ACCELERATION

Memory Expansion and Storage Acceleration with CCIX Technology

Millind Mittal, Fellow, Xilinx

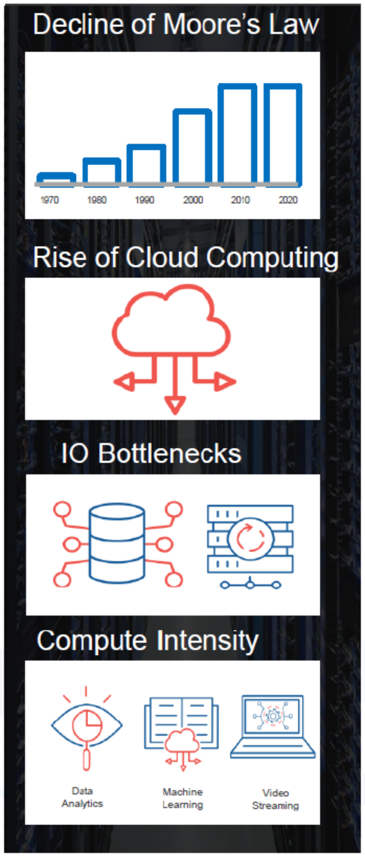
Jason Lawley, DC Platform Architect, Xilinx

Agenda

- Brief introduction to CCIX
- Memory Expansion through CCIX
- Persistent Memory support
- Storage with Compute offload
- Q&A

CCIX Context

- Slow down of performance scaling and efficient of general purpose processors
- Increasing “workload specific” computation requirements
  - Data analytics, 400G, ML, Security, compression, ……
- Lower latency requirements
  - cloud based services, IoT, 5G, …..
- Need for open standard for advancing IO Interconnect to enable seamless expansion of compute and memory resources
  - Enable accelerator SoCs to be like a NUMA sockets from Data Sharing perspective



Decline of Moore’s Law

Rise of Cloud Computing

IO Bottlenecks

Compute Intensity

The CCIX Consortium

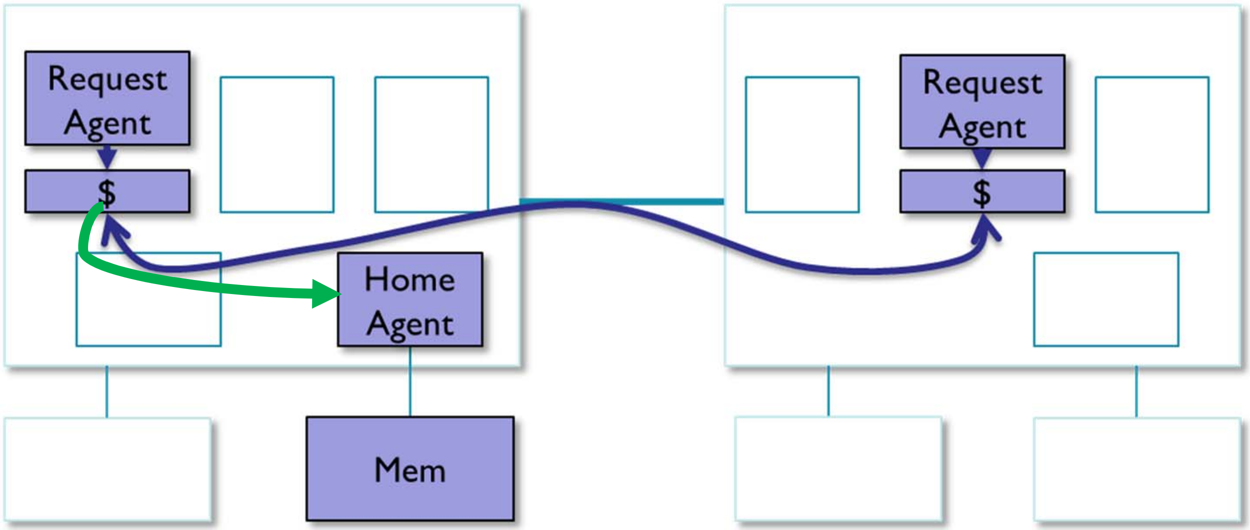
- 53 Members covering all aspects of ecosystems; Servers, CPU/SoC, Accelerators, OS, IP/NoC, Switch, Memory/SCM, Test & Measurement vendors.
- Specification Status
  - Rev 1.0 - 2018
  - Rev 1.1/Rev1.2 – 2019
  - SW Guide Rev 1.0- Sept, 19
- CCIX Hosts:
  - ARM 7nm test Processor SoC providing CCIX interface (N1SDP)
  - Huawei announced Kunpeng 920
  - A 3rd party ARM SoC, Sample 12/19
- CCIX Accelerator / EP
  - Xilinx VU3xP family
  - Alveo boards (U50 and U280) available
  - 7nm chip Versal with CCIX support announced
- SW Enablement
  - In progress ; Key enablement to be completed Sept, 19

Promotors
AMD
arm
HUAWEI
Mellanox
QUALCOMM
XILINX
Contributors
AMPERE
Amphenol
avery design systems
cādence
CRAY
FUJITSU
GUC
IBM
IDT
iol InterOperability Laboratory
KEYSIGHT TECHNOLOGIES
Lenovo
Linaro
LUXSHARE ICT
MARVELL
Micron
Microsoft
redhat
SAMSUNG
samtec
SK hynix
SYNOPSYS
TE
Tektronix
TELEDYNE LECROY Everywhereyoulook
TOSHIBA MEMORY
tsmc
Western Digital

Adopters

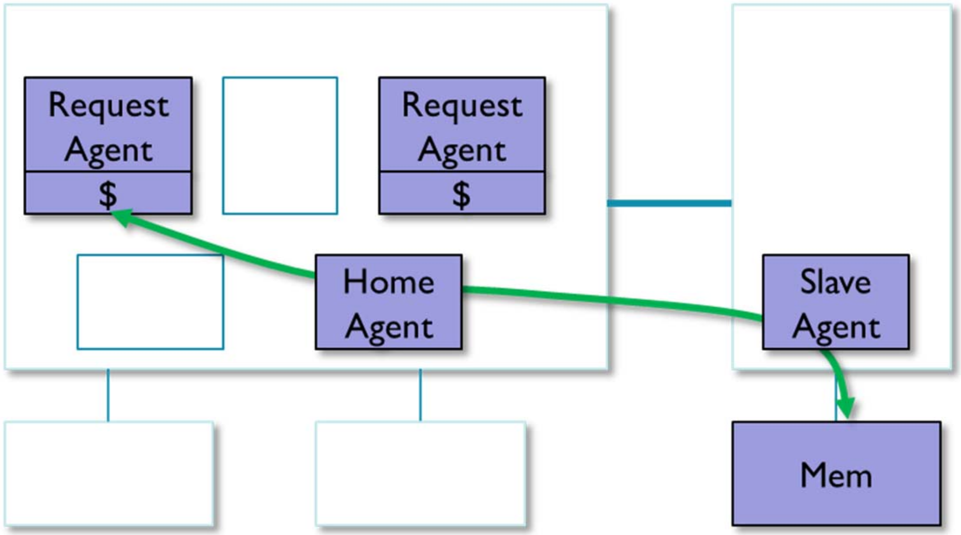
Alibaba (China) Co., Ltd. ∞ Arteris, Inc. ∞ Baikal Electronics ∞ Bitman Technologies, Inc.
Chengdu Higon Integrated Circuit ∞ Data Vortex Technologies ∞ Design Co., Ltd. ∞ Ericsson AB
Iluvatar CoreX Inc. Nanjing Netlist, Inc. ∞ Nokia Solutions and Networks ∞ Phytium Technology Co., Ltd.
Mentor. A Siemens Business ∞ PLDA ∞ SAFARI Research Group at ETH Zurich
Shanghai Zhaoxin Semiconductor Co., Ltd. ∞ SmartDV Technologies India Private Ltd.
Socionext Inc. ∞ Sony Semiconductor Solutions Corporation
Technische Universität (TU) Darmstadt ∞ Viavi Solutions, Inc. ∞ VMware, Inc.

Use of Caches for System Performance



Role of Slave Agent

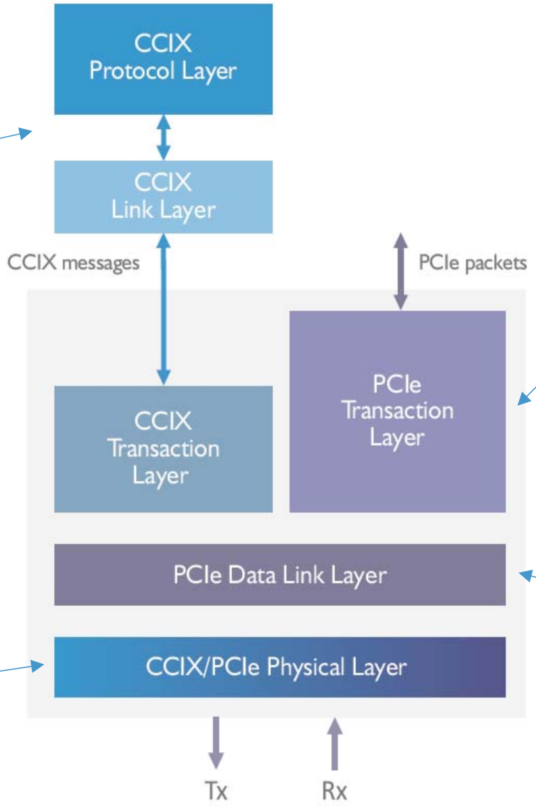
- Slave Agent provides additional memory to a Home Agent
- Slave Agent is only protocol visible when residing on a different chip



CCIX -Transport and Layered Architecture

CCIX Protocol Layer

- Responsible for the coherency including memory read and write flows
- CCIX Link Layer
- Responsible for formatting CCIX traffic for the target transport and non-blocking behavior between two CCIX devices
- Currently PCIe but could be mapped over a different transport layer in the future



CCIX and PCIe Transaction Layers

- Responsible for handling their respective packets
- PCIe & CCIX packets are split across virtual channels (VCs) sharing same link
- Optimized CCIX packets: Eliminates the PCIe overhead

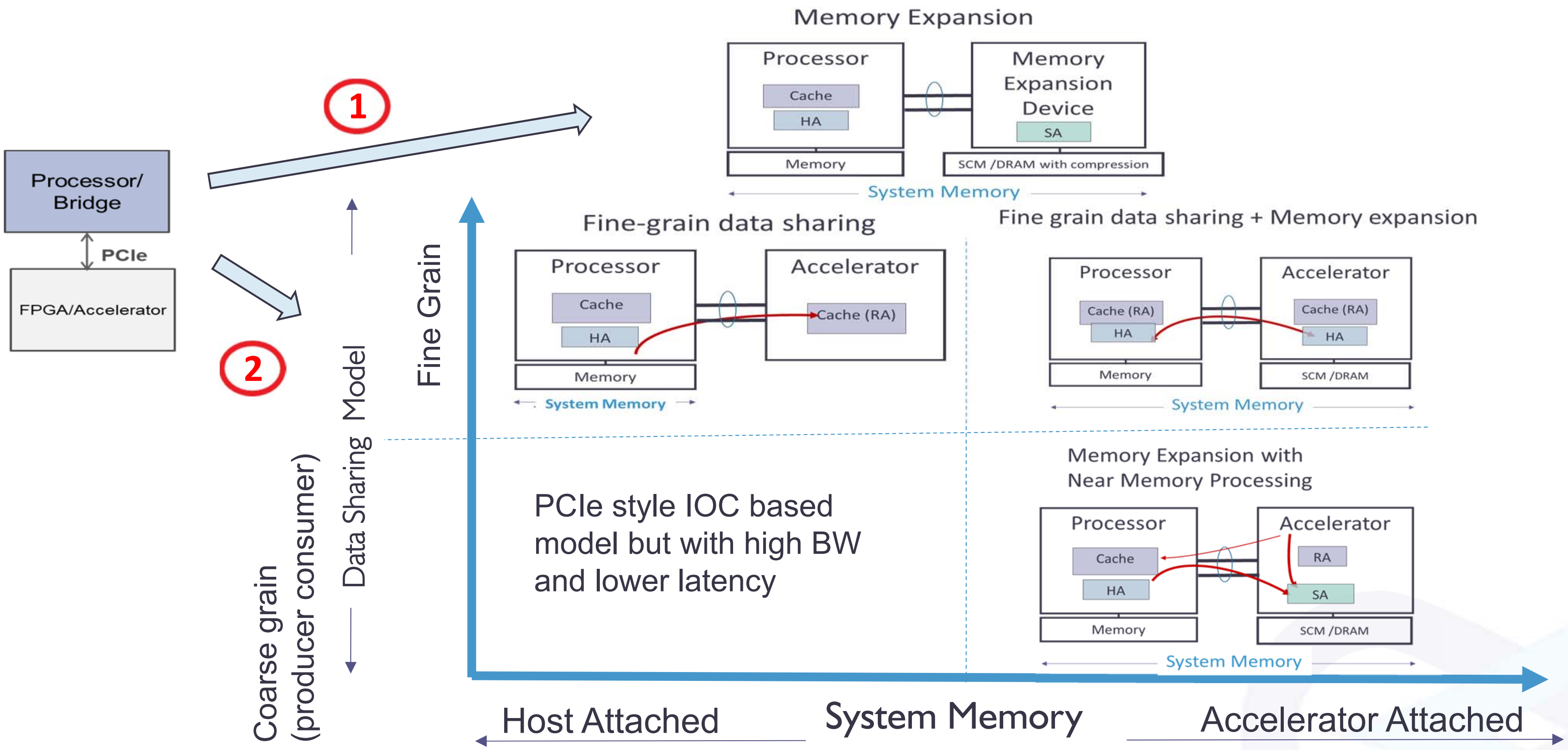
PCIe Data Link Layer

- Performs normal functions of the data link layer

CCIX/PCIe Physical Layer

- Faster speed, known as ESM (Extended Speed Mode)

CCIX – Open Standard Memory Expansion and Fine-Grain Data Sharing Model with Accelerators

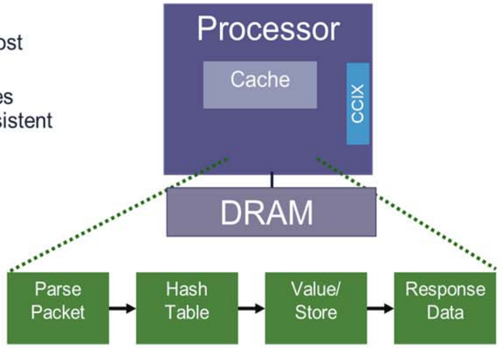


PCIe style IOC based model but with high BW and lower latency

Enabling Seamless Expansion of Compute and Memory Resources – Accelerator SoCs are seen as NUMA Socket

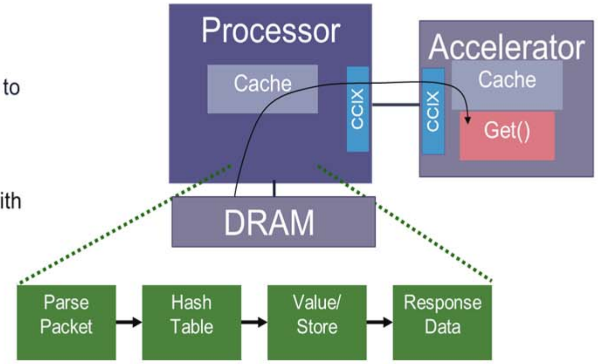
KVS Database with Host-Only Processing (Default)

- All operations works out of host memory
- Adding persistence to updates requires additional IO to Persistent storage, e.g. NVMe



KVS Database with Processing Acceleration

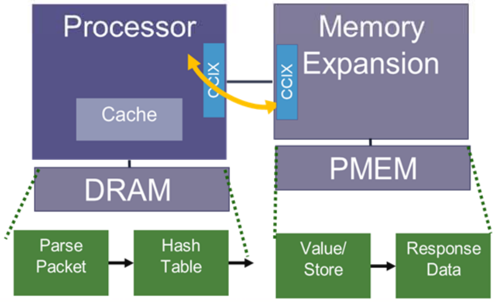
- No disruption to Networking
- Control processing (Set, Delete, etc) remain on host processor
- Fast path operations (Get) move to accelerator
- CCIX enabled shared data structures, no copies
- Increase throughput multi-gets with almost no increase in CPU utilization



KVS Database with Memory Expansion using CCIX to connect to Persistent Memory (PMEM)

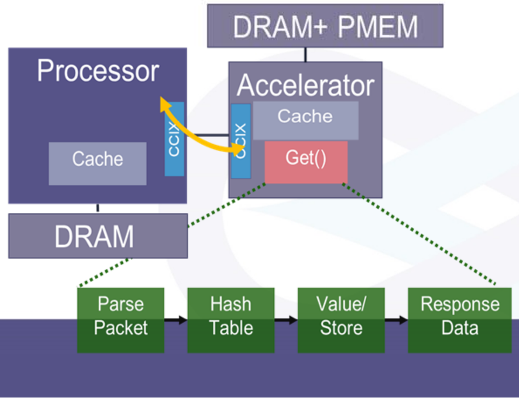
- Supports larger databases than DDR
- Eliminates filesystem processing for log/checkpoint work
- No risk of data loss even though there is no explicit back-up done
- Instantaneous restarts

Application works directly on PMEM



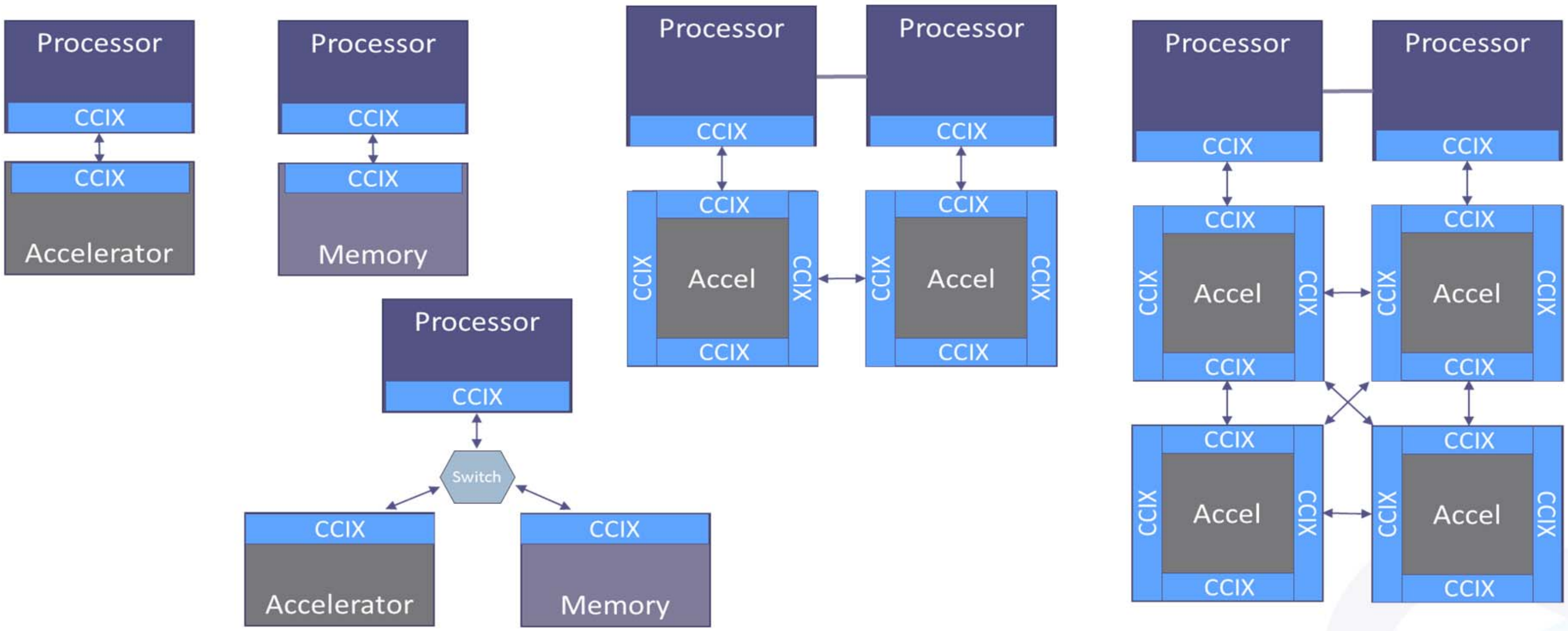
KVS Database with Memory Expansion Plus Acceleration

Combined Benefit of Memory Expansion with PMEM and Processing Offload



CCIX - Flexible Topologies

Direct attached, daisy chain, mesh and switched topologies



SW enablement in progress

- ACPI 6.3 and UEFI 2.8 enhancements for CCIX
  - Specific-purpose Memory
  - Generic Initiator Affinity Structure and associated _OSC bit
  - HMAT Table Enhancements
  - New CPER record for CCIX
- Ongoing Reference Code Implementation jointly done by Linaro, Arm and other members
  - Mail list ccix@linaro.org
  - JIRA Initiative https://projects.linaro.org/browse/LDCG-713
  - Work presented at Linaro Connect BKK19 in April 2019
  - UEFI Firmware code is available as part of project

Memory Expansion Through CCIX

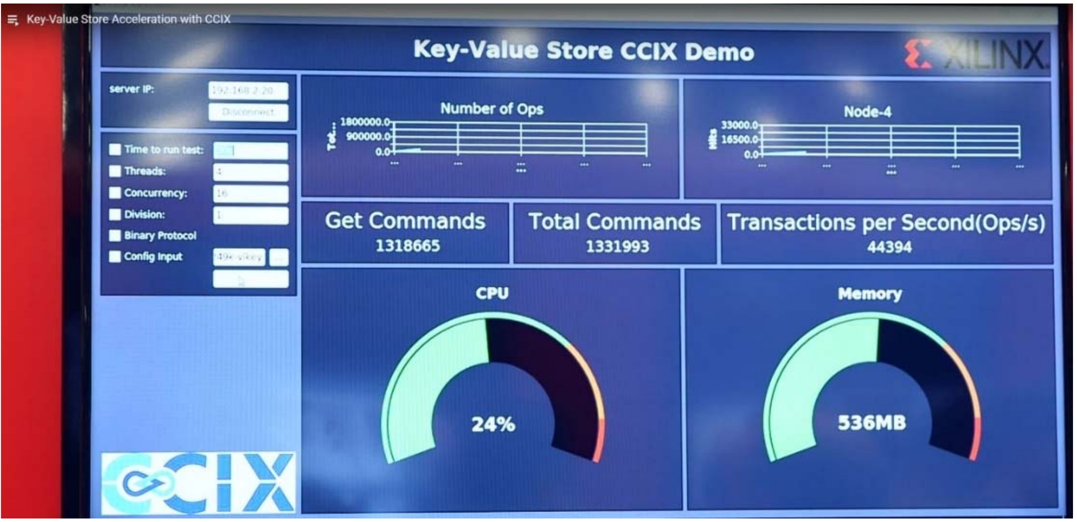
Memory Expansion Through NUMA

Demonstrated Extended memory through NUMA over CCIX at SC18

KVS Database (Memcached) was enhanced to make use of NUMA expansion model over CCIX

Key allocations are done in Host DDR, where as corresponding values were allocated on remote FPGA memory

Expansion memory can also be a persistent memory connected over CCIX link



Key-Value Store Acceleration with CCIX

Key-Value Store CCIX Demo

server IP: 192.168.2.20

□ Time to run test:

□ Threads: 4

□ Concurrency: 16

□ Division: 1

□ Binary Protocol

□ Config Input 49+.vkey

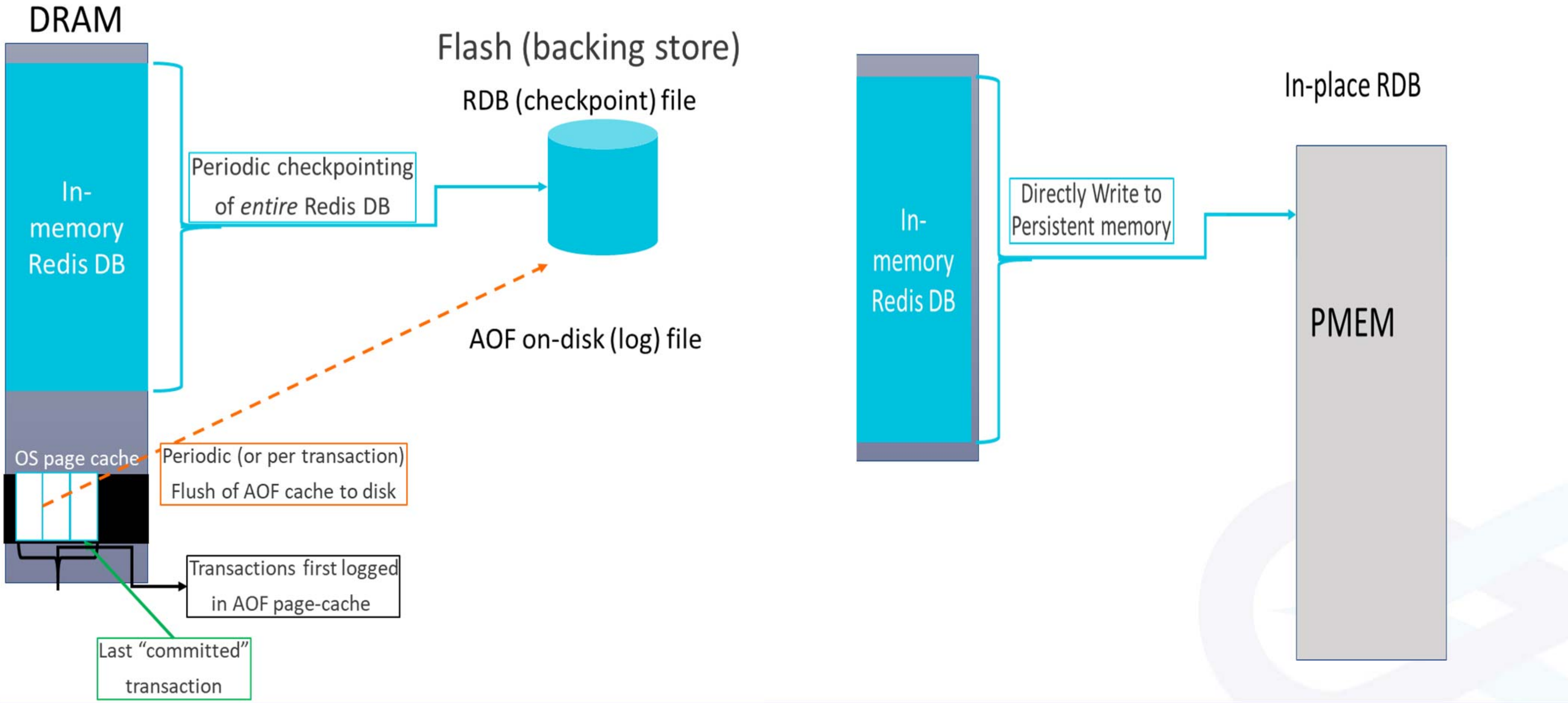
Get Commands 1318665

Total Commands 1331993

Transactions per Second(Ops/s) 44394

https://www.youtube.com/watch?v=drIu4vlubxE&list=PLRr5m7hDN9TLI3vuw1OqLbF7YcGi3UO9c&index=9

Redis with Persistent Memory support

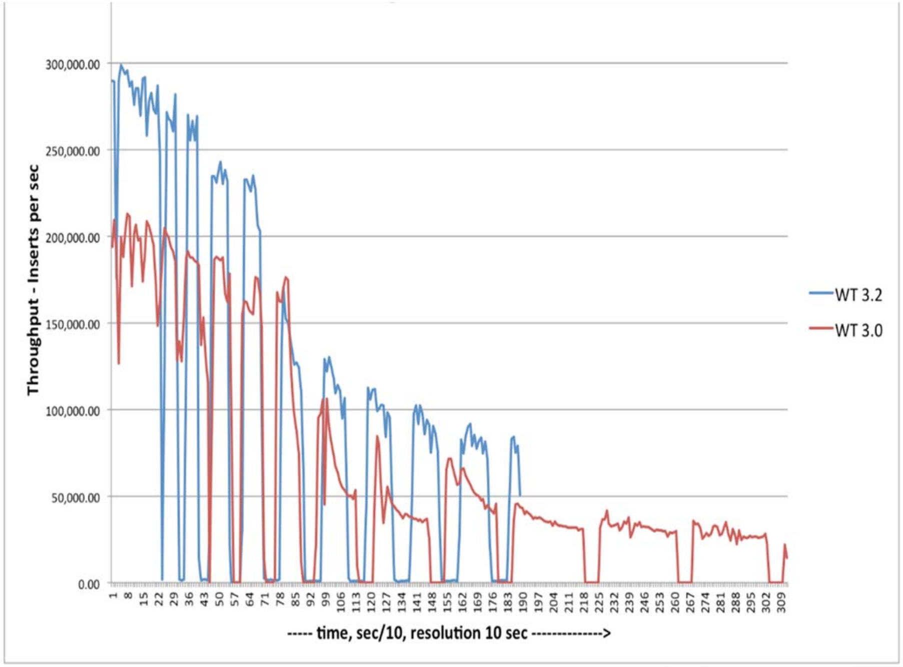

DRAM
In-memory Redis DB
OS page cache
Flash (backing store)
RDB (checkpoint) file
Periodic checkpointing of *entire* Redis DB
AOF on-disk (log) file
Periodic (or per transaction) Flush of AOF cache to disk
Transactions first logged in AOF page-cache
Last "committed" transaction
In-memory Redis DB
Directly Write to Persistent memory
In-place RDB
PMEM

Storage with Compute Offload

Analysis and Inference

- WiredTiger is an performance, scalable, production quality, NoSQL, Open Source extensible platform for data management

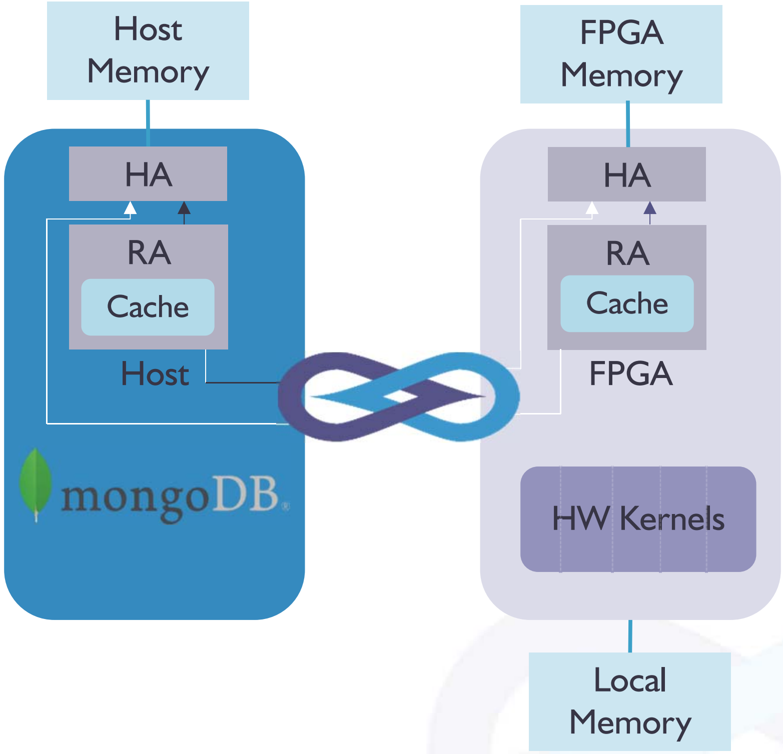
- Run two performance bench marking tests & collected call stacks
  - https://github.com/johnlpage/POCDriver
  - https://github.com/mdcallag/iibench-mongodb
- Major hot spots were identified as
  - WiredTiger IO operations (IO intense)
  - Compression (CPU intense)



WiredTiger Storage Engine (http://source.wiredtiger.com/

Accelerated Design Over CCIX

- IOPs are limited due to OS context switch and other SW overheads
- Enable user space calls to FS directly
- Offload performance critical operations (writes/reads) fully to FPGA with interface to storage
  - File system Meta data structures are maintained in shared FPGA memory
  - Actual file data is stored over FPGA connected storage class memory which is faster than SSDs
- Inline efficient Compression
- Seamless acceleration architecture through shared meta-data enabled by CCIX

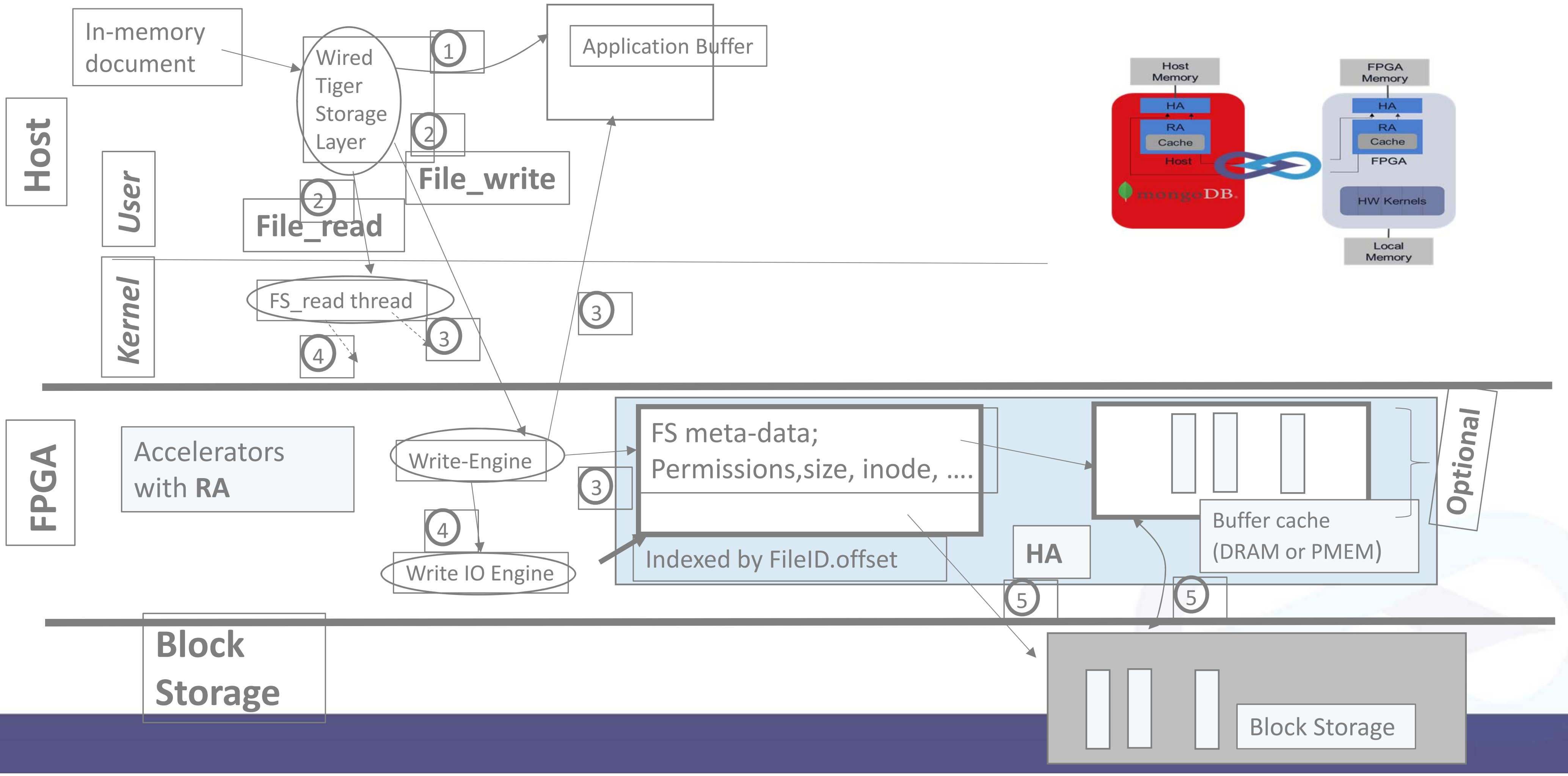


Split File System Operation Distribution Between Host & FPGA

- Instead of full file system offload we propose a split file system with Metadata share over CCIX interface
- CPU Handled operations:
  - fs_open – Creates new file or reopens the existing file
  - fs_exist – Checks whether the file exists
  - fs_rename – Renames existing file
  - fs_terminate – closes the file system
  - fs_create – creates the file system
  - file_size – Returns the file size
  - file_close – closes the file
  - file_truncate – truncates the file to the specified size
  - fs_read – Reads a data block from file
- All these operations need not be sent to FPGA as these can read/edit the shared structures
- Only handle fs_write in FPGA with the focus to achieve accelerated performance for Writes.
  - Be able to ingest the data into NoSQL DBs like MongoDB.

SC19 processing flow

Without data compression



Host

User

Kernel

FPGA

In-memory document

Wired Tiger Storage Layer

Application Buffer

File_write

File_read

FS_read thread

Accelerators with RA

Write-Engine

Write IO Engine

FS meta-data;
Permissions,size, inode, ….

Indexed by FileID.offset

HA

Buffer cache
(DRAM or PMEM)

Optional

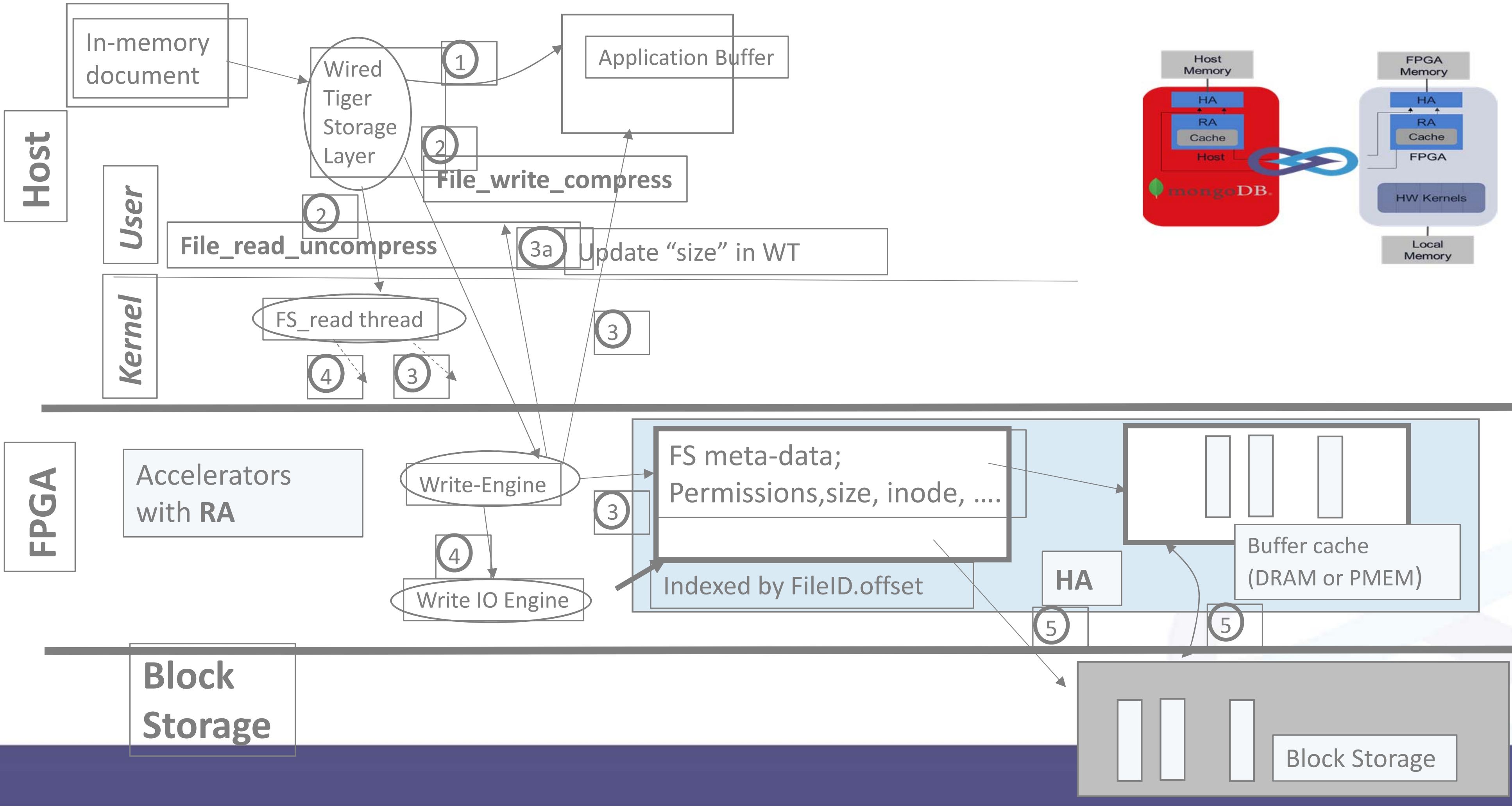
Block Storage

Block Storage

1 2 2 3 3 3 3 4 4 5 5

SC19 processing flow

With data compression



Host

User

Kernel

FPGA

In-memory document

Wired Tiger Storage Layer

1

Application Buffer

2

File_write_compress

2

File_read_uncompress

3a

Update “size” in WT

FS_read thread

4

3

3

Accelerators with **RA**

Write-Engine

3

4

Write IO Engine

FS meta-data; Permissions,size, inode, ….

Indexed by FileID.offset

HA

Buffer cache (DRAM or PMEM)

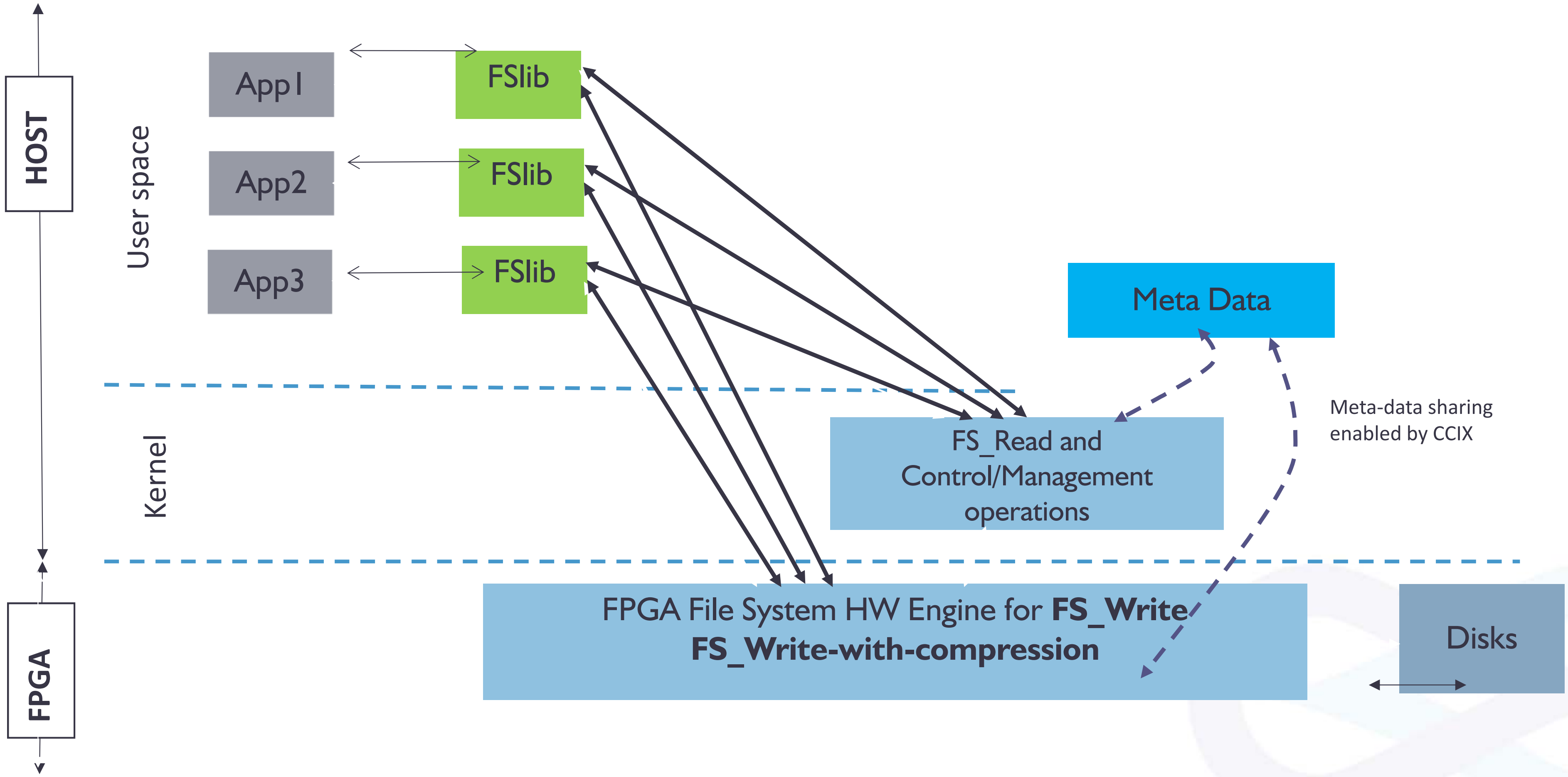
5

5

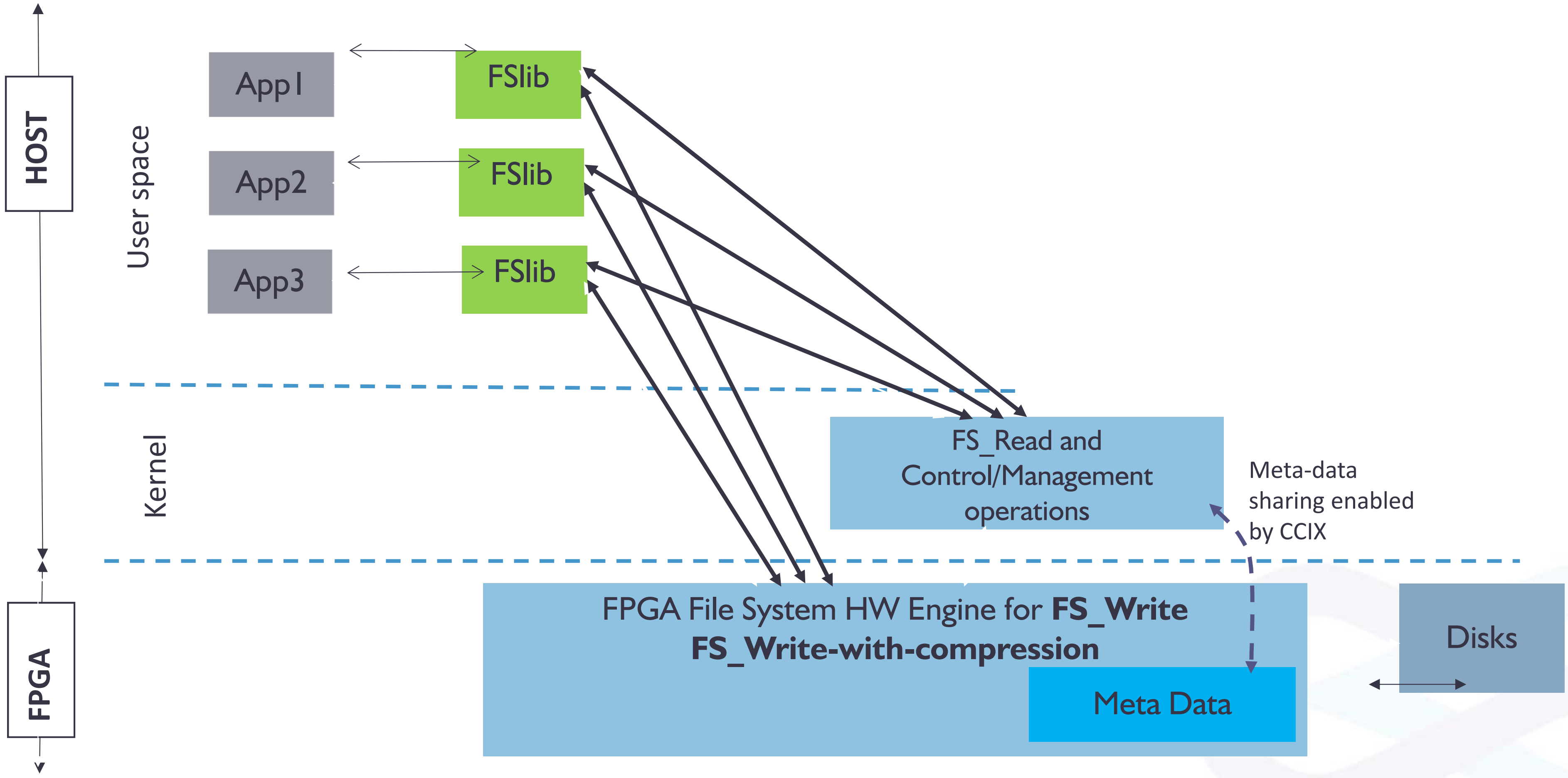
Block Storage

Block Storage

Split File System Operation Distribution Between Host & FPGA


HOST
User space
App1
FSlib
App2
FSlib
App3
FSlib
Meta Data
Kernel
FS_Read and Control/Management operations
Meta-data sharing enabled by CCIX
FPGA
FPGA File System HW Engine for **FS_Write FS_Write-with-compression**
Disks

Meta-data in the FPGA Attached Memory


HOST
FPGA
User space
Kernel
App1
App2
App3
FSlib
FSlib
FSlib
FS_Read and Control/Management operations
Meta-data sharing enabled by CCIX
FPGA File System HW Engine for **FS_Write**
FS_Write-with-compression
Meta Data
Disks

Current PoCs underway

- Storage layer acceleration
  - PMDK framework enablement for ARM processors for SCM
  - Write IO-Ops acceleration for MongoDB ← Show case at SC19
- Memory expansion on Xilinx Versal device ← XDF 19

Summary

- CCIX enables new platform level capability to enable accelerated solutions for storage and other verticals
- CCIX technology is ready to develop PoCs and products
- Contact below to learn more

https://www.ccixconsortium.com/ or

You can contact me at millind@Xilinx.com