Memory Expansion and Storage Acceleration with CCIX Technology

Ravi Kiran Gummaluri

Xilinx

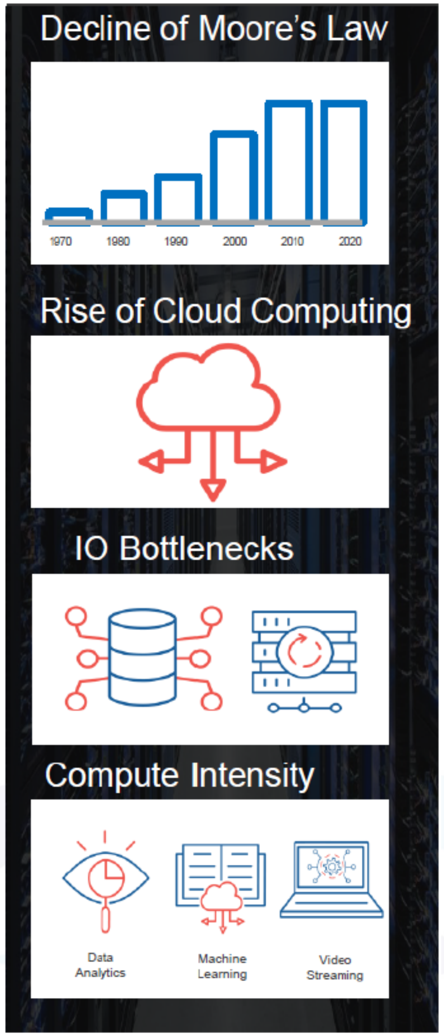
Agenda

- Brief Introduction to CCIX
- Memory Expansion Through CCIX
- Storage with Compute Offload
- Supporting Persistent Memory
- Q&A

Brief Introduction to CCIX

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 beyond processor SoCs with the focus on high BW, low latency and ease of use
  - Enable accelerator SoCs to be like a NUMA sockets from Data Sharing perspective



Flash Memory Summit

The CCIX Consortium

- Incorporated in 2016
- 50+ Members covering all aspects of ecosystems:
  - Servers, CPU/SoC, Accelerators, OS, IP/NoC, Switch, Memory/SCM, Test & Measurement vendors
- CCIX Hosts:
  - Arm/Cadence/Xilinx collaboration – A 7nm test Processor SoC providing CCIX interface
  - Huawei announced Kunpeng 920
  - Other processors under development
- CCIX Accelerator / EP:
  - Xilinx VU3xP family CCIX-enabled FPGAs silicon available
  - Versal with CCIX support announced

Promotors

AMD, arm, HUAWEI, Mellanox Technologies, Qualcomm, XILINX

Contributors

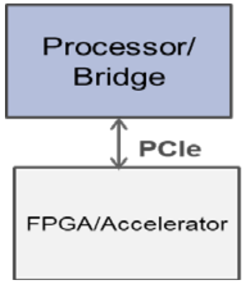
AMPERE, Amphenol, avery design systems, cadence, CRAY, FUJITSU, GUC, IBM, IDT, iol InterOperability Laboratory, KEYSIGHT TECHNOLOGIES, Lenovo, Linaro, LUXSHARE ICT, MARVELL, Micron, Microsoft, redhat, SAMSUNG, samtec, SK hynix, SYNOPSYS, TE, Tektronix, TELEDYNE LECROY Everywhereyoulook, TOSHIBA MEMORY, tsmc, Western Digital

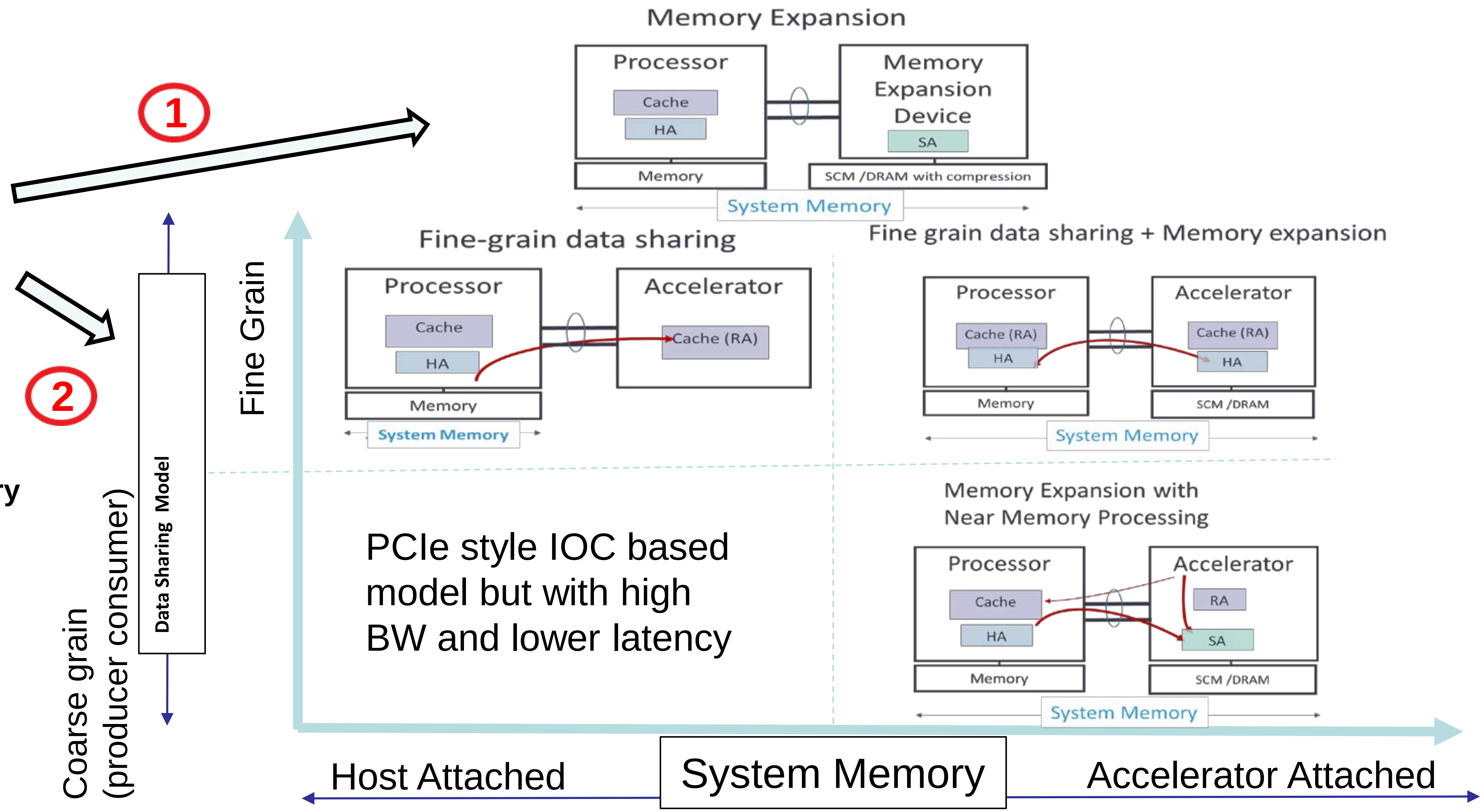
Adopters

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.

Flash Memory Summit

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





1. **Meta data in expansion memory**
2. **Peer processing**

RA: Request Agent (Initiator + cache)

HA: Home Agent

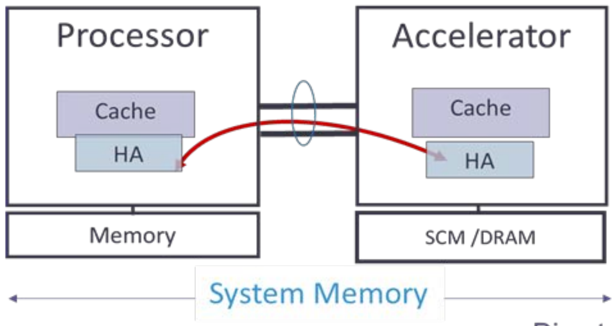
SA: Slave Agent

#CCIX #MemoryExpansion

CCIX - Key New Attributes

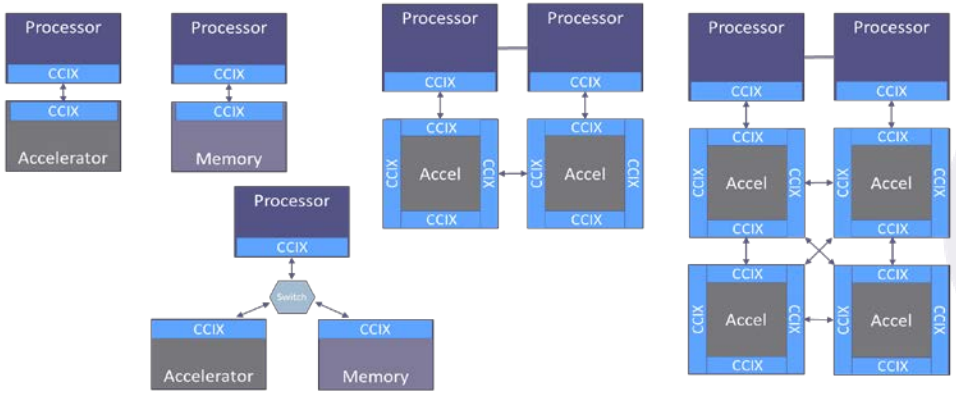
1 Memory expansion and new data sharing models including fine-grain peer processing

Memory Expansion + Fine grain data sharing



2 Flexible topologies

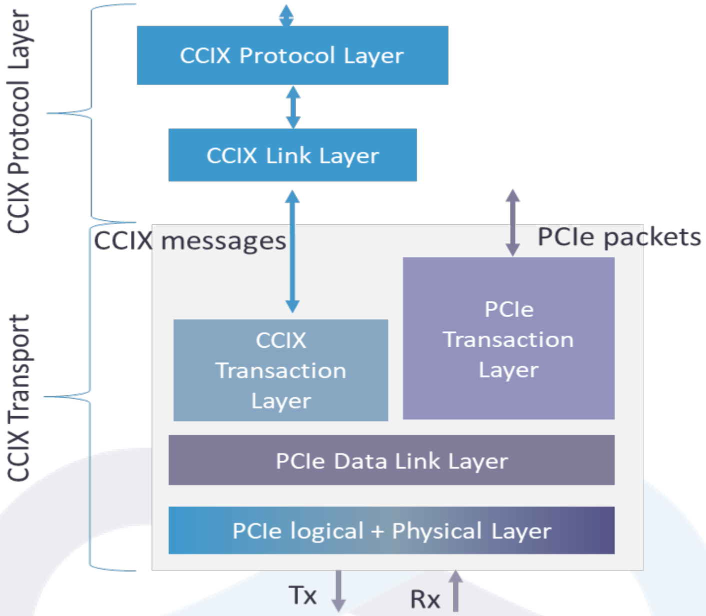
Direct attached, daisy chain, mesh and switched topologies



3 Layered architecture model

- Support different transports in future

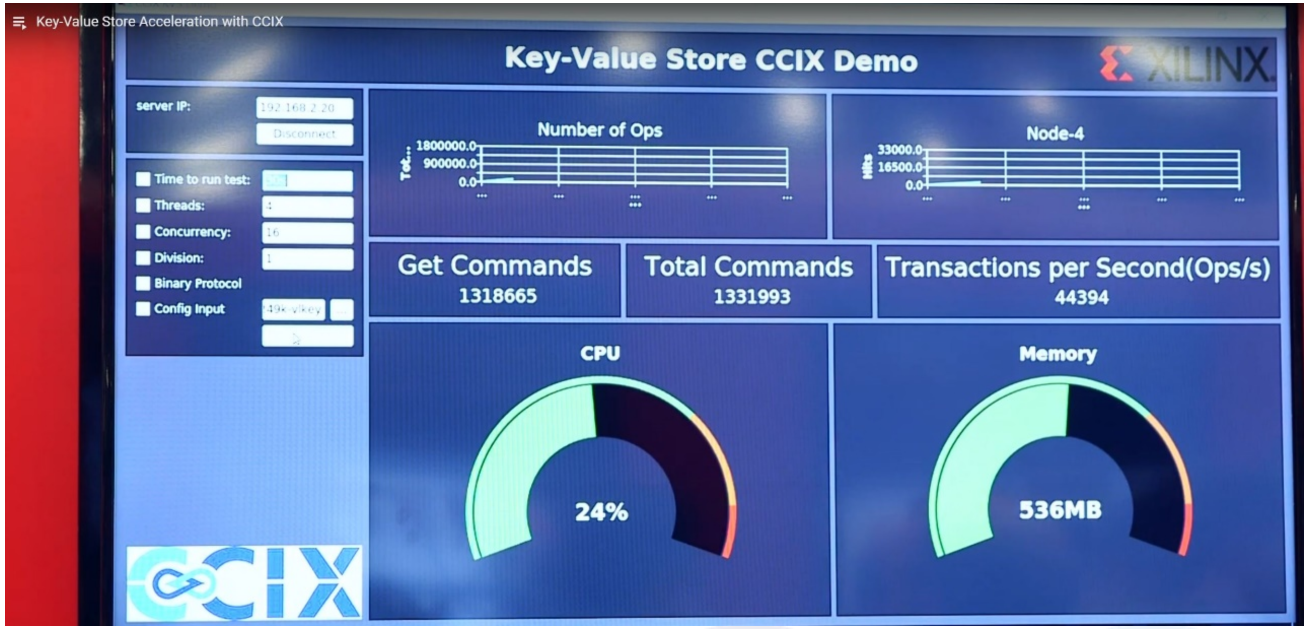
CCIX Architecture Model



Memory Expansion Through CCIX

Memory Expansion Through NUMA

- Demonstrated Extended memory through NUMA over CCIX at Super Computing 2018
- 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



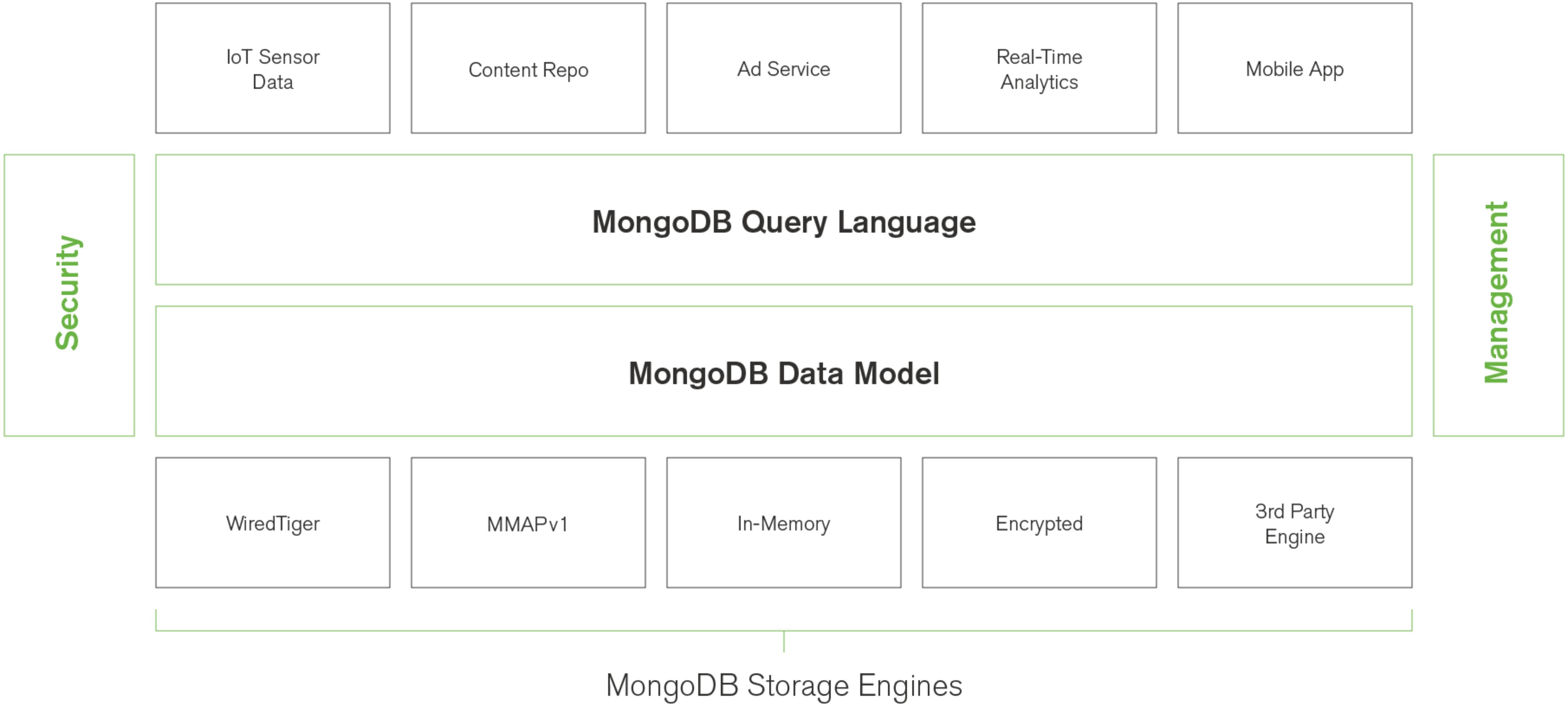
https://www.youtube.com/watch?v=drIu4vlubxE&list=PLRr5m7hDN9TLI3vuw1OqLbF7YcGi3UO9c&index=9

Storage with Compute Offload

Flash Memory Summit

mongoDB®

“MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need”



https://www.mongodb.com/ Figure from MongoDB Architecture Guide

Analysis and Inference

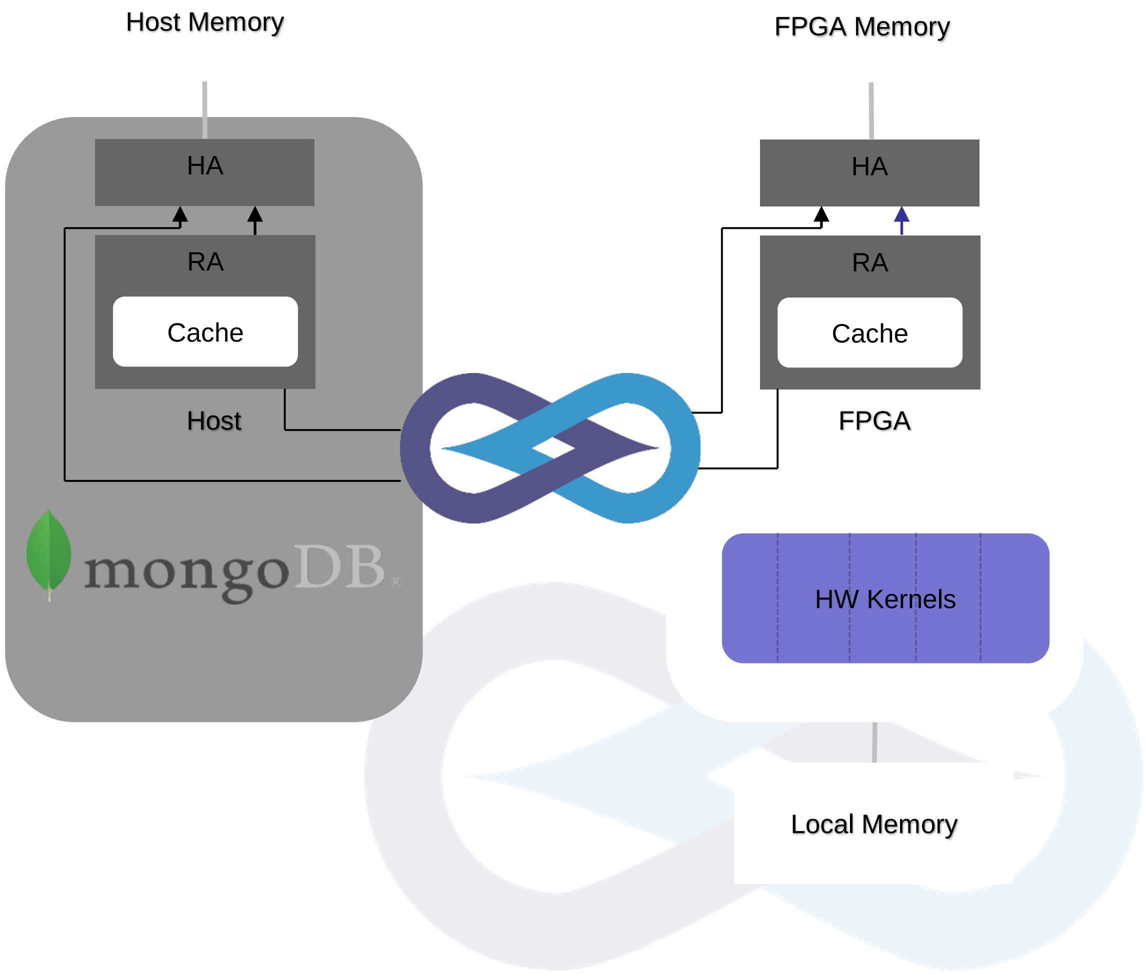
- Run two performance bench marking tests & collected call stacks
  - https://github.com/johnlpage/POCDriver
  - https://github.com/mdcallag/iibench-mongodb
- Major hot spots were identified as
  - Compression (CPU intense)
  - WiredTiger IO operations (IO intense)

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

- WiredTiger is an performance, scalable, production quality, NoSQL, Open Source extensible platform for data management
- Starting in MongoDB 3.2, the WiredTiger storage engine is the default storage engine
- Provides extensions for custom implementations of compression, encryption, File System etc.

Accelerated Design Over CCIX

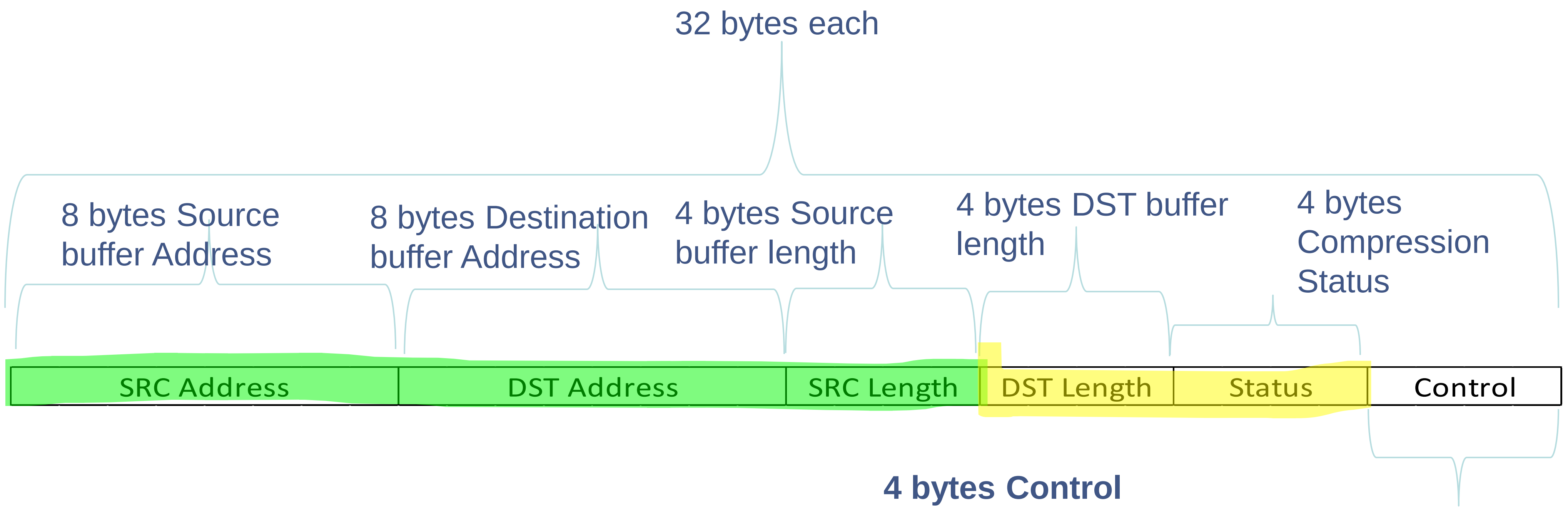
- An efficient Compression algorithm is implemented in HW kernel
- Split File system implementation with critical operations managed by FPGA
  - File system Meta data structures are maintained in shared FPGA memory
  - Actual file data is stored over FPGA connected storage class memory which is faster than SSDs
- Application interacts with file manager through custom descriptors shared over CCIX
- Seamless acceleration architecture through SVM.



Split File System Operation Distribution Between CPU & 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
- All these operations need not be sent to FPGA as these can read/edit the shared structures
- In total, for a 5 minute WT performance run these functions were called for around 150 times
- These functions need not be offloaded
- Can be implemented in CPU only when shared memory model is enabled
- FPGA Handled operations:
  - fs_read – Reads a data block from file
  - fs_write – writes a data block to file
- In total, for a 5 minute WT performance there are ~1565000 reads+writes

Custom Shared Structure Over CCIX



4 bytes Control

1 bit in control will be used to decide, if descriptor is with HOST/FPGA

Bit 0:

1 – Host updated required fields and FPGA needs to compress

0 – FPGA finished its job and updated required fields

Descriptor ring

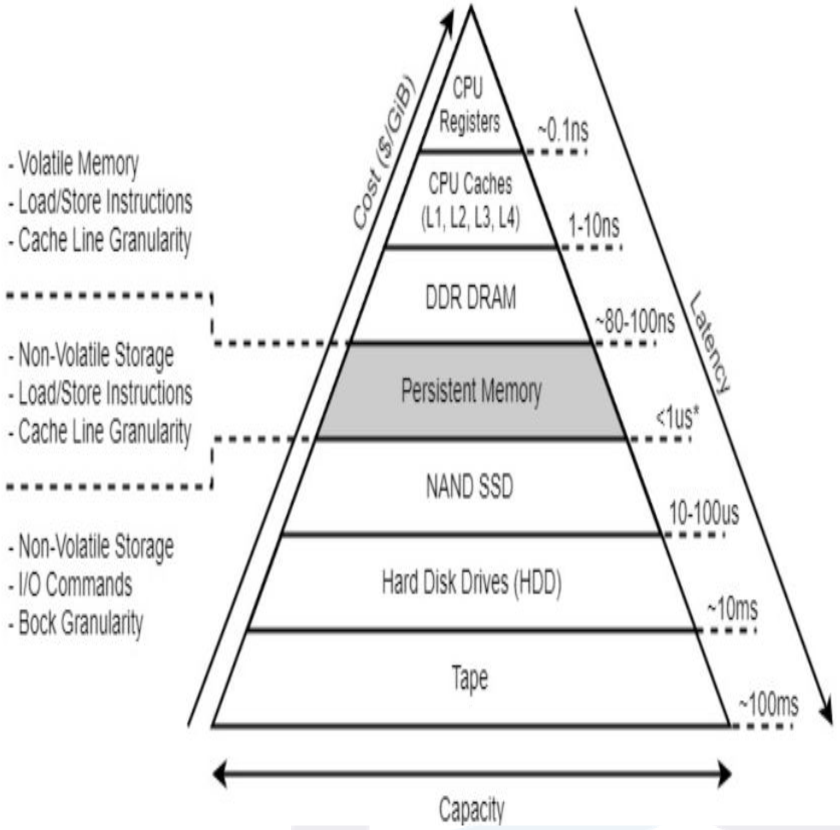
#descriptors = # HW kernels

SRC Address	DST Address	SRC Length	DST Length	Status	Control
SRC Address	DST Address	SRC Length	DST Length	Status	Control
SRC Address	DST Address	SRC Length	DST Length	Status	Control
…					

Increasing CCIX Adaptability with Persistent Memory

Growing Interest on PMEM

- PMEM enable fast, non volatile byte addressable memory
- Huge data can be accessed directly using load/store instructions
- Negligible down time on app restart as entire memory space is readily available
- RDMA can be done directly to/from pmem without interrupting application, avoids doing multiple copies



The figure is taken from https://docs.pmem.io/getting-started-guide/what-is-pmdk

PMEM Usecases

1. **Use Case 1: Treating PMEM as extended memory**
   - Application can treat NVDIMM as normal DIMM on a different NUMA node
2. **Use Case 2: PMEM as standard SSD**
   - Application to file-system interface is maintained. File system storage semantics modified to have memory semantics. DAX driver in the kernel acts as a wrapper between these two. EXT4-DAX and XFS-DAX are popular examples
3. **Use Case 3: PMEM aware application:**
   - Application needs to be rewritten with load /store operations done directly on PMEM using PMDK
   - Lot of popular databases like Redis Aerospike and MongoDB already have source code with PMDK API support
   - PMDK is vendor neutral and open source
4. **Use Case 4: Fusion of PMEM aware application (FUSE) and filesystem:**
   - User level file system with FUSE where application continues to using existing file operations on this minimal file system.
   - There is zero-copy operation support for moving data from file-system data and user space data

Flash Memory Summit

CCIX Based PMEM Solutions

- Expansion memory in CCIX can be utilized as PMEM with load store and cache coherency
- The FPGA can bring in additional benefits for storage acceleration like Compression/Encryption on expansion memory
- Near memory compute will benefit HPC applications
- Extend the custom distributed filesystem for zero copy network transmission and storage
- FPGA pooling might be done for adding additional storage and compute. Entire data is visible to Host CPU

Questions?