Gaining NVMe-oF™ System Performance Through Smart CPU Core Affinity

Sponsored by NVM Express organization, the owner of NVMe® specifications

Speaker

Adi Folkman
Product Manager

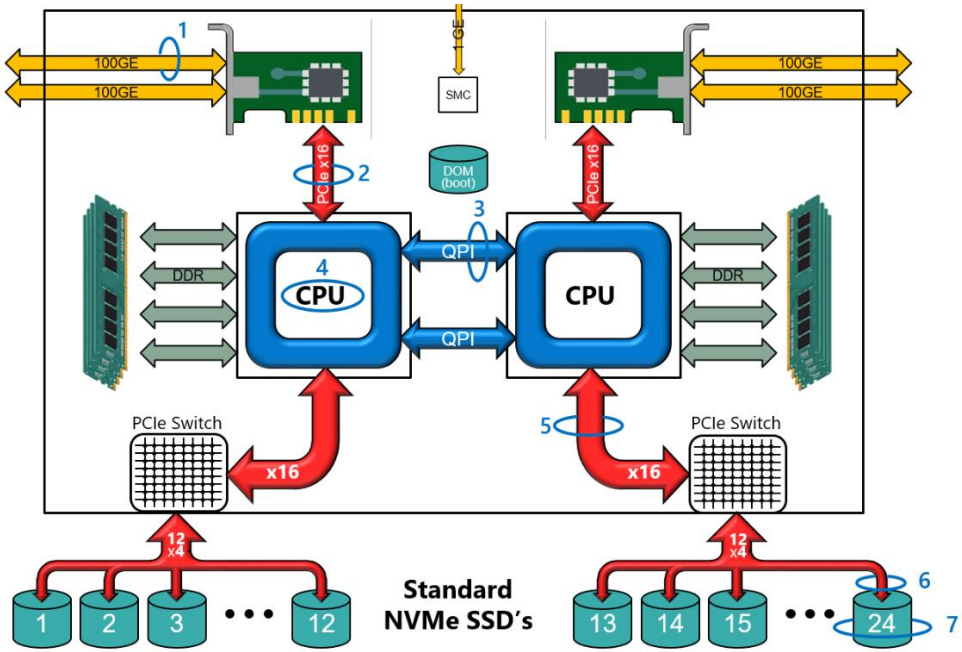
KIOXIA

Agenda

- Problem statement
- What is CPU core affinity?
- Solution proposal and best practices
- Performance enhancements – profiles and results

Problem Statement

- NUMA architecture
- Bottlenecks:
  - Bus (QPI)
  - Context switching



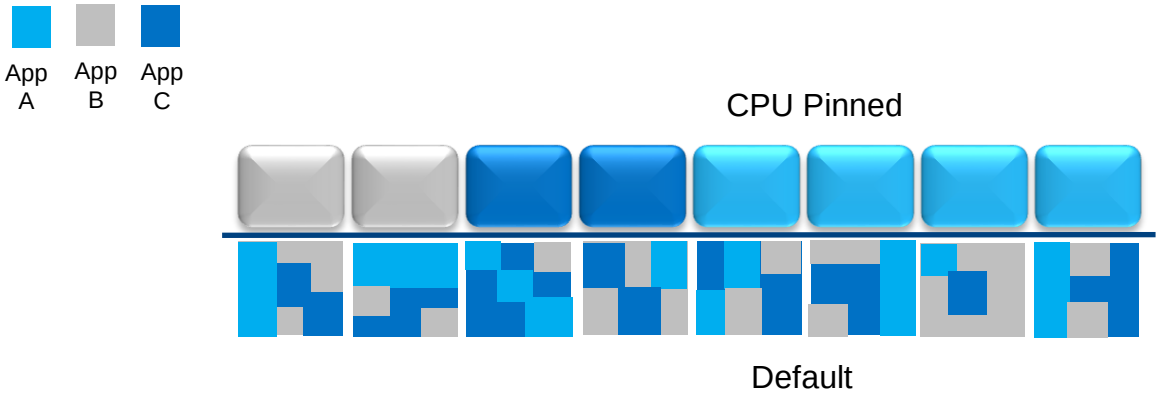
- These cause under utilization of system resources, mainly the SSDs

  → You are not fully utilizing all the computation power/resources available in your system

What is CPU Core Affinity?

- Binding a thread to a specific CPU core or a set of cores.

- It is a controllable, per-process property, describing which processors the process may execute upon.



Key Benefits:
Working in an NVMe-oF™ Environment

- ✓ Improve performance, as it is using dedicated CPU(s)
- ✓ Maximize CPU cache efficiency
- ✓ Improve memory access speed
- ✓ Separation from noisy neighbors

But:

- ❖ Platform dependent, not trivial to configure properly
- ❖ Over-working busy CPUs, might affect both application and system performance

An Industry Perspective

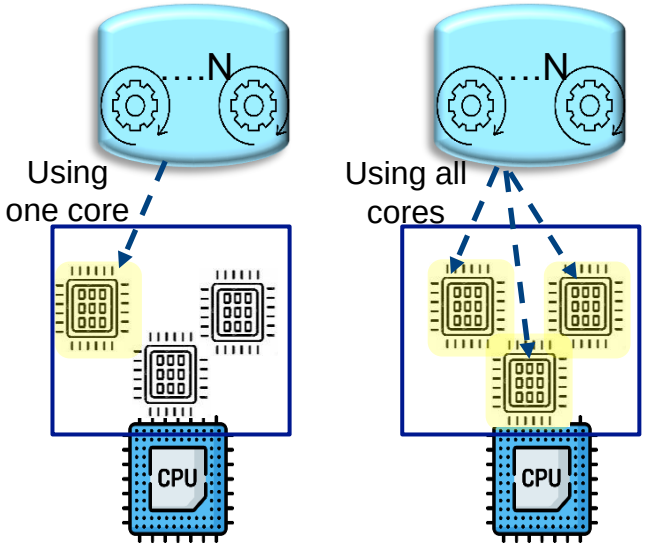
- There is much evidence that affinization has a significant impact on processing efficiency. Different types of affinization characteristics perform differently
- One of the challenges for the industry is to locate the different technologies, architectures, and system topologies. Difficulties may occur due to processor architecture changes

  For example: newer processors have frequency scaling technologies that allow scaling up of specific cores on-demand. The development of this feature in CPU hardware means that the CPU affinity setting, may make a dramatic difference in performance

- Future work will probably include setting processor affinity in systems with many cores, some of which are optimized for different processes

Solution Proposal and Best Practices

Motivation: optimize CPU utilization to gain performance, on different platforms



✓ Higher CPU utilization, in multi core systems

- In storage servers, this enables hardware resource reservations to guarantee quality of service, and avoid storage device under utilization

✓ Leveraging Non-Uniform Memory Access (NUMA)

- Cluster microprocessors in a way memory may be accessed locally
- Allow application threads to be executed on the CPU core which is closest to its memory ban

Solution Proposal and Best Practices

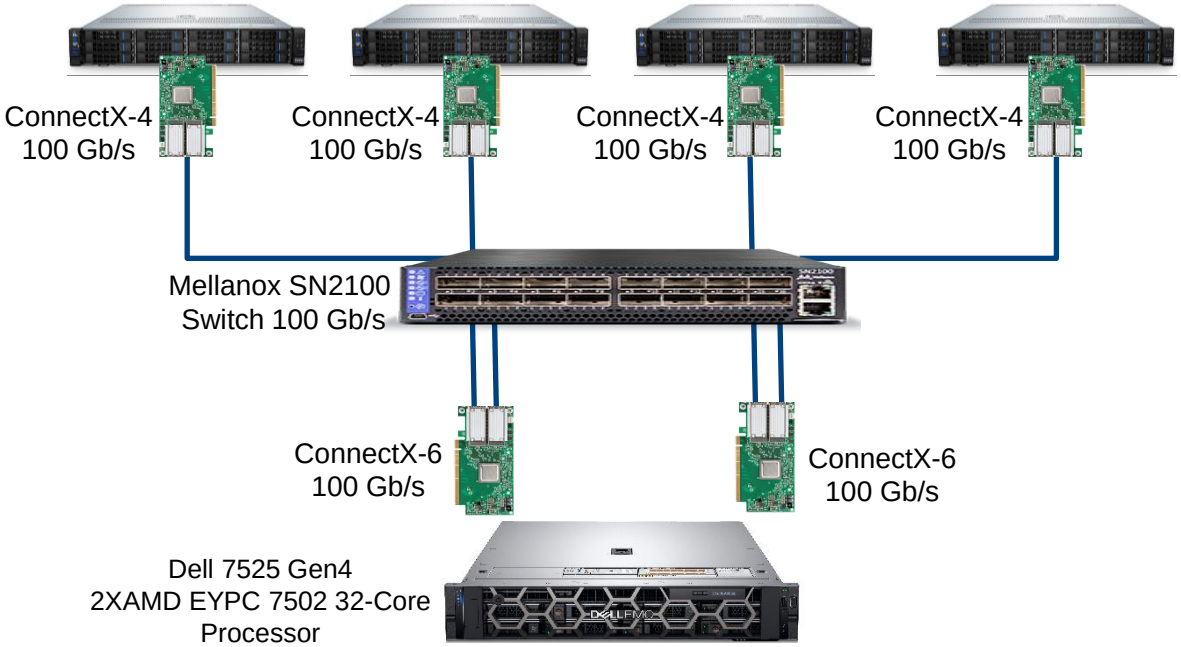
- ✓ Define and design according to the desired total # of cores reserved for management:
  - The reserved cores need to be evenly distributed among all existing NUMAs

    OR
  - Located on the NUMAs that are not attached to any interface for data transfer

- ✓ Use optimization methods, like a cost model, which will output an optimized set of parameters according to the set up.

Performance Enhancements – An Example

- Dell 7525: 2x AMD EPYC™ 7502 32-Core Processor PCIe® 4.0 Architecture
- PCIe 4.0 servers have 16 NUMAs, 4 cores per NUMA.



Performance Enhancements – An Example

Dell 7525: 2x AMD EPYC™ 7502 32-Core Processor PCIe® 4.0 Technology

Questions?

Architected for Performance