POSTECH
포항공과대학교

DCS: A Fast and Scalable Device-Centric Server Architecture

Jaehyung Ahn, Dongup Kwon, Youngsok Kim,
Mohammadamin Ajdari, Jaewon Lee, and Jangwoo Kim

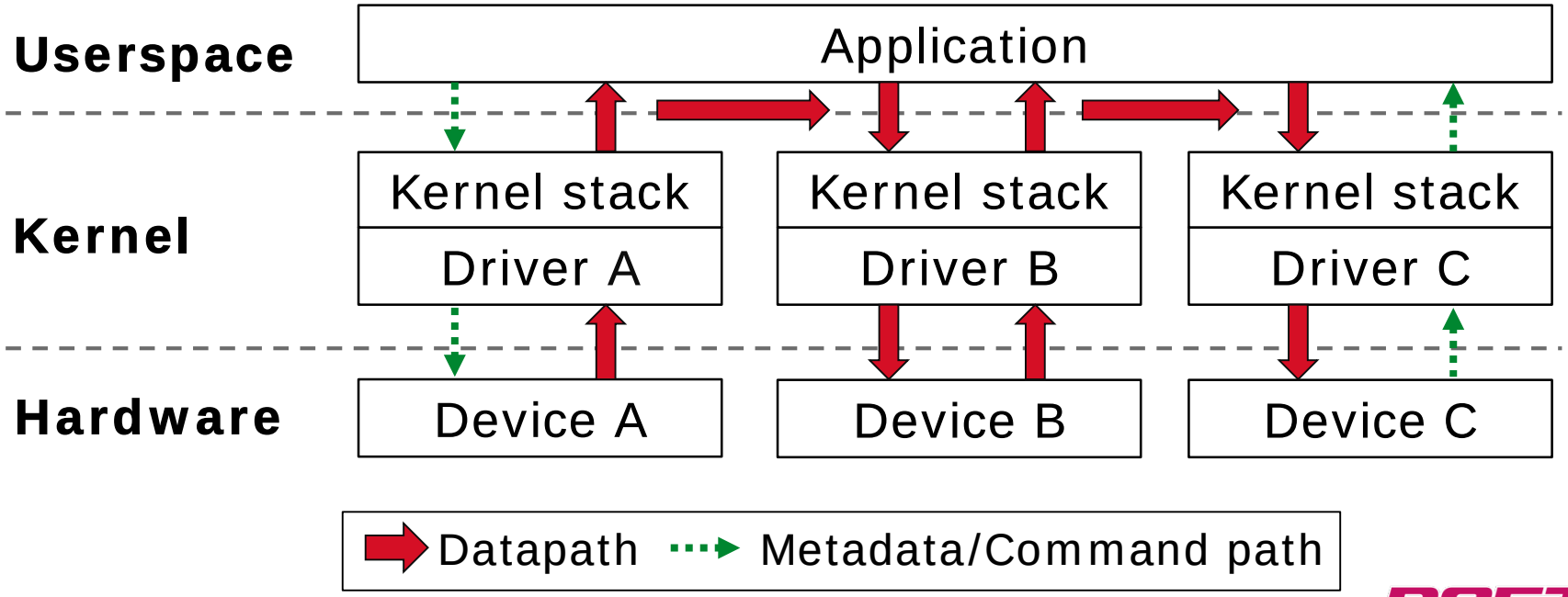
{jh2ekd, nankdu7, elixir, majdari, spiegel0, jangwoo}@postech.ac.kr

High Performance Computing Lab
Pohang University of Science and Technology (POSTECH)

POSTECH

Inefficient device utilization

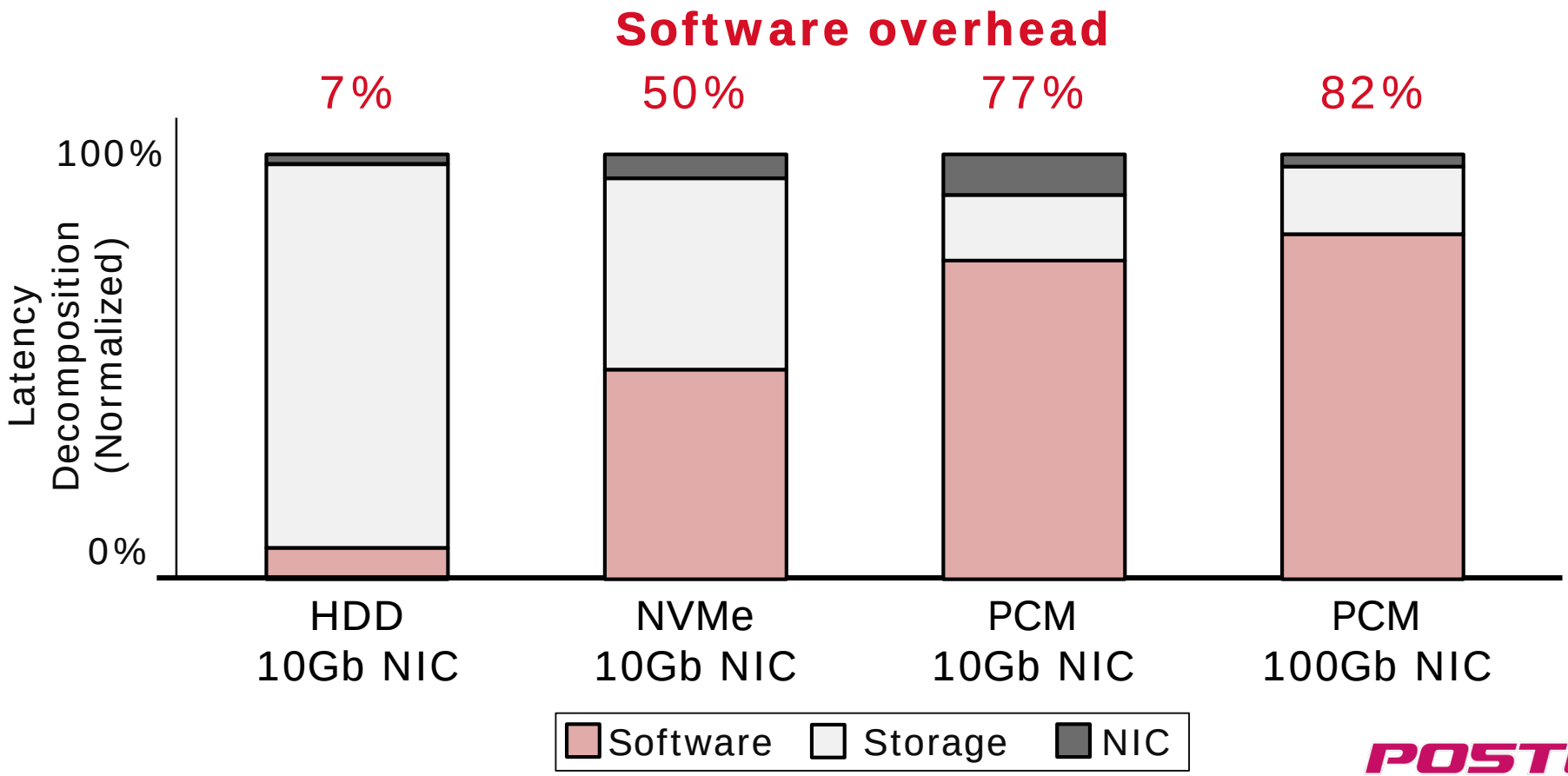
- **Host-centric device management**
  - Host manages every device invocation
  - Frequent host-involved layer crossings
    - Increases latency and management cost



POSTECH

Latency: High software overhead

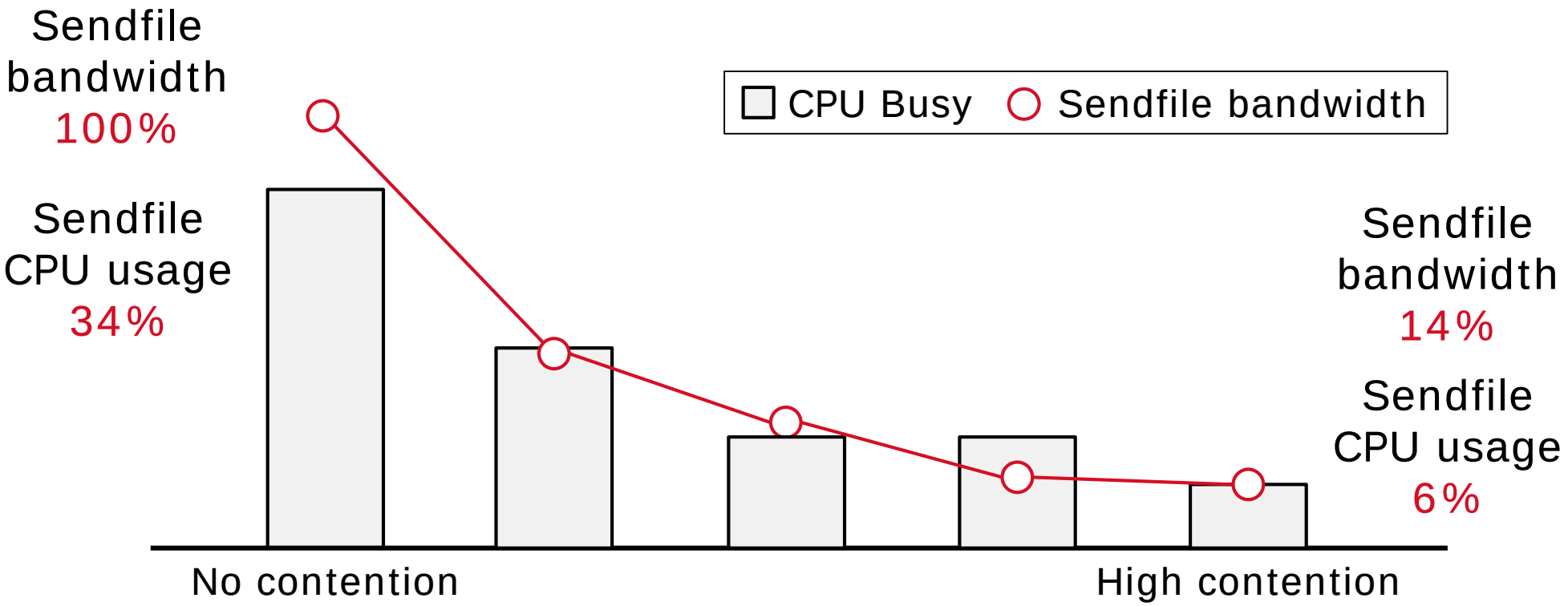
- **Single sendfile: Storage read & NIC send**
  - Faster devices, more software overhead



POSTECH

Cost: High host resource demand

- **Sendfile under host resource (CPU) contention**
  - Faster devices, more host resource consumption



*Measured from NVMe SSD/10Gb NIC

Index

- Inefficient device utilization
- **Limitations of existing solutions**
- DCS: Device-Centric Server architecture
- Experimental results
- Conclusion

Limitations of existing work

- **Single-device optimization**
  - Do not address inter-device communication

    e.g., Moneta (SSD), DCA (NIC), mTCP (NIC), Arrakis (Generic)
- **Inter-device communication**
  - Not applicable for unsupported devices

    e.g., GPUNet (GPU-NIC), GPUDirect RDMA (GPU-Infiniband)
- **Integrating devices**
  - Custom devices and protocols, limited applicability

    e.g., QuickSAN (SSD+NIC), BlueDBM (Accelerator – SSD+NIC)

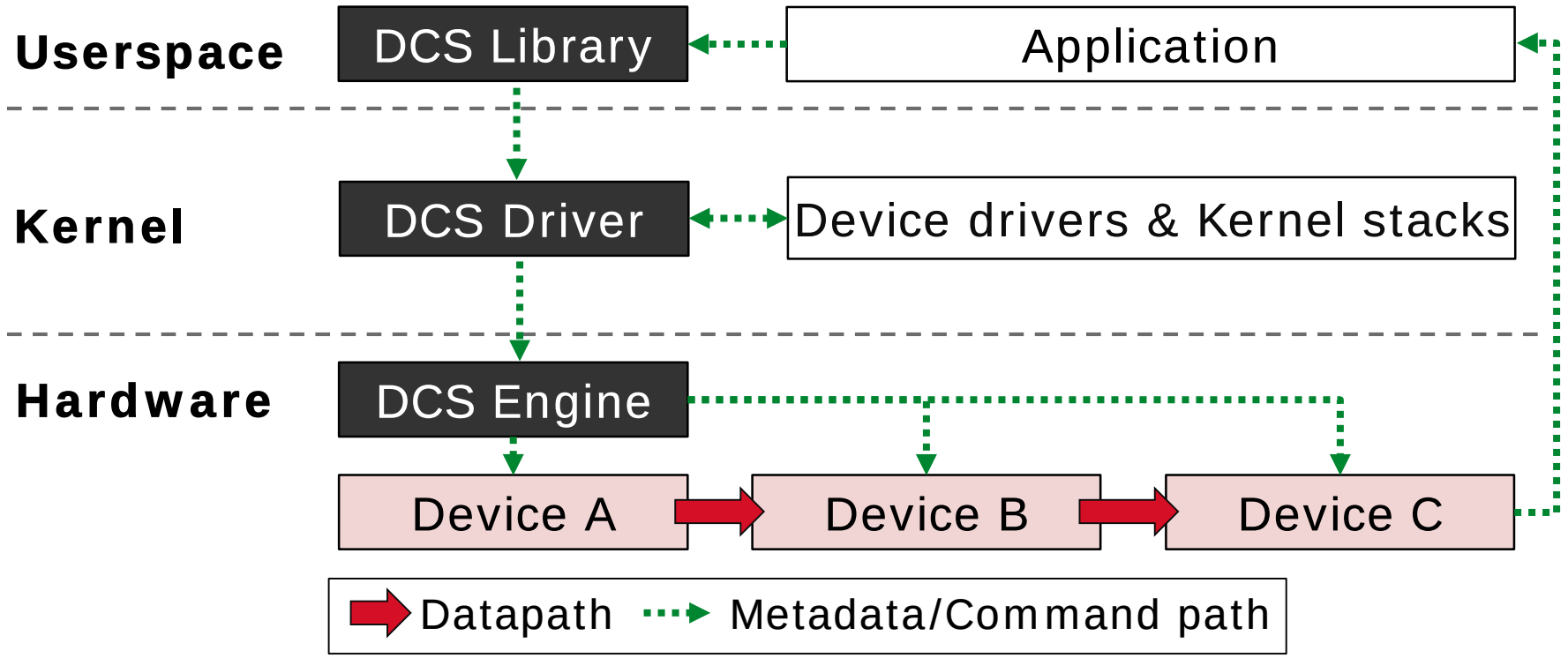
Need for fast, scalable, and generic inter-device communication

Index

- Inefficient device utilization
- Limitations of existing solutions
- **DCS: Device-Centric Server architecture**
  - **Key idea and benefits**
  - Design considerations
- Experimental results
- Conclusion

DCS: Key idea

- **Minimize host involvement & data movement**



Single command → Optimized multi-device invocation

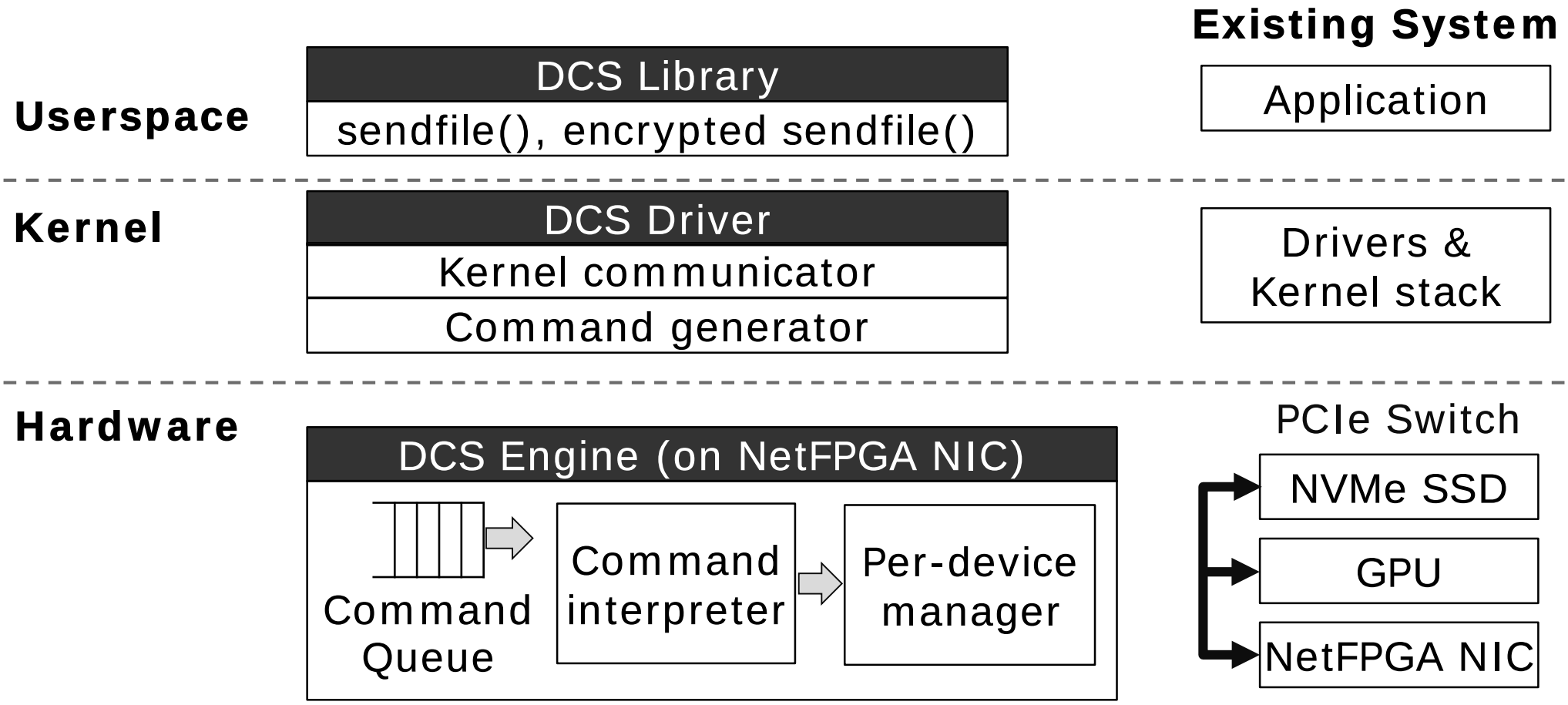
DCS: Benefits

- **Better device performance**
  - Faster data delivery, lower total operation latency
- **Better host performance / efficiency**
  - Resource/time spent for device management now available for other applications
- **High applicability**
  - Relies on existing drivers / kernel supports / interfaces
  - Easy to extend and cover more devices

Index

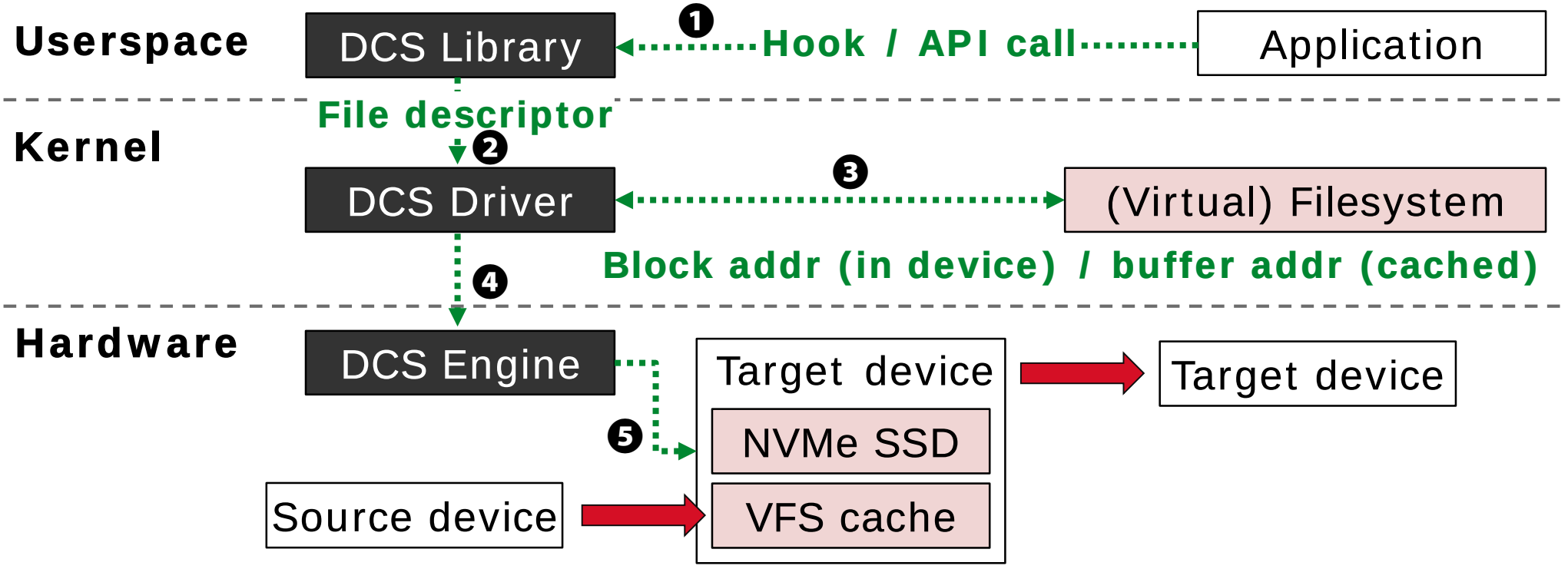
- Inefficient device utilization
- Limitations of existing solutions
- **DCS: Device-Centric Server architecture**
  - Key idea and benefits
  - **Design considerations**
    - By discussing implementation details
- Experimental results
- Conclusion

DCS: Architecture overview



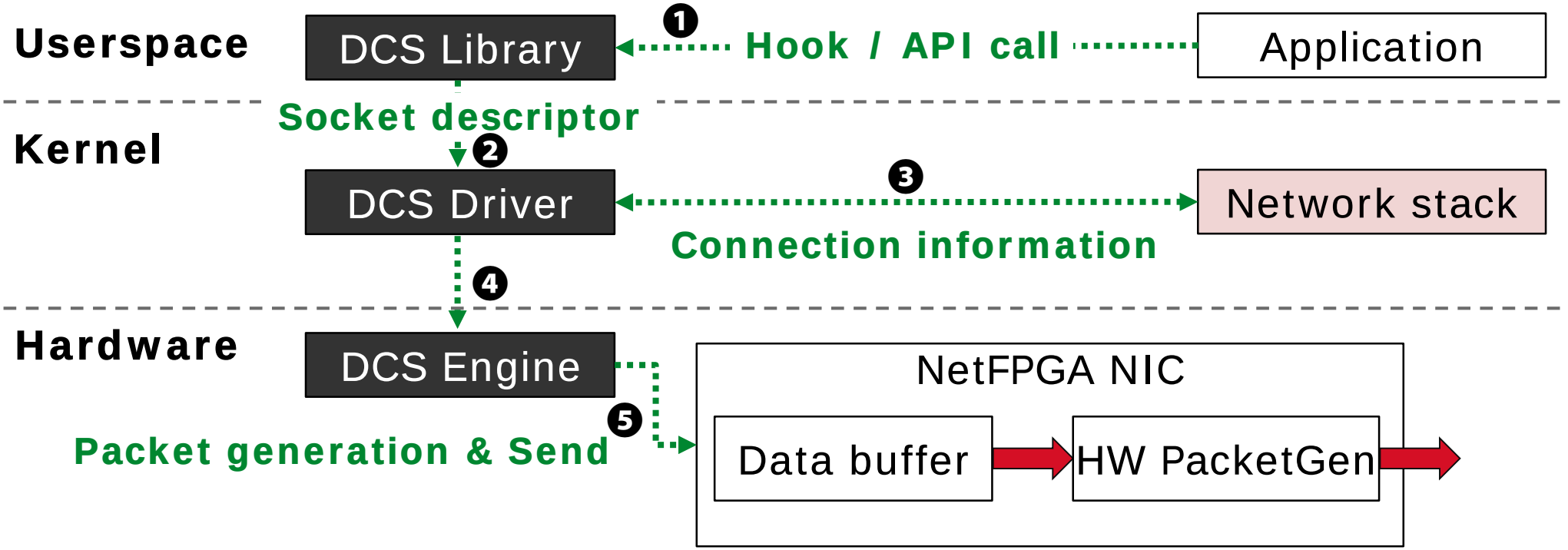
Fully compatible with existing system

Communicating with storage



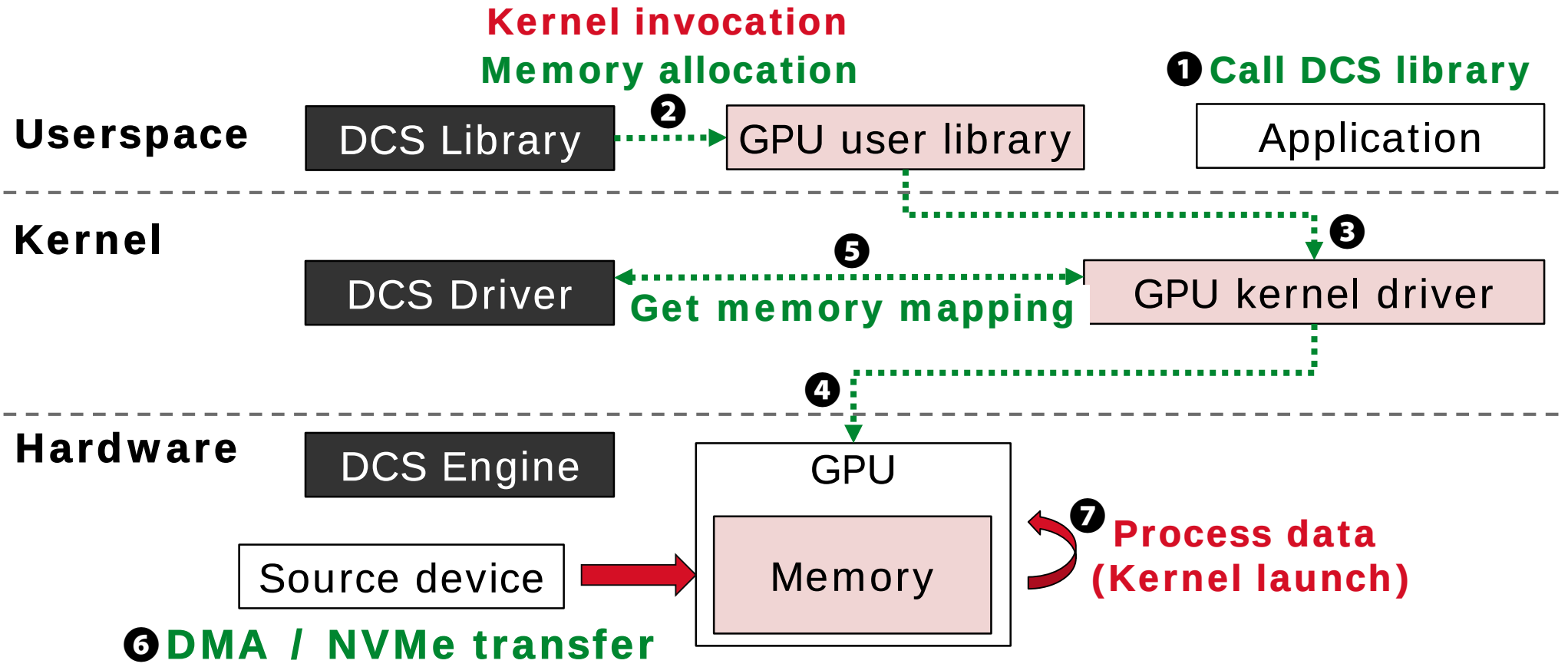
Data consistency guaranteed

Communicating with network interface



HW-assisted packet generation

Communicating with accelerator



Direct data loading without memcpy

Index

- Inefficient device utilization
- Limitations of existing solutions
- DCS: Device-Centric Server architecture
- **Experimental results**
- Conclusion

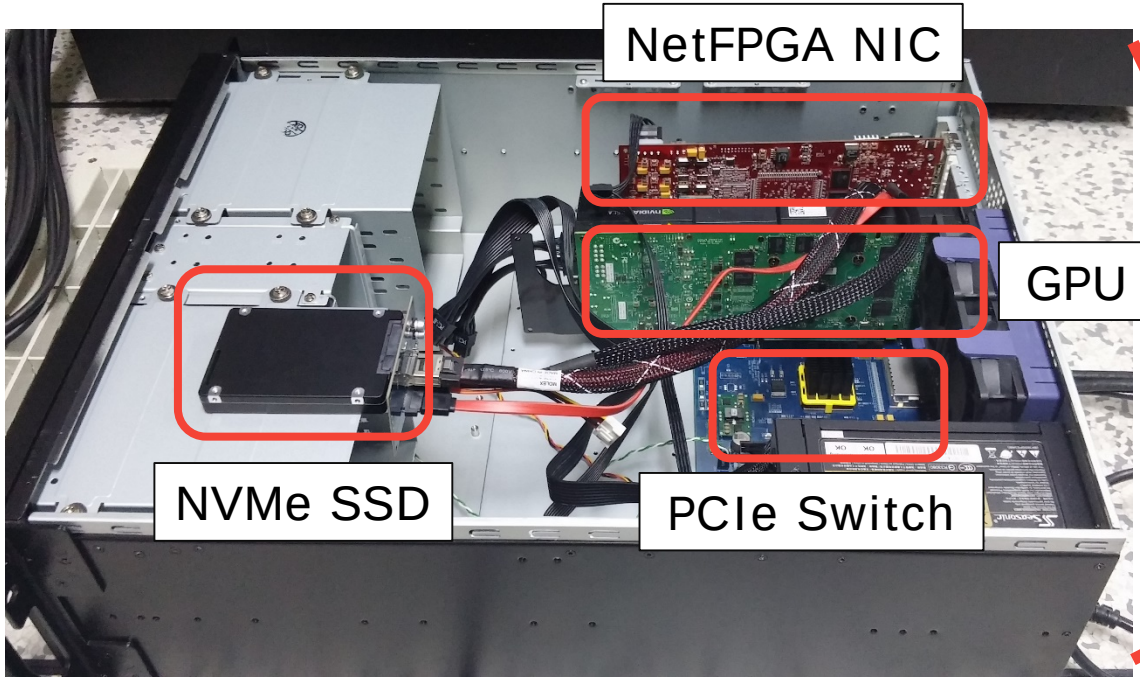
Experimental setup

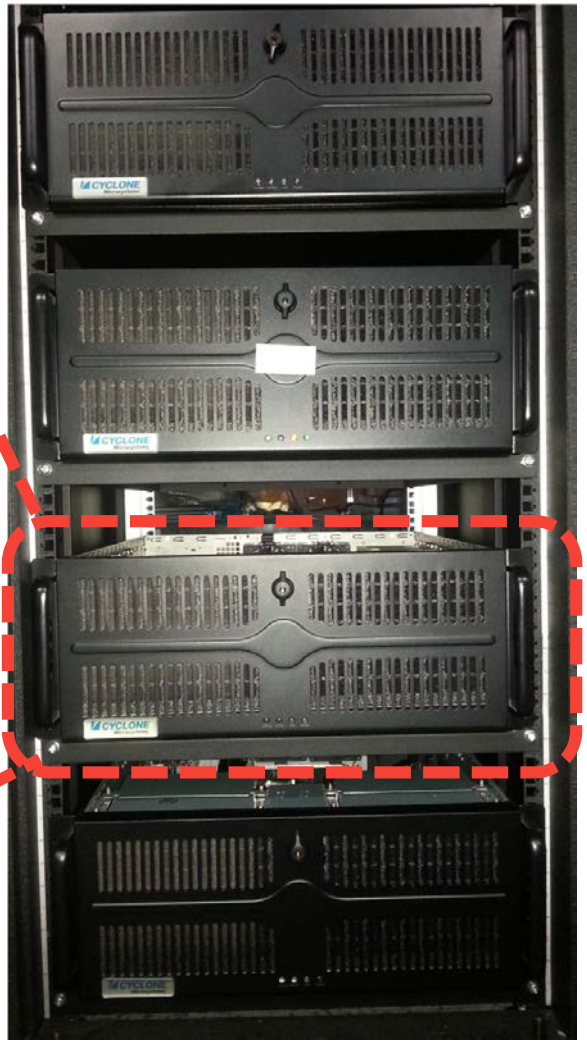
- **Host: Power-efficient system**
  - Core 2 Duo @ 2.00GHz, 2MB LLC
  - 2GB DDR2 DRAM

- **Device: Off-the-shelf emerging devices**
  - Storage: Samsung XS1715 NVMe SSD
  - NIC: NetFPGA with Xilinx Virtex 5 (up to 1Gb bandwidth)
  - Accelerator: NVIDIA Tesla K20m
  - Device interconnect: Cyclone Microsystems PCIe2-2707 (Gen 2 switch, 5 slots, up to 80Gbps)

DCS prototype implementation

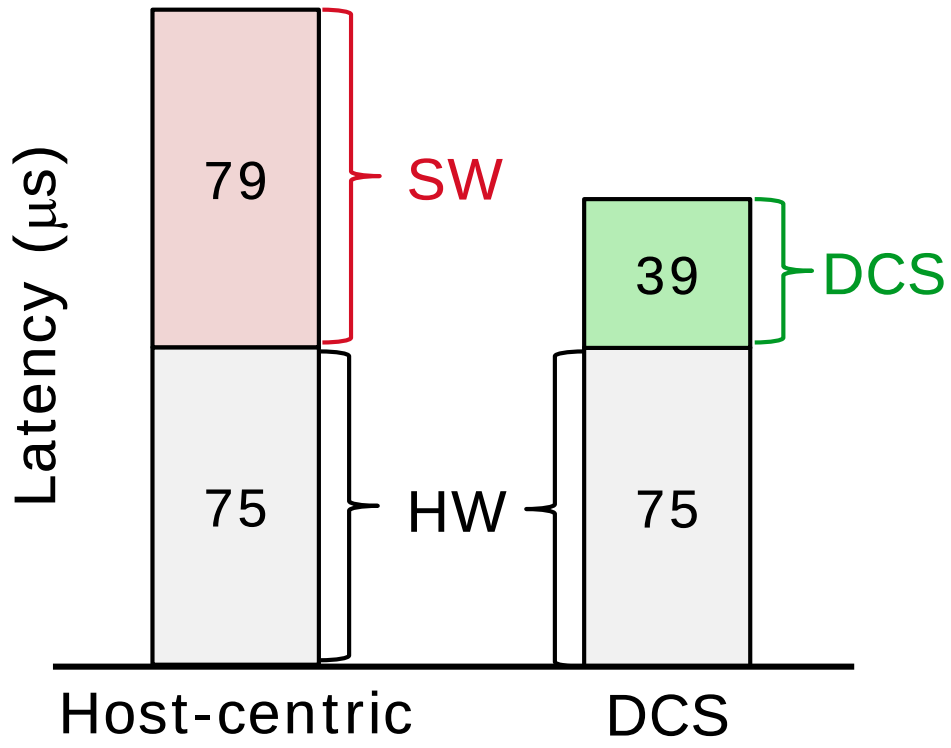
- **Our 4-node DCS prototype**
  - Can support many devices per host





Reducing device utilization latency

- **Single sendfile: Storage read & NIC send**
  - Host-centric: Per-device layer crossings
  - DCS: Batch management in HW layer

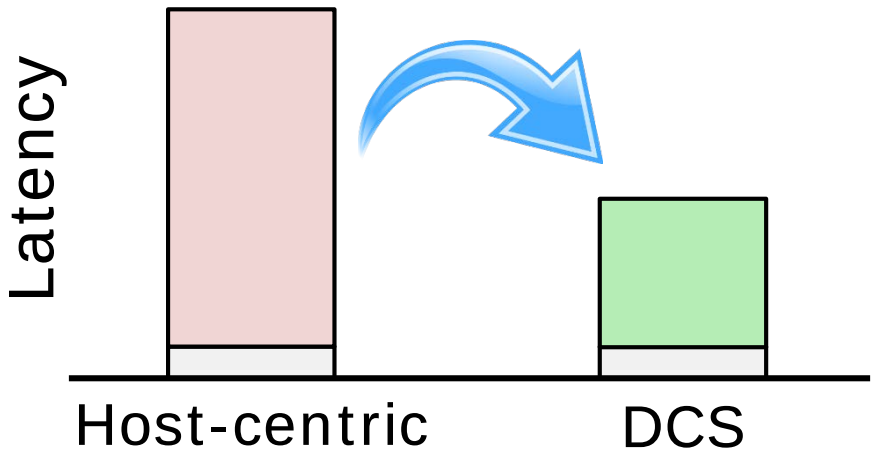


POSTECH

Reducing device utilization latency

- **Single sendfile: Storage read & NIC send**
  - Host-centric: Per-device layer crossings
  - DCS: Batch management in HW layer

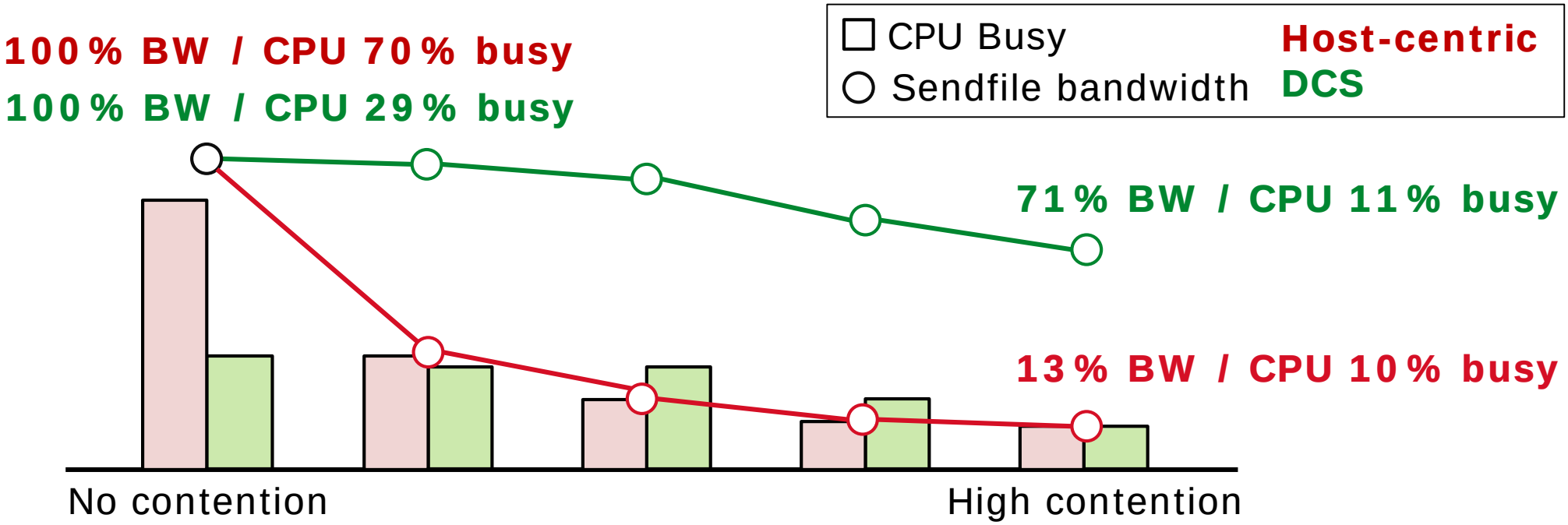
2x latency improvement
(with low-latency devices)



POSTECH

Host-independent performance

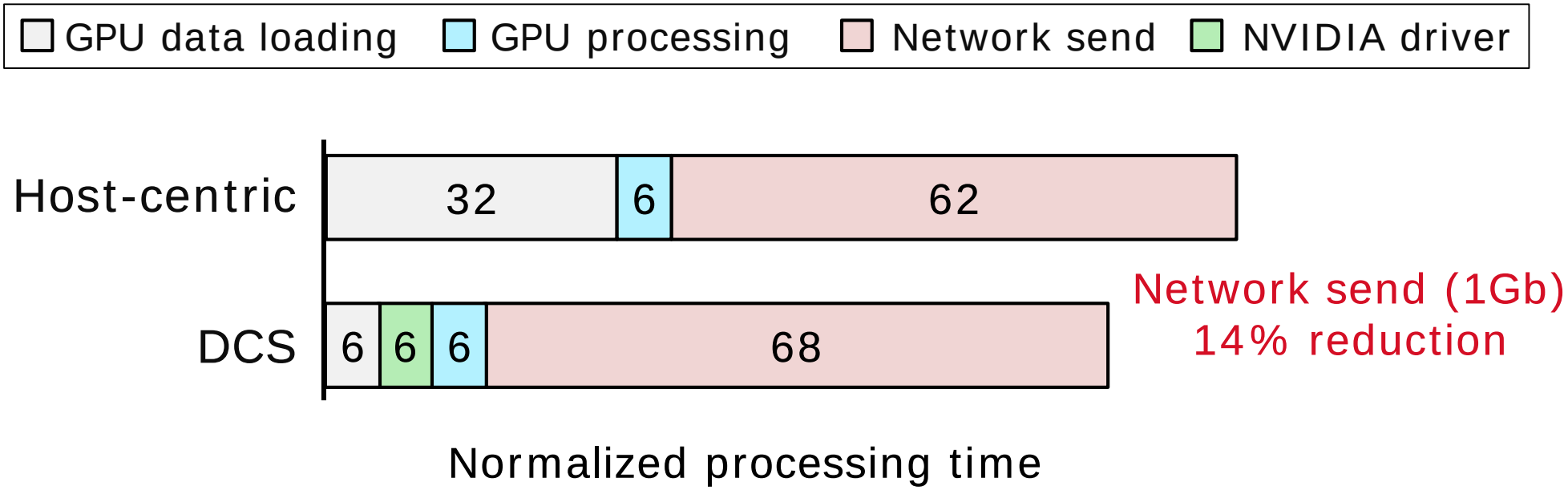
- **Sendfile under host resource (CPU) contention**
  - Host-centric: host-dependent, high management cost
  - DCS: host-independent, low management cost



High performance even on weak hosts

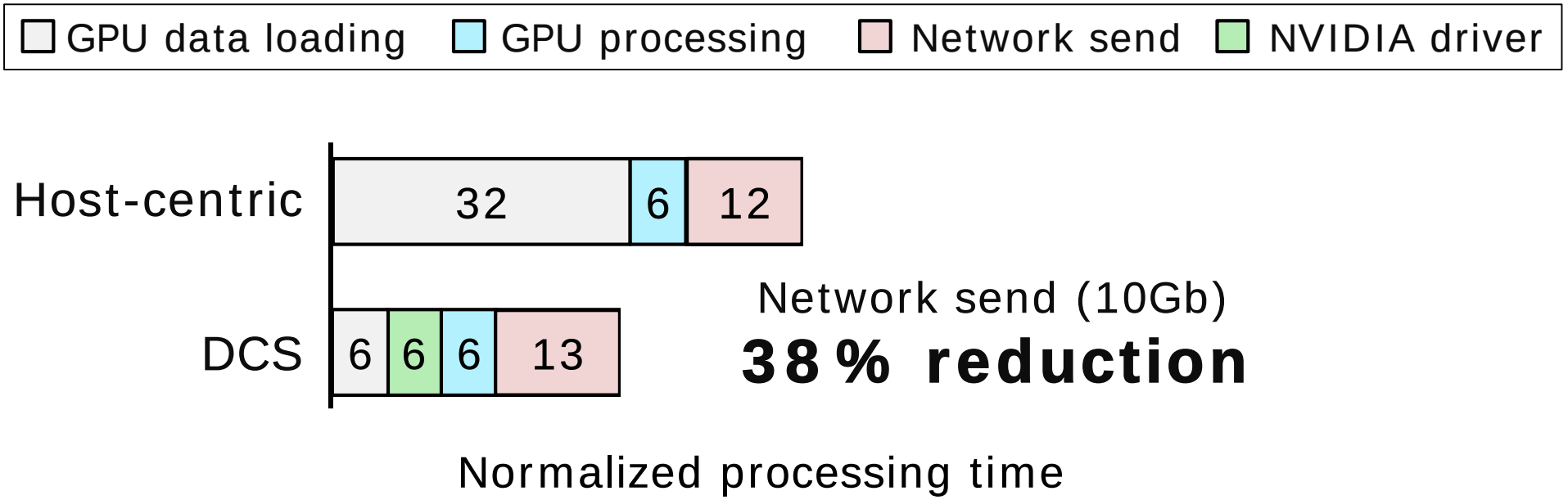
Multi-device invocation

- **Encrypted sendfile (SSD → GPU → NIC, 512MB)**
  - DCS provides much efficient data movement to GPU
  - Current bottleneck is NIC (1Gbps)



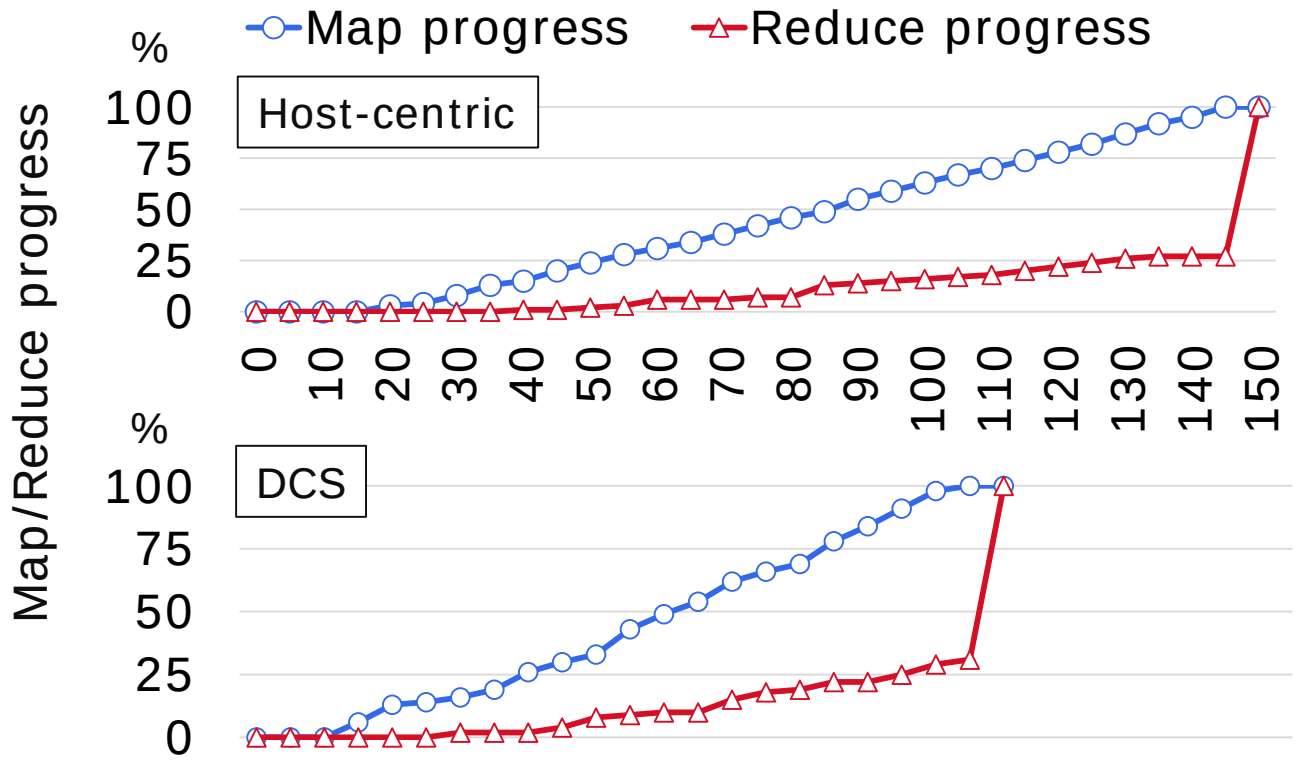
Multi-device invocation

- **Encrypted sendfile (SSD → GPU → NIC, 512MB)**
  - DCS provides much efficient data movement to GPU
  - Current bottleneck is NIC (1Gbps)



Real-world workload: Hadoop-grep

- **Hadoop-grep (10GB)**
  - Faster input delivery & smaller host resource consumption

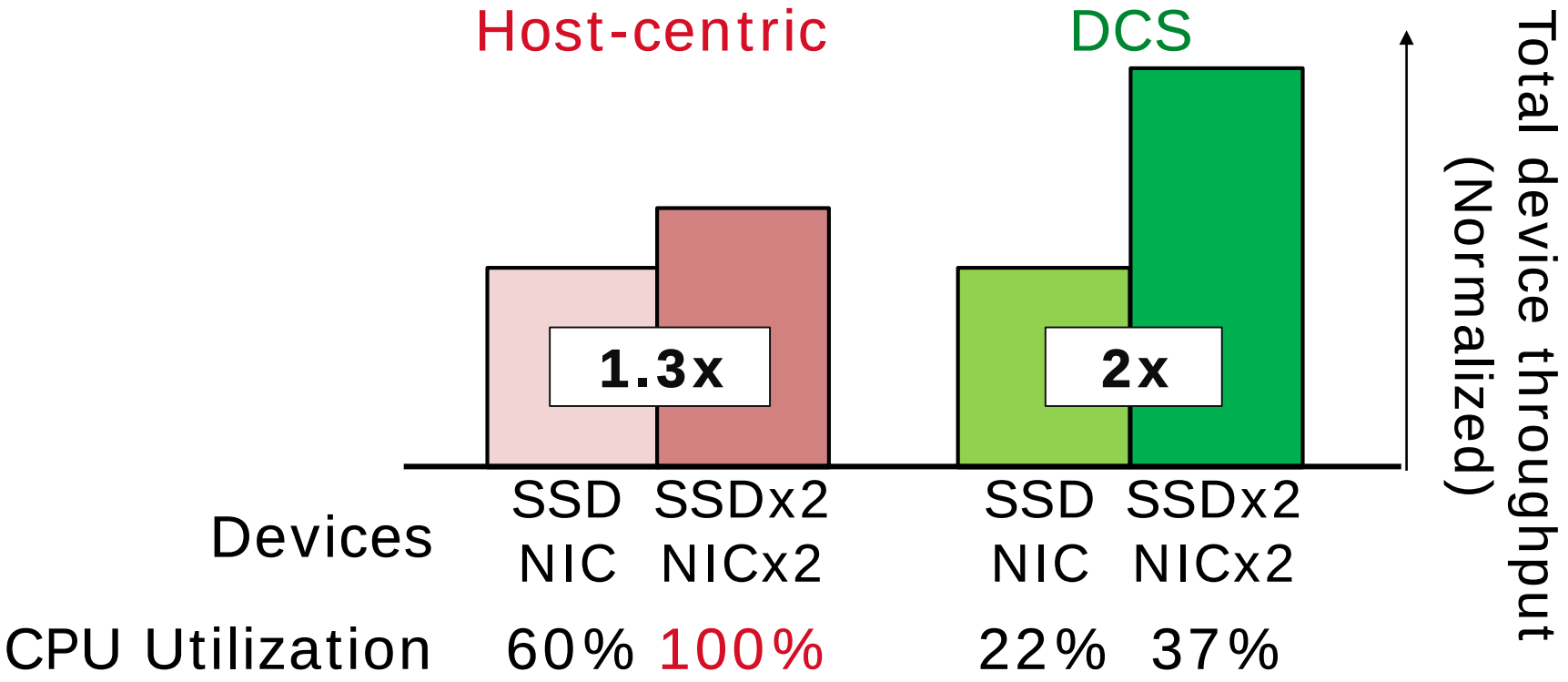


38% faster processing

POSTECH

Scalability: More devices per host

- **Doubling # of devices in a single host**



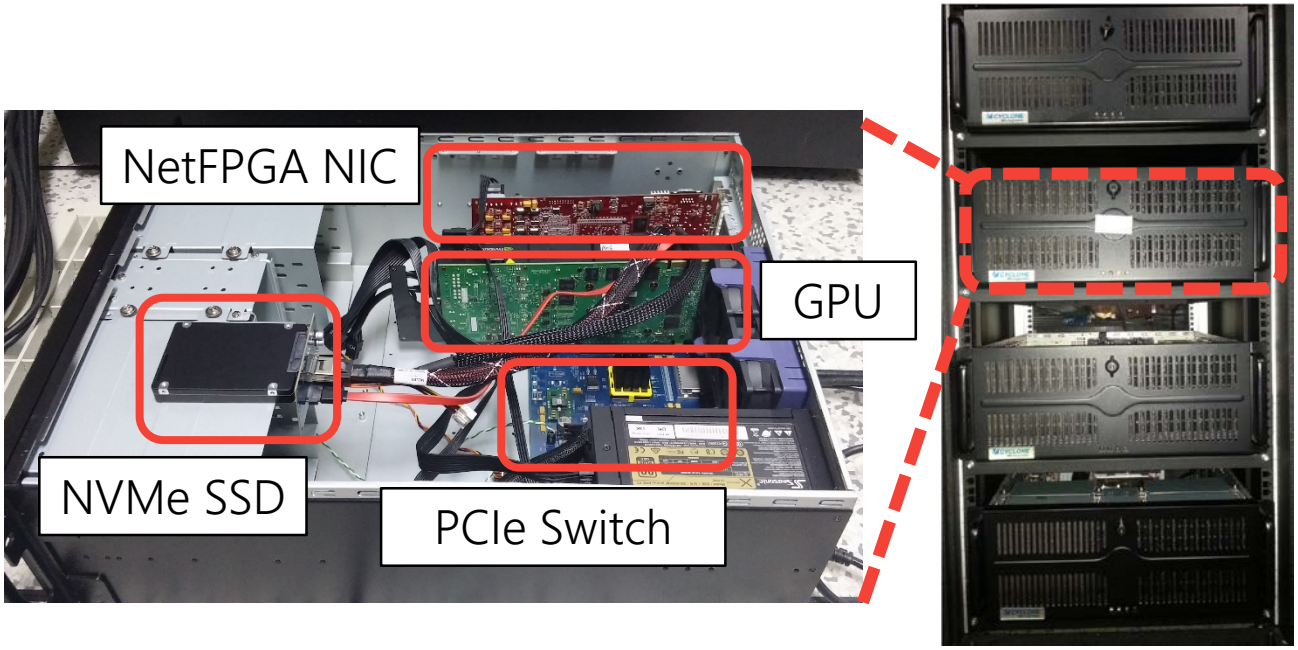
Scalable many-device support

Conclusion

- **Device-Centric Server architecture**
  - Manages emerging devices on behalf of host
  - Optimized data transfer and device control
  - Easily extensible modularized design

- **Real hardware prototype evaluation**
  - Device latency reduction: ~25%
  - Host resource savings: ~61%
  - Hadoop-grep speed improvement: ~38%

Thank you!



Device latency reduction ~25%

Host resource savings ~61%

Hadoop-grep speed improvement ~38%

High Performance Computing Lab

Pohang University of Science and Technology (POSTECH)