Los Alamos
NATIONAL LABORATORY

EST. 1943

NNSA
National Nuclear Security Administration
Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

OpenSNAPI: Toward a Unified API for SmartNICs

Brody Williams[*^],
Wendy Poole[‡], Steve Poole[‡]

August 13, 2020

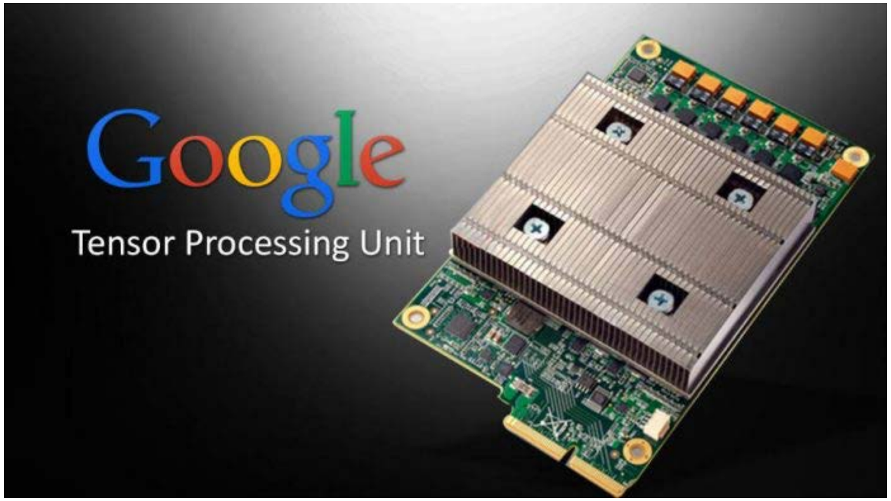
* Presenter, ^ Texas Tech University, ‡ Los Alamos National Laboratory

Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

LA-UR-20-25826

Motivation

- Performance improvements derived from silicon level CPU enhancements have stalled
  - End of Moore's Law & Dennard Scaling
- Paradigm shift in computer architecture
  - Distinct devices optimized for execution of specific workloads
  - Different uarchs and ISAs
  - GPUs, TPUs, etc.
- Hardware/Software Codesign
  - Purpose-built systems
- The Future is Heterogenous



SmartNICs

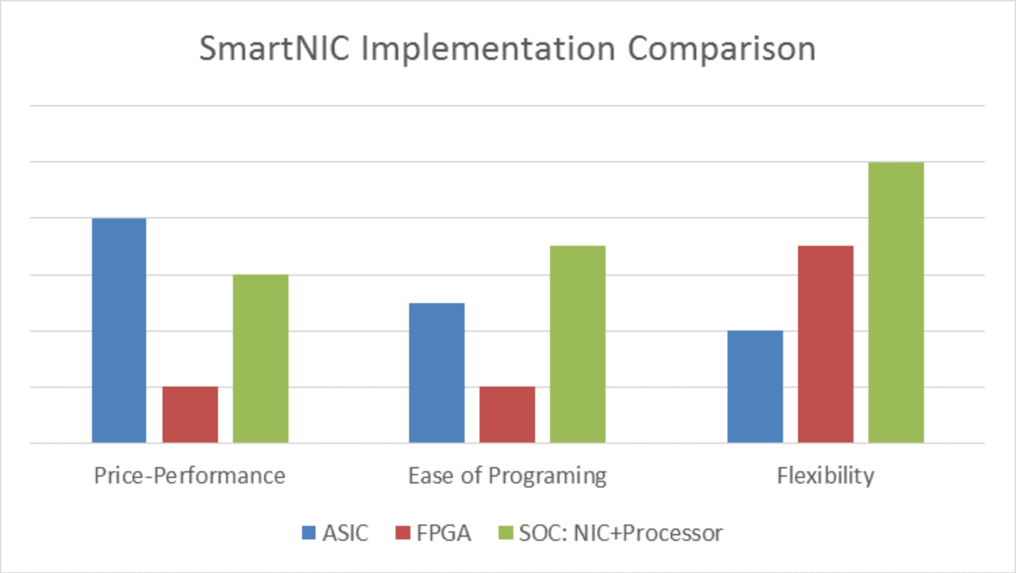
Different varieties of SmartNICs

- ASIC Based
  - Excellent price-performance
  - Vendor development cost high
  - Programmable and extensible
    - Easy to program but flexibility is limited to pre-defined capabilities

- FPGA Based
  - Good performance but expensive
  - Very difficult to program
  - Workload specific optimization

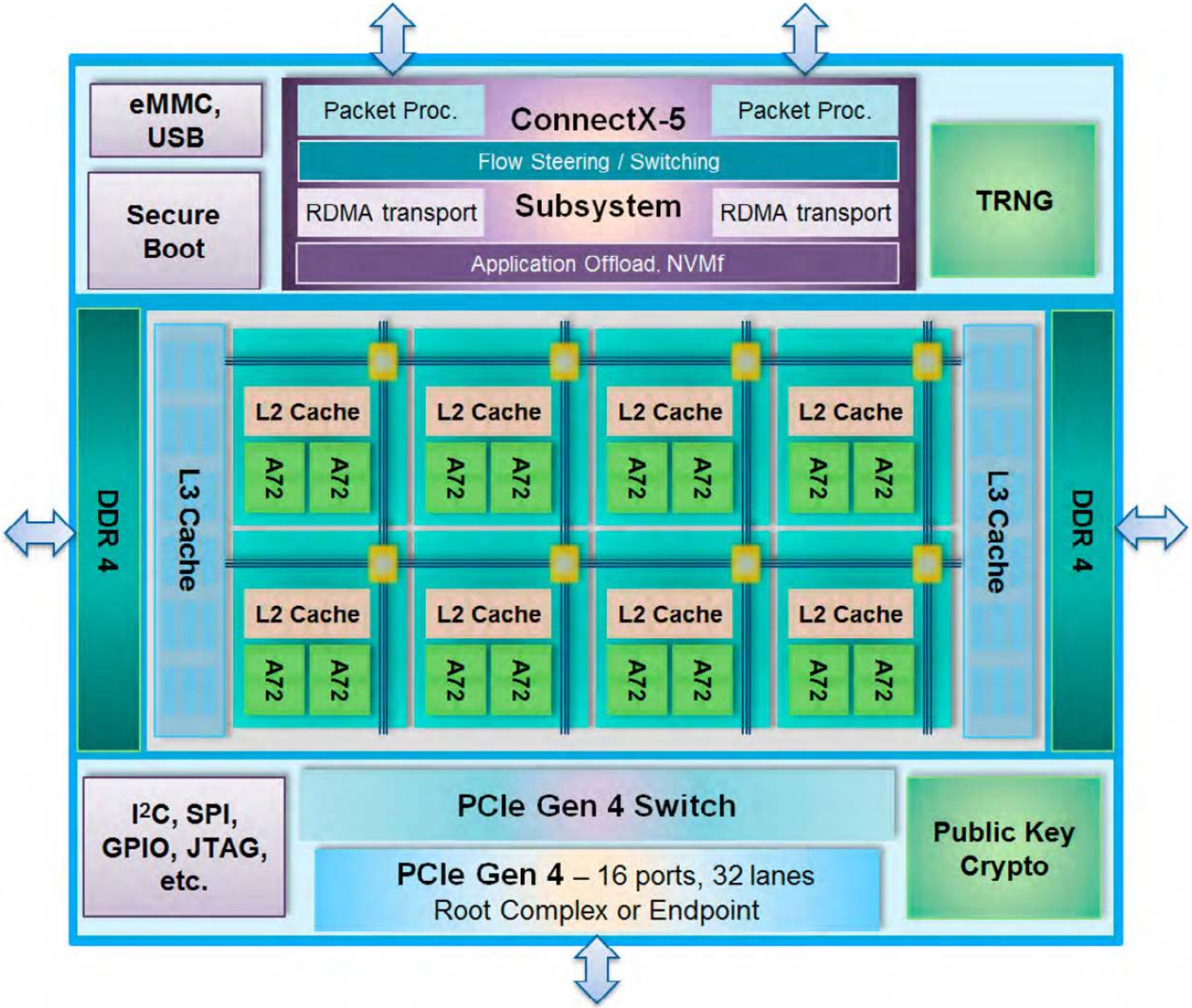
- SOC Based

  System on Chip - NIC + CPU
  - Good price-performance
  - C Programmable Processors
  - Highest Flexibility
  - Easiest programmability



Source: https://blog.mellanox.com/2018/08/defining-smartnic/

Mellanox BlueField SoC Architecture



SmartNICs Revisited

- Traditional Use Case
  - Offload low-level networking functionality from the CPU
  - ASIC/FPGA SmartNICs well-suited to this role
    - Set it and forget it

- SoC-based SmartNICs
  - Powerful and Flexible
    - ARM cores
    - Onboard memory
  - Easy to program
    - High level languages
      - E.G - C, C++
  - Can we use these additional resources to accelerate performance in heterogeneous architectures in a manner similar to other accelerators?

SmartNICs Revisited

- Benefits of SmartNIC Accelerators
  - Operate on data in place at network edge
    - Minimize data movement
  - Take advantage of data “in-flight”
  - Collective operations
  - Atomic memory operations (AMOs)
  - Energy efficiency
    - Edge/In-network computing is efficient
    - Arm cores are typically more energy-efficient than x86_64 analogs
    - Reduces operational costs

- Goal: Acceleration at the application level
  - Computational kernel & communication offloading
  - Real-world scientific workloads
    - Computation and communication, dependencies, irregular memory access patterns
    - Not just about LINPACK scores

OpenSNAPI

- OpenSNAPI is a project of the UCF Consortium

- Straight from the source:
  - *“OpenSNAPI is a collaboration between industry, laboratories and academia with the goal to create a standard application programming interface (API) for accessing the compute engines on the network, and specifically on the smart network adapter. OpenSNAPI allows application developers to leverage the network compute cores in parallel to the host compute cores for accelerating application runtime, and to perform operations and processing closer to the data.”*

Source: https://www.ucfconsortium.org/projects/opensnapi/

OpenSNAPI – Possible Realizations

- OpenSNAPI could be modeled after several existing paradigms
  - Compile and run independent pieces of code on the SmartNICs
    - Easily doable with SHMEM, MPI, etc.
    - May increase code complexity
    - Likely to incur performance overheads
  - Function calls into SmartNIC-resident library
    - Similar to CUDA
  - Computation offloading and data movement via #pragma directives
    - OpenMP, OpenACC, …
    - Easier to read and incorporate into existing application
      - Conditional compilation
- Alternatively, a completely new language could be designed from the ground up
  - Large development overhead

Work Thus Far

- Focus on exploring what is possible with SmartNIC-based acceleration
  - Examine feasibility of both computation and communication offloading with scientific applications

- Utilize SmartNICs within SHMEM/MPI as if completely distinct nodes
  - Minimize experimentation overhead
  - Distinct code segments for host and sNIC PEs
    - MPMD model

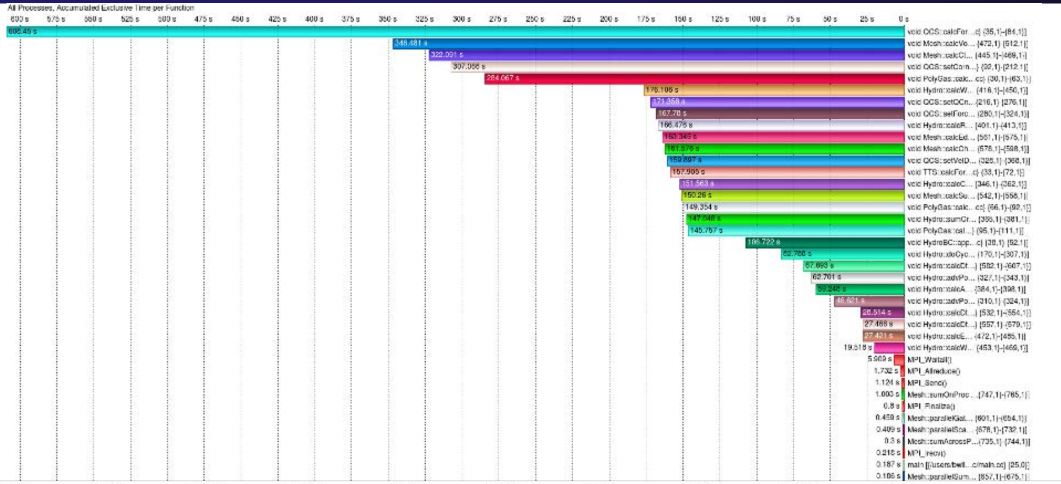
- Benchmarks
  - Two Department of Defense proxy applications
  - PENNANT
  - HPC Challenge RandomAccess, or GUPs, benchmark

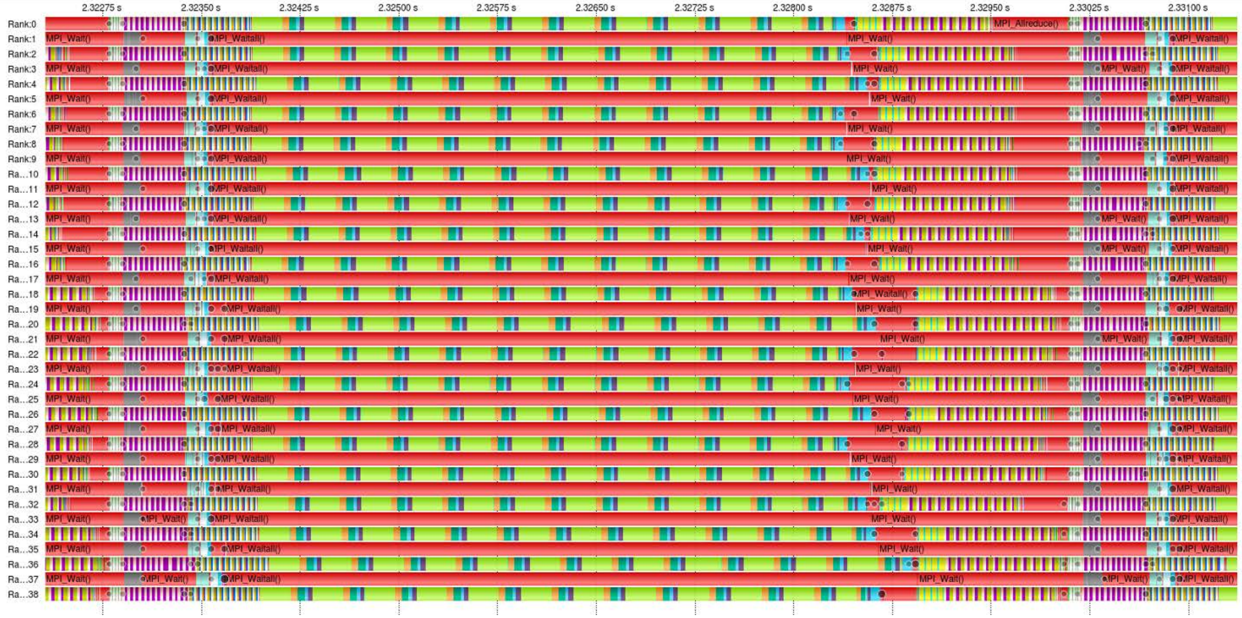
Getting Started:
Building Your Own sNIC-Accelerated Applications - Tips

- Analyze potential cost/benefits of kernel offloading within the application
  - I.E. What can be offloaded? What are the ramifications?
- Plan mapping of host and sNIC processes
  - MPI/SHMEM + X
    - Simple, 1 PE per device
  - MPI/SHMEM within a node
    - 1:1 mapping is simplest
      - Odd/Even or Upper/lower half schemes
    - May not utilize all available resources
- Track host and sNIC memory consumption/requirements
  - For SHMEM applications, monitor the symmetric heap
- Data movement between host and sNIC is expensive
  - Minimize and overlap with computation wherever possible
- Monitor dependencies carefully

Getting Started: Building Your Own sNIC-Accelerated Applications - Tips

- Thoroughly analyze your target application beforehand
  - TAU
  - CrayPat
- Profiling is good, tracing is better
- Also analyze your sNIC-enabled variant to avoid unnecessary head-scratching





Getting started – Compiling and Running

- LANL/USRC SmartNIC Platforms
  - Ghost & Nymeria
    - 2x Xeon E5-2687W v4 Processors
      - 12 cores each
    - 128 GB RAM
    - CentOS 7
  - Mellanox Bluefield SmartNICs
    - 16 ARMv8 Cortex-A72 Cores
    - 16 GB RAM
    - Cent OS 7
    - Accessible via SSH from host
  - Broadcom Stingray SmartNICs
    - Forthcoming

Getting started – Compiling and Running

- SHMEM and/or MPI are needed for communication between the host and SmartNIC
  - Different ISAs mandate distinct binaries
  - x86 binaries can be installed on the NFS
  - aarch binaries on the local SmartNIC file system

- For simplicity's sake, we built OpenMPI (+ OpenSHMEM) v4.0.4 on top of UCX v1.8.0 (release tarballs)
  - First configure & install UCX

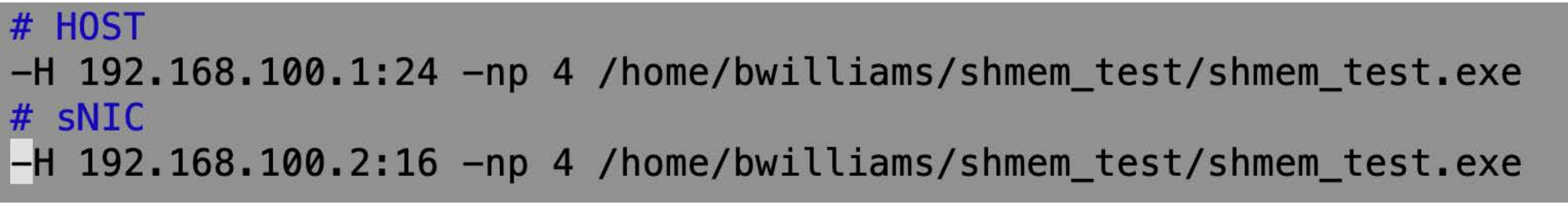
    -
      ```
      ./configure --prefix=/path/to/install --enable-cma --enable-mt
      --disable-numa
      ```
  - Then configure & install OMPI/OSHMEM

    -
      ```
      ./configure --prefix=/path/to/install --with-ucx=/path/to/ucx/install
      --with-hwloc=internal --with-slurm=no --with-zlib=no --with-verbs=no
      ```
  - The same configuration options should be utilized for the host and SmartNIC

Getting started – Compiling and Running

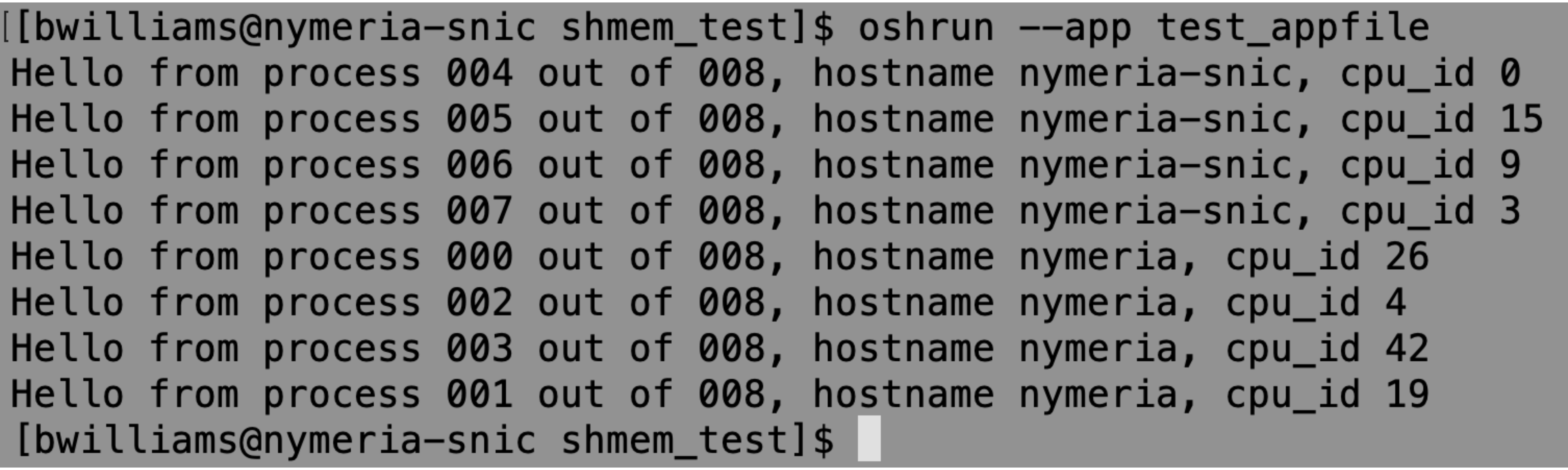
- SSH publickey authorization between the host and SmartNIC is necessary to facilitate interprocess communication
  - If needed, generate a new key:
    - `ssh-keygen -t rsa`
  - Mark the key as trusted on the destination:
    - `ssh-copy-id -i ~/.ssh/id_rsa.pub 192.168.100.1 (sNIC -> Host)`
  - Must be done in both directions!

- Running a cross-device job with OMPI/OSHMEM
  - Must be launched from the SmartNIC
  - You can use –H switch directly or an appfile



```
# HOST
-H 192.168.100.1:24 -np 4 /home/bwilliams/shmem_test/shmem_test.exe
# sNIC
-H 192.168.100.2:16 -np 4 /home/bwilliams/shmem_test/shmem_test.exe
```

Getting started – Test Output



```
[[bwilliams@nymeria-snic shmem_test]$ oshrun --app test_appfile
Hello from process 004 out of 008, hostname nymeria-snic, cpu_id 0
Hello from process 005 out of 008, hostname nymeria-snic, cpu_id 15
Hello from process 006 out of 008, hostname nymeria-snic, cpu_id 9
Hello from process 007 out of 008, hostname nymeria-snic, cpu_id 3
Hello from process 000 out of 008, hostname nymeria, cpu_id 26
Hello from process 002 out of 008, hostname nymeria, cpu_id 4
Hello from process 003 out of 008, hostname nymeria, cpu_id 42
Hello from process 001 out of 008, hostname nymeria, cpu_id 19
[bwilliams@nymeria-snic shmem_test]$ 
```

Acknowledgements and Links

- I want to thank the following:
  - Wendy & Steve Poole – LANL
  - Liliana Aguirre-Esparza – NMSU
  - Parks Fields – LANL/USRC
  - Jack Snyder – Duke University
  - Mellanox Team
  - Charles Ferenbaugh – LANL

- More information about the UCF Consortium and its SmartNIC projects can be found at the links below.
  - UCF Consortium: https://www.ucfconsortium.org/
  - OpenSNAPI: https://www.ucfconsortium.org/projects/opensnapi/
    - Press release: https://www.businesswire.com/news/home/20200623005105/en/UCF-Consortium-Announces-OpenSNAPI-Project-Develop-Open
  - OpenHPCA: https://www.ucfconsortium.org/projects/hpca-benchmark/

Thank You!

Questions?