

# An HPEC Retrospective: Development and application of a hybrid programming environment on an ARM/DSP system for High Performance Computing

---

**Gaurav Mitra**

gmitra@fb.com

OpenSuperComputing BoF, HPEC

Sep 22, 2021

## Disclaimer

---

*This talk and its contents, slides, collateral represent my own thoughts and research done prior to 2018 and do not represent any work done or research conducted at Facebook at any point of time. Research conducted in this project and covered in this talk was done prior to my joining Facebook.*

## Development and application of a hybrid programming environment on an ARM/DSP system for High Performance Computing

Gaurav Mitra  
Texas Instruments Inc.,  
Texas, USA  
gaurav@ti.com

Jonathan Bohmann  
Southwest Research Institute,  
Texas, USA  
jonathan.bohmann@swri.org

Ian Lintault  
nCore HPC,  
Petaluma, USA  
ian@ncorehpc.com

Alistair P. Rendell  
Australian National University,  
Canberra, Australia  
Alistair.Rendell@anu.edu.au

**Abstract**—The nCore *Brown-Dwarf* system has a unique architecture where each node is comprised of two different low-power System-on-Chip (LPSoC) processors from Texas Instruments; the ARM/DSP Keystone II SoC and the DSP based Keystone I SoC. These LPSoC processors have, through use of the C66x multi-core DSP, been shown to be capable of running floating-point intensive HPC application codes. However, it is non-trivial to run such codes across all processing elements of node simultaneously. This paper demonstrates a hybrid programming environment that combines OpenMP, OpenCL and MPI to enable application execution across multiple Brown-Dwarf nodes. This environment is evaluated using two diverse application codes. The first is Level-3 BLAS matrix multiplication (GEMM), which is a standard HPC floating-point intensive benchmark. The second is a unique real-world scientific code for biostructure based drug design developed by the Southwest Research Institute called *Rhodium*<sup>TM</sup>. Performance and energy-efficiency of Rhodium<sup>TM</sup> is presented alongside comparisons with conventional x86 based HPC systems with attached accelerators. Results indicate that the Brown-Dwarf system remains competitive with contemporary systems for memory-bound computations.

**Index Terms**—ARM, C66x, DSP, Keystone II, Brown-Dwarf, Protein Docking, Drug Design, Protein Engineering, Accelerator, OpenMP, OpenCL, Energy Efficiency

### I. INTRODUCTION

Hybrid programming environments are increasingly popular

HPC manufacturers such as Fujitsu [1], Cray [2] and HPE [3] have announced systems comprising ARM SoCs. An early example of such an ARM based SoCs is the Texas Instruments Keystone II (K2). More importantly, the K2 SoCs showed how on-chip accelerators such as multi-core Digital Signal Processors (DSP) could also be used for HPC application codes providing significantly higher performance compared to the host ARM CPU cores. Given the importance of ARM for HPC, an assessment of the suitability of ARM CPUs in conjunction with unconventional accelerators such as DSPs for HPC application codes provides an opportunity for timely, critical, relevant and good research.

The first commercially available HPC system to integrate Keystone SoCs was the nCore Brown-Dwarf released in 2013 and remains one of the few available heterogeneous systems integrating ARM CPUs with DSP accelerators. Each Brown-Dwarf node combined a single K2 ARM/DSP SoC and two Keystone I (K1) DSP SoCs together using a 50GigaBaud point-to-point interconnect known as Hyperlink. The initial programming environment available for use on a Brown-Dwarf node consisted of OpenCL and OpenMP 4.0 for running applications across the ARM and DSP cores on the K2 SoC. A means of communication to the K1 DSP from the K2 ARM cores, while simultaneously orchestrating program execution

## Motivation

Brown-Dwarf: SoC, Hardware, Communication Framework

Building and executing a hybrid fat binary on Brown-Dwarf

Applications: Partitioning and Implementation for Brown-Dwarf

Results: Overheads, DMA Bandwidth, GEMM & Rhodium Performance

Energy Efficiency Analysis

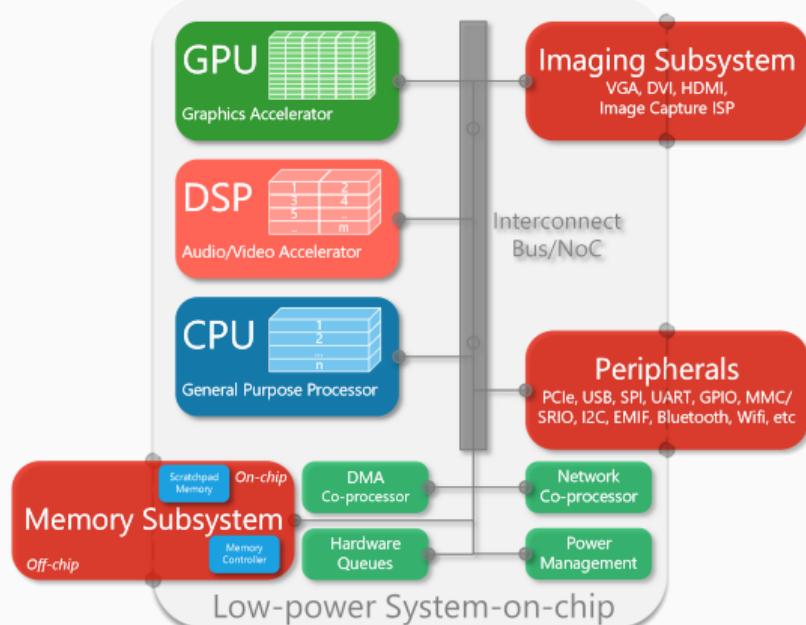
Conclusion & Future Work

## Motivation

---

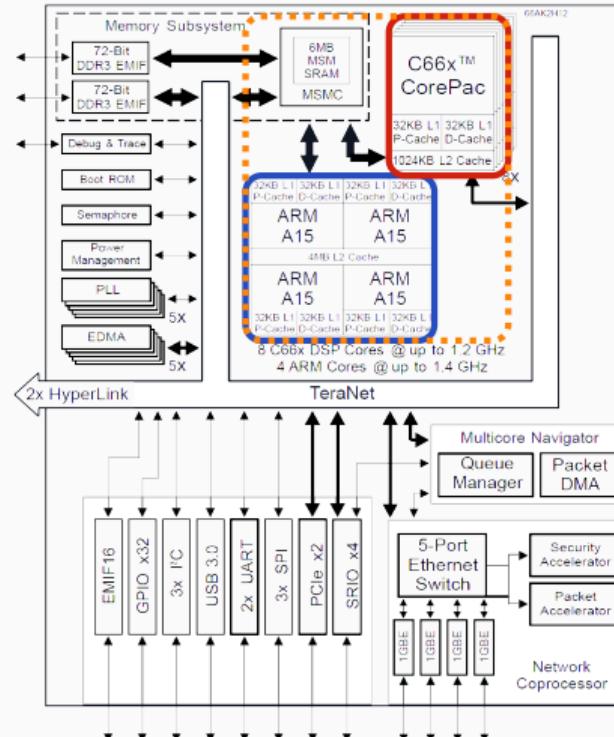
# Motivation: Heterogeneity

- Dominance of accelerators such as GPUs used for HPC
- Ubiquity of multi-core CPUs, multi-node systems with distributed memory
- Mass market success of Low-power System-on-Chip (LPSoC) processors
- Rising adoption of ARM in HPC
- On-chip accelerators on LPSoCs remain largely unexploited
- Low-power → energy-efficient?
- Energy consumption a huge (only?) problem for Exascale HPC
- Heterogeneous systems with attached accelerators dominate Green500 list



# Motivation: ARM + Accelerator for HPC?

- **TI Keystone II SoC:** An early ARM based SoC with unconventional DSP accelerator
- Are heterogeneous ARM SoCs suitable for HPC workloads?
- How do such ARM SoCs compare against conventional HPC systems in terms of both power and performance?
- Is it possible to effectively partition work across all processing elements (PE)?
- What is a typical programming environment for a commercial ARM based HPC system?



## Contributions

---

Hybrid programming environment for an early ARM based HPC system: **nCore Brown-Dwarf**

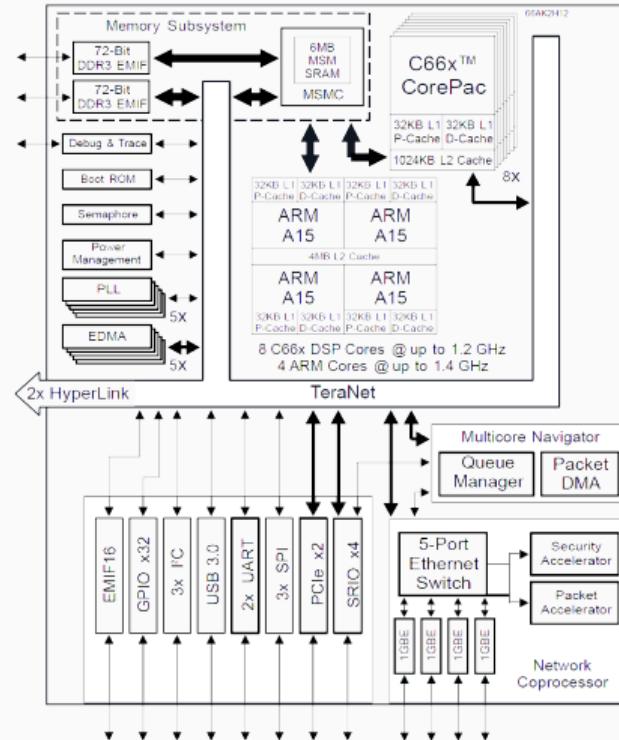
- Software design
  - Message passing communication framework
  - Creation of hybrid self-extracting fat binary
  - Use of hybrid OpenMP/OpenCL/MPI programming model to execute across all PEs
- Proof-of-concept implementation
- Implementing benchmark and real-world application using environment on Brown-Dwarf
- Evaluation of environment using implemented applications
- Performance and energy efficiency comparison with conventional Intel+NVIDIA HPC system

# **Brown-Dwarf: SoC, Hardware, Communication Framework**

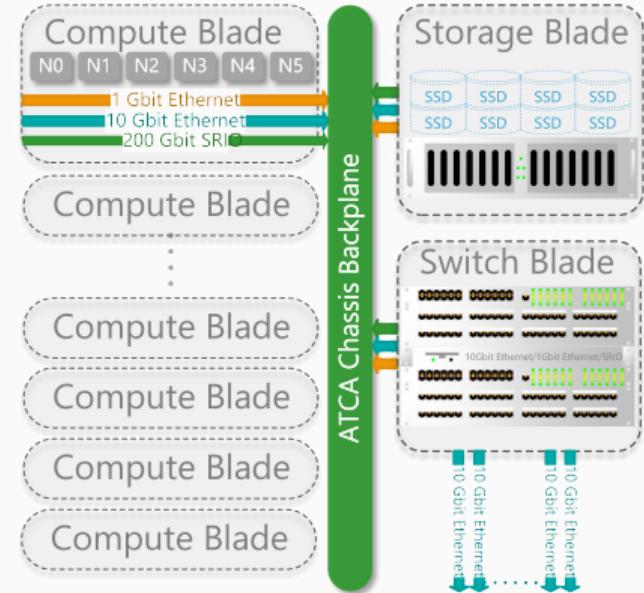
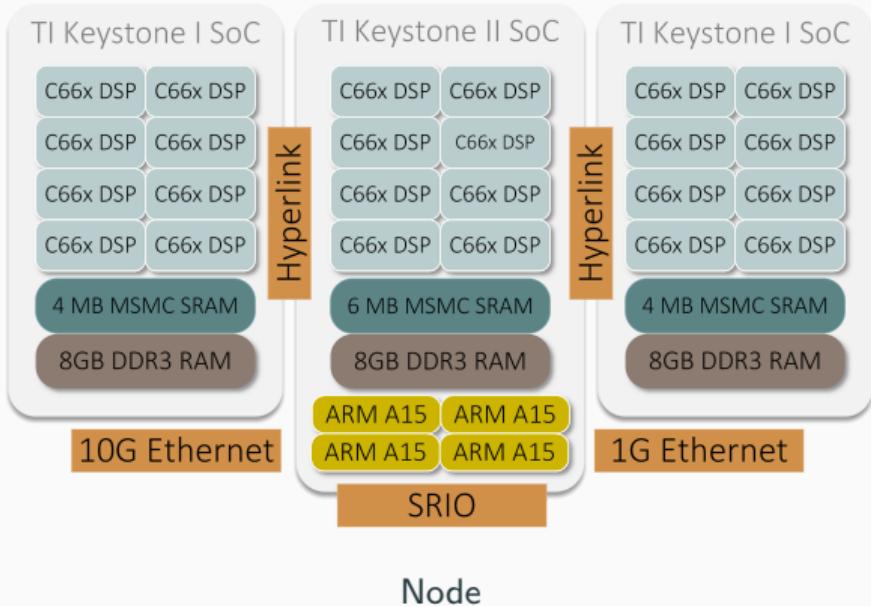
---

# The Texas Instruments Keystone II LPSoC

- Four ARM Cortex-A15 Cores
  - Up to 1.4 GHz
  - 4MB L2 shared cache
  - 32KB L1P and L1D cache per core
  - (38.4 SP, 9.6 DP) GFLOPS
- Eight C66x DSP Cores
  - Up to 1.25 GHz
  - 32K L1P, 32K L1D, 1M L2 per core
  - (160 SP, 40 DP) GFLOPS
- Hardware queues with atomic access
- 6MB Fast shared (between ARM, DSP) scratchpad memory
- Power consumption ~ 15 Watts TDP

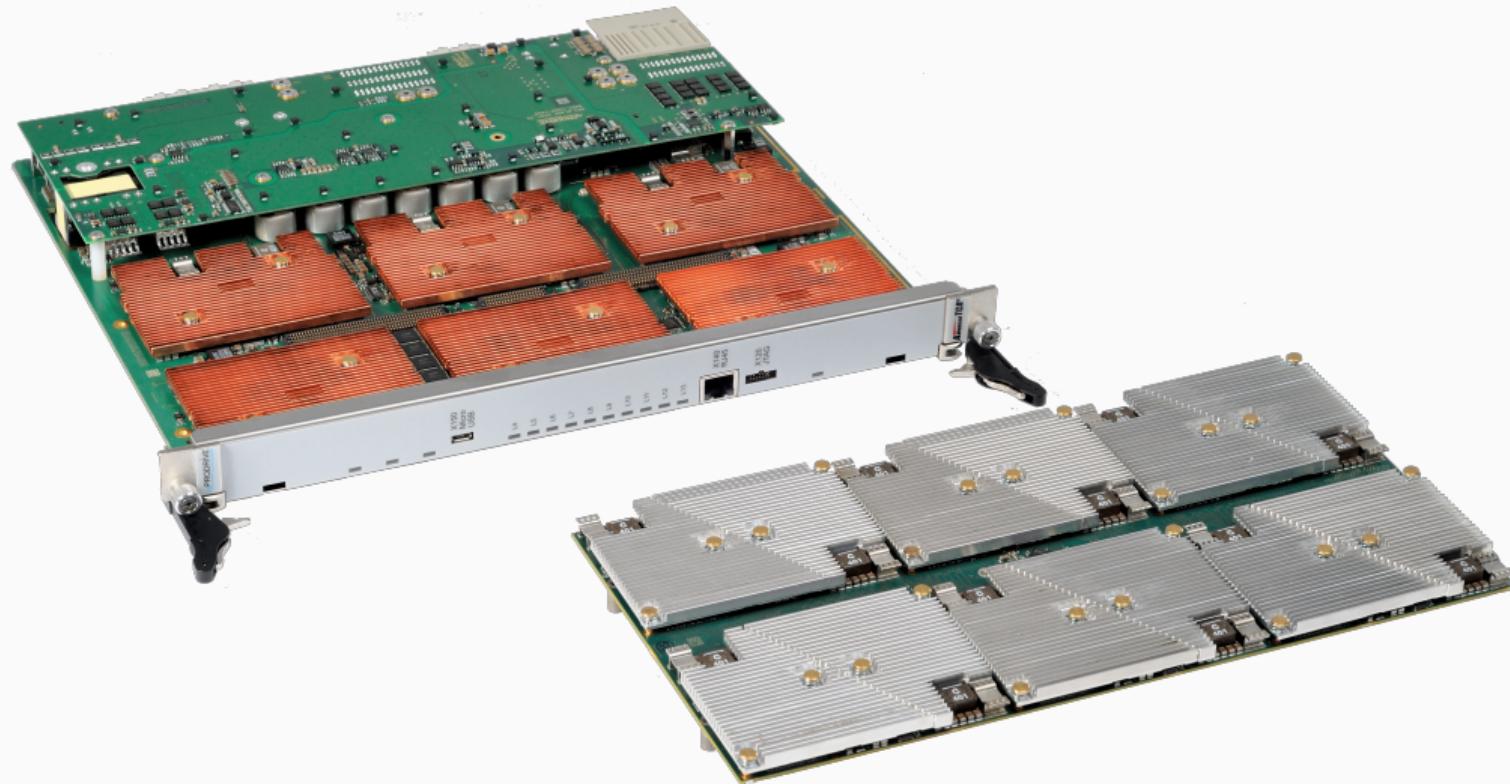


# nCore Brown-Dwarf System

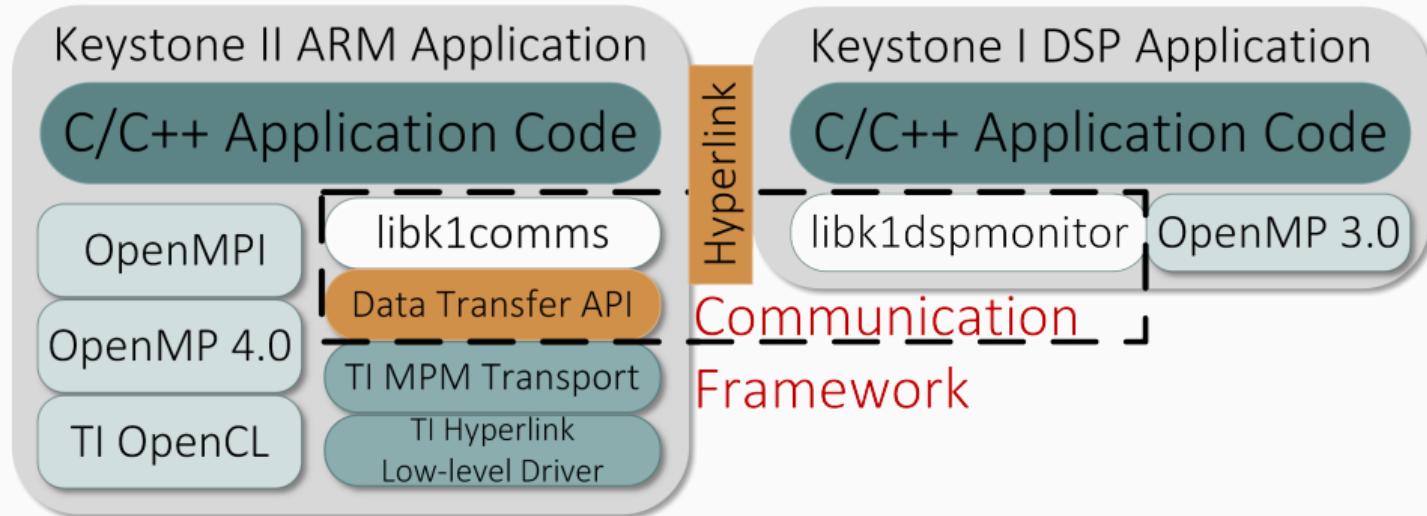


## Blade and Chassis Configuration

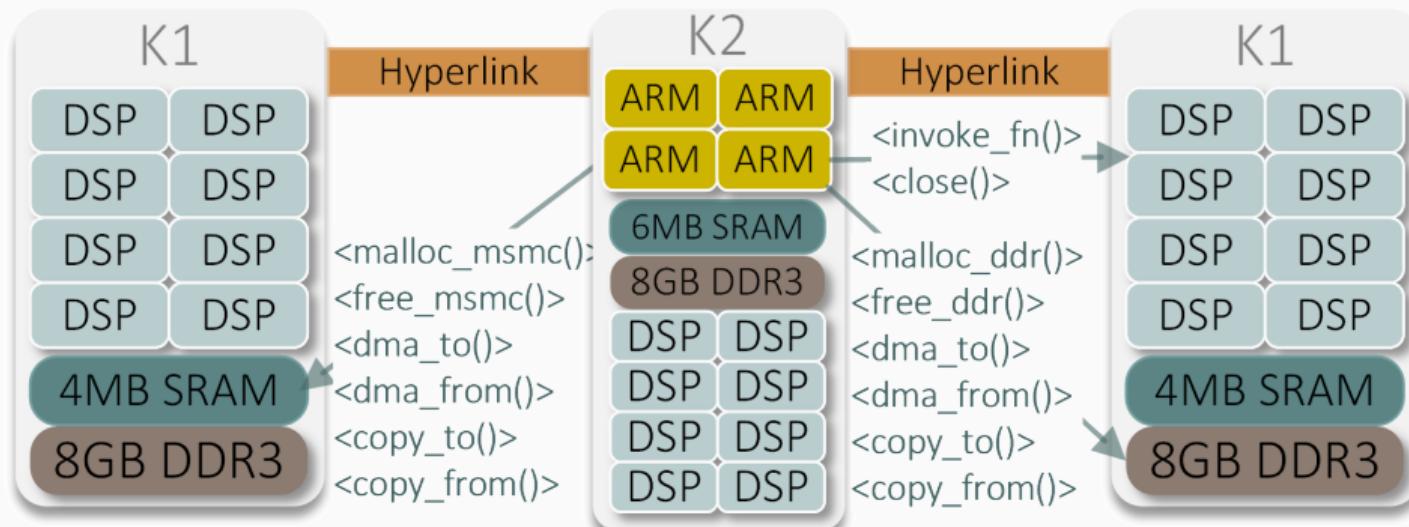
# nCore Brown-Dwarf Compute Blade



# Application Software Stack



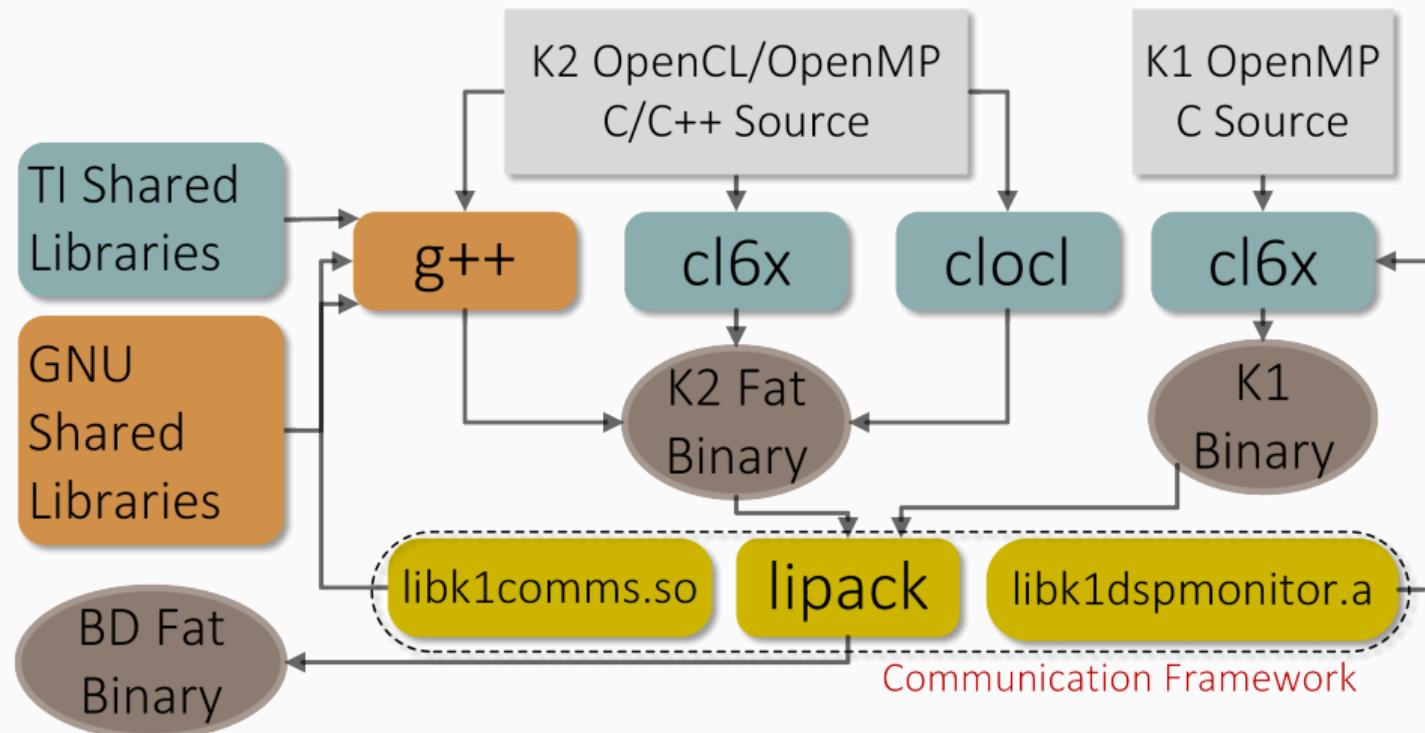
# Data transfer and work offload API



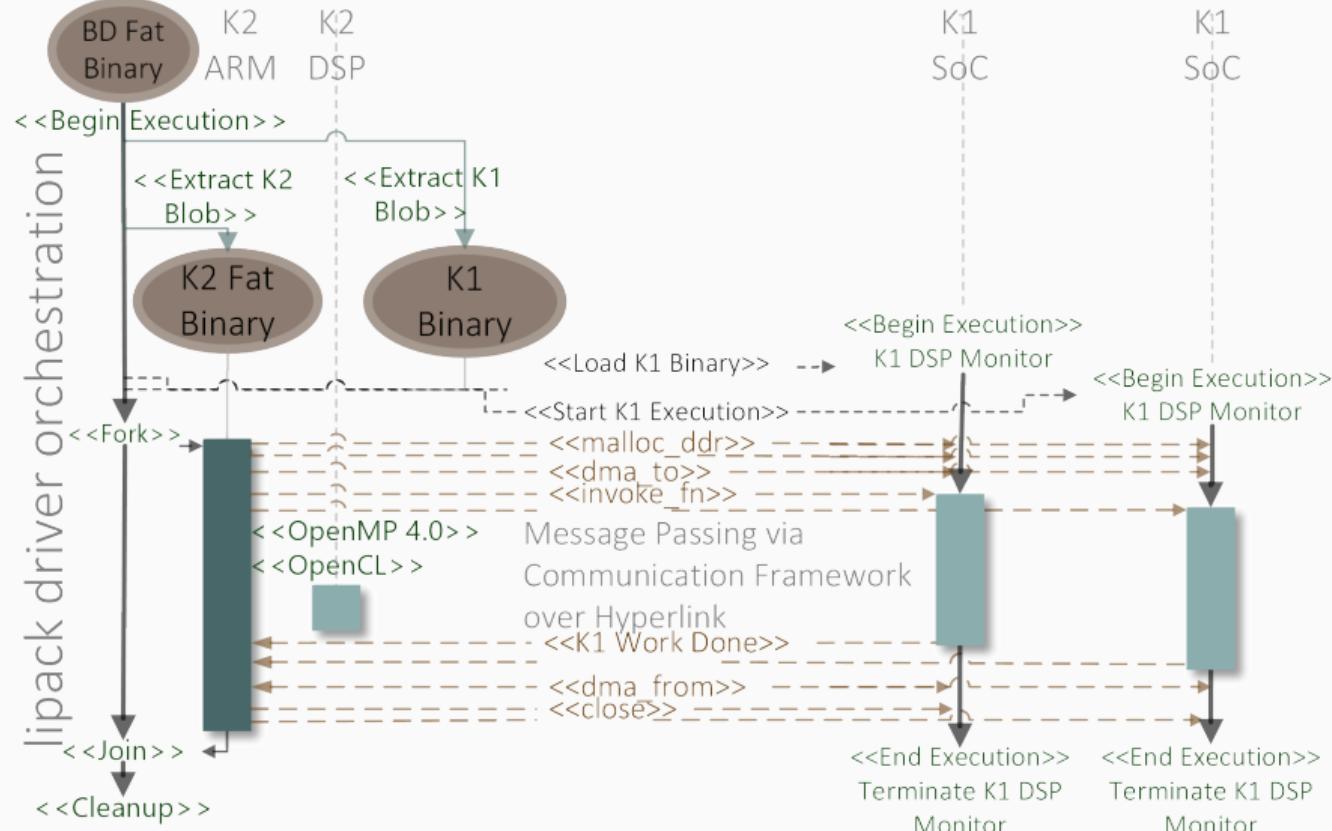
## **Building and executing a hybrid fat binary on Brown-Dwarf**

---

## Hybrid Fat Binary: Construction



# Hybrid Fat Binary: Execution



## **Applications: Partitioning and Implementation for Brown-Dwarf**

---

## Applications implemented for Brown-Dwarf

- L3 BLAS: Matrix Multiplication (GEMM)
  - Fundamental to most scientific computing
  - Traditional floating-point benchmark  $\sim O(n^3)$
- The Rhodium™ protein docking program
  - Biostructure-based drug design simulation i.e. *docking*
  - Hypothetical drug molecules are docked or *matched* to a protein *target*
  - The level of match is expressed as a *docking score*, docked location is known as a *pose*
  - Three computationally intensive stages (in increasing order of complexity):
    1. Establishment of search grid
    2. Low-resolution search: Generation of trial poses at each grid point with simple scoring
    3. High-resolution search: Refinement of selected poses by Multi-directional Search (MDS) method
  - Computational characteristics similar to single-precision n-body simulations  $\sim O(n^2)$
  - Two different workloads benchmarked
    1. COX: Drug-protein  $\sim 10^5$  atomic pairwise interactions
    2. 1BRS: Protein-protein,  $\sim 10^7$  atomic pairwise interactions, memory-bound

## GEMM: Work Partition

$$\mathbf{C} = \mathbf{A} \times \mathbf{B}$$

$$\left[ \begin{array}{c|c} \mathbf{C}_1 & \mathbf{C}_2 \end{array} \right] = \left[ \begin{array}{c} \mathbf{A} \end{array} \right] \times \left[ \begin{array}{c|c} \mathbf{B}_1 & \mathbf{B}_2 \end{array} \right]$$


$$\mathbf{C}_1 = \mathbf{A} \times \mathbf{B}_1$$

$$\left[ \begin{array}{c} \mathbf{C}_1 \end{array} \right] = \left[ \begin{array}{c} \mathbf{A} \end{array} \right] \times \left[ \begin{array}{c} \mathbf{B}_1 \end{array} \right]$$

Device X

$$\mathbf{C}_2 = \mathbf{A} \times \mathbf{B}_2$$

$$\left[ \begin{array}{c} \mathbf{C}_2 \end{array} \right] = \left[ \begin{array}{c} \mathbf{A} \end{array} \right] \times \left[ \begin{array}{c} \mathbf{B}_2 \end{array} \right]$$

Device Y

## GEMM: Work Partition

- The amount of work partitioned between different PEs is critical
- ADaptive Work PARTitioning Algorithm: ADAPART(X,Y)
  1. Divide work equally between devices X and Y
  2. Run computation on each device and measure time for each device
  3. Based on each device performance, give more work to better performing device
  4. Repeat until convergence or steady state when both devices finish together
- Apply ADAPART in sequence to get optimal partition between all PEs in a single Brown-Dwarf node
  1. ADAPART(K2 ARM, K2 DSP): K2 SoC
  2. ADAPART(K2 SoC, K1 SoC): K2 SoC + K1 SoC
  3. ADAPART(K2 SoC + K1 SoC, K1 SoC): Brown-Dwarf Node
- Divide work equally between multiple Brown-Dwarf nodes using MPI

# GEMM: Implementation

```
#pragma omp parallel default(shared) num_threads(4){ /* On a single Brown-Dwarf Node */
    #pragma omp single nowait{
        #pragma omp task/* Keystone II ARM: Using BLIS/ATLAS */
        if (node_work_distribution[K2H_ARM] > 0){
            cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, M, node_work_distribution[K2H_ARM], K,
                        alpha,
                        A, /* lda = */ M, B + node_start_list[K2H_ARM]*K, /* ldb = */ K,
                        beta, C + node_start_list[K2H_ARM]*M, /* ldc = */ M); }
    }

    #pragma omp task/* Keystone II DSP: Using OpenMP 4.0 */
    if (node_work_distribution[K2H_DSP] > 0){
        dsp_cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, M, node_work_distribution[K2H_DSP],
                        K, alpha,
                        A, /* lda = */ M, B + node_start_list[K2H_DSP]*K, /* ldb = */ K,
                        beta, C + node_start_list[K2H_DSP]*M, /* ldc = */ M); }
}

#pragma omp task/* Keystone I DSP: Using BD Communication Framework */
if (node_work_distribution[K1_SHN0] > 0){
    cblas_dgemm_k1_compute(SOC_SHN0); }

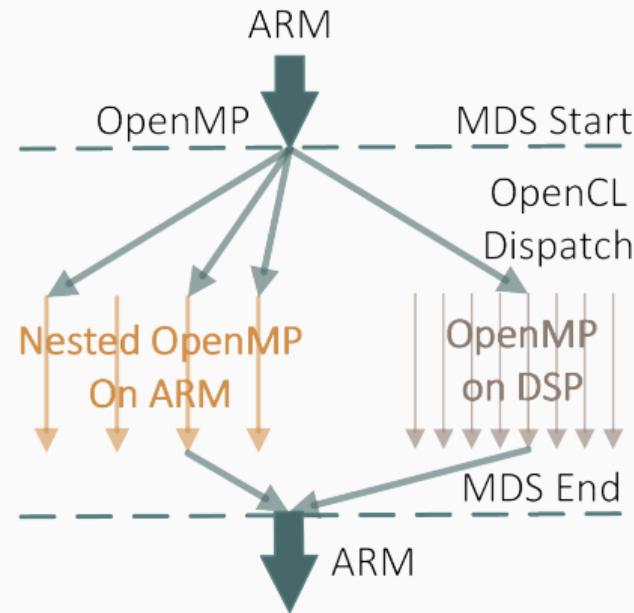
#pragma omp task/* Keystone I DSP: Using BD Communication Framework */
if (node_work_distribution[K1_SHN1] > 0){
    cblas_dgemm_k1_compute(SOC_SHN1); }

}
}

#pragma omp taskwait
```

# Rhodium: Work Partition and Implementation

- Search grid established using ARM cores
- Low-resolution OpenCL kernel executed on K2 DSP
- MDS implementation distributed across all PEs on a node
  - K2 ARM: OpenMP 3.0 kernel
  - K2 DSP: OpenCL dispatch with OpenMP 3.0 kernel (TI extension)
  - K1 SoC: BD communication framework dispatch with OpenMP 3.0 kernel
- ADAPART does not apply apriori since MDS iterations can have different computational characteristics
- Each PE given equal work in first iteration
- In each successive iteration give more work to better performing PE with goal of minimizing idle time and coordinating finish time
- Divide work equally between multiple Brown-Dwarf nodes using MPI



## **Results: Overheads, DMA Bandwidth, GEMM & Rhodium Performance**

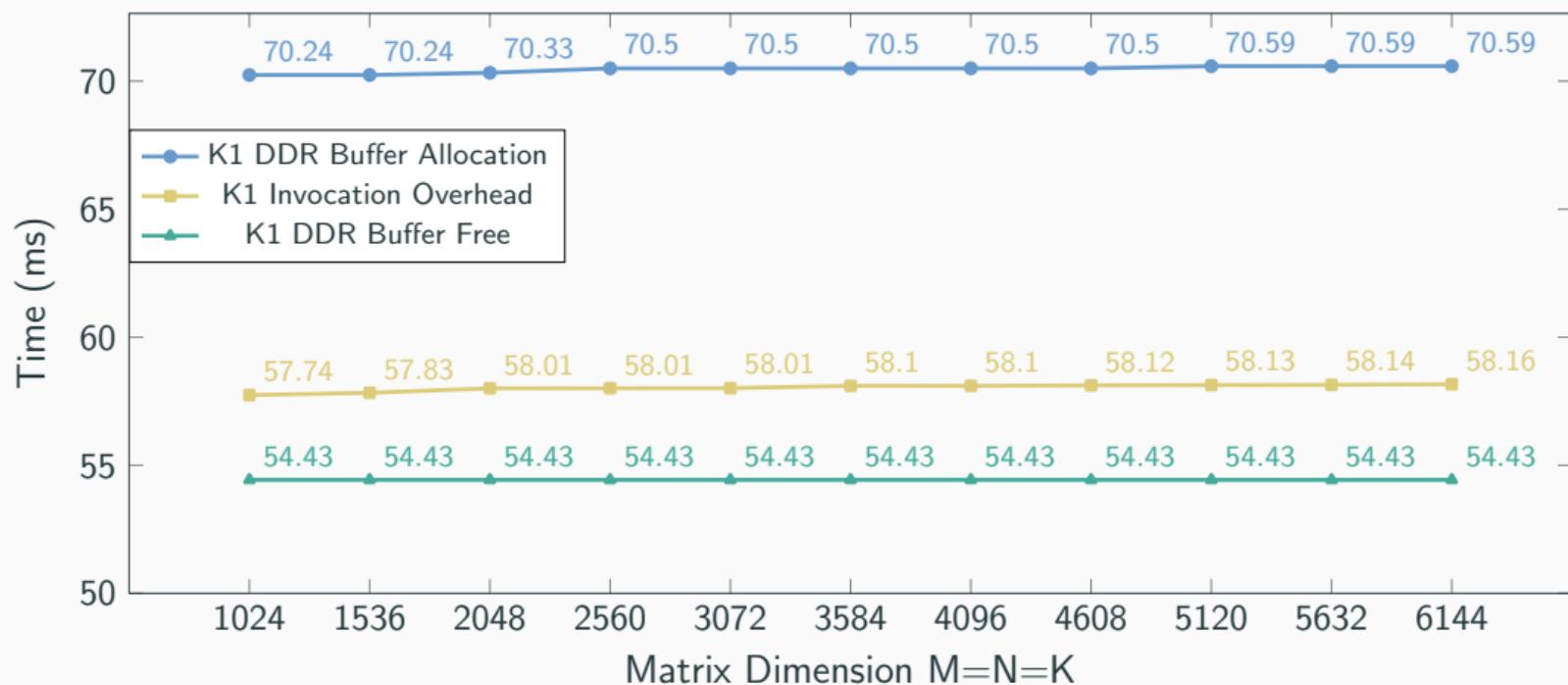
---

## Results: Environment

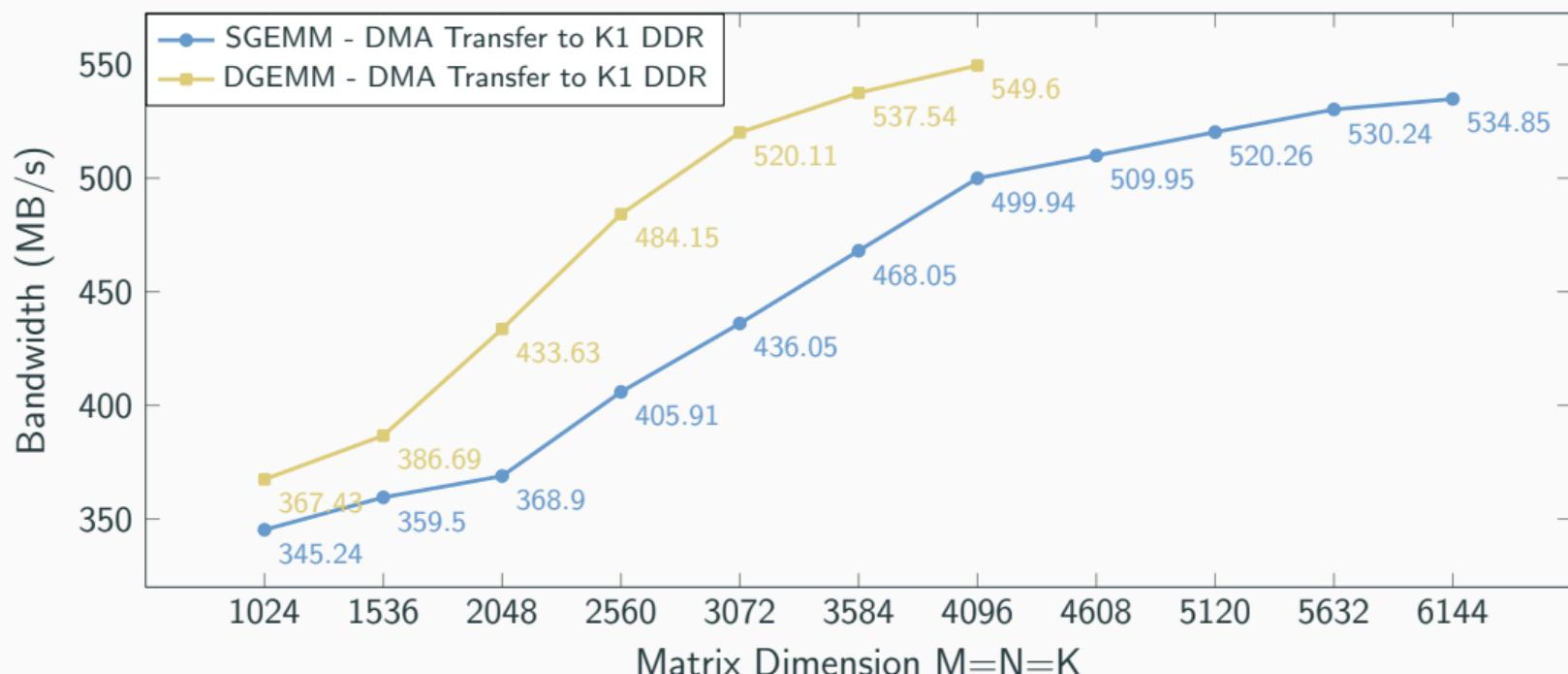
---

- Both applications benchmarked across 12 Brown-Dwarf nodes
- SGEMM and DGEMM benchmarked in GFLOPS
  - Communication framework overhead measured
  - DMA data transfer bandwidth across Hyperlink measured
  - Strong scaling (adding resources while keeping constant problem size) results reported
- Rhodium™ benchmarked in Datasets/Day
- Rhodium™ has a *task-parallel* mode: Molecules distributed across 12 nodes and run on a one-job-per-node basis, instead of splitting each job across 12 nodes
- Conventional HPC System used to evaluate Rhodium™ results
  - Intel IvyBridge 24-core Xeon E5-2695 v2 CPU: Running grid establishment and MDS
  - Two attached NVIDIA K20m GPU accelerators: Running low-resolution OpenCL kernel
- Brown-Dwarf ARM OS
  - Debian Linux filesystem with Linux kernel 3.10.72
  - ARM GCC 4.9.2, TI DSP CGT (cl6x) 8.1.2, TI OpenCL 1.1, TI OpenMP 2.03.01.00

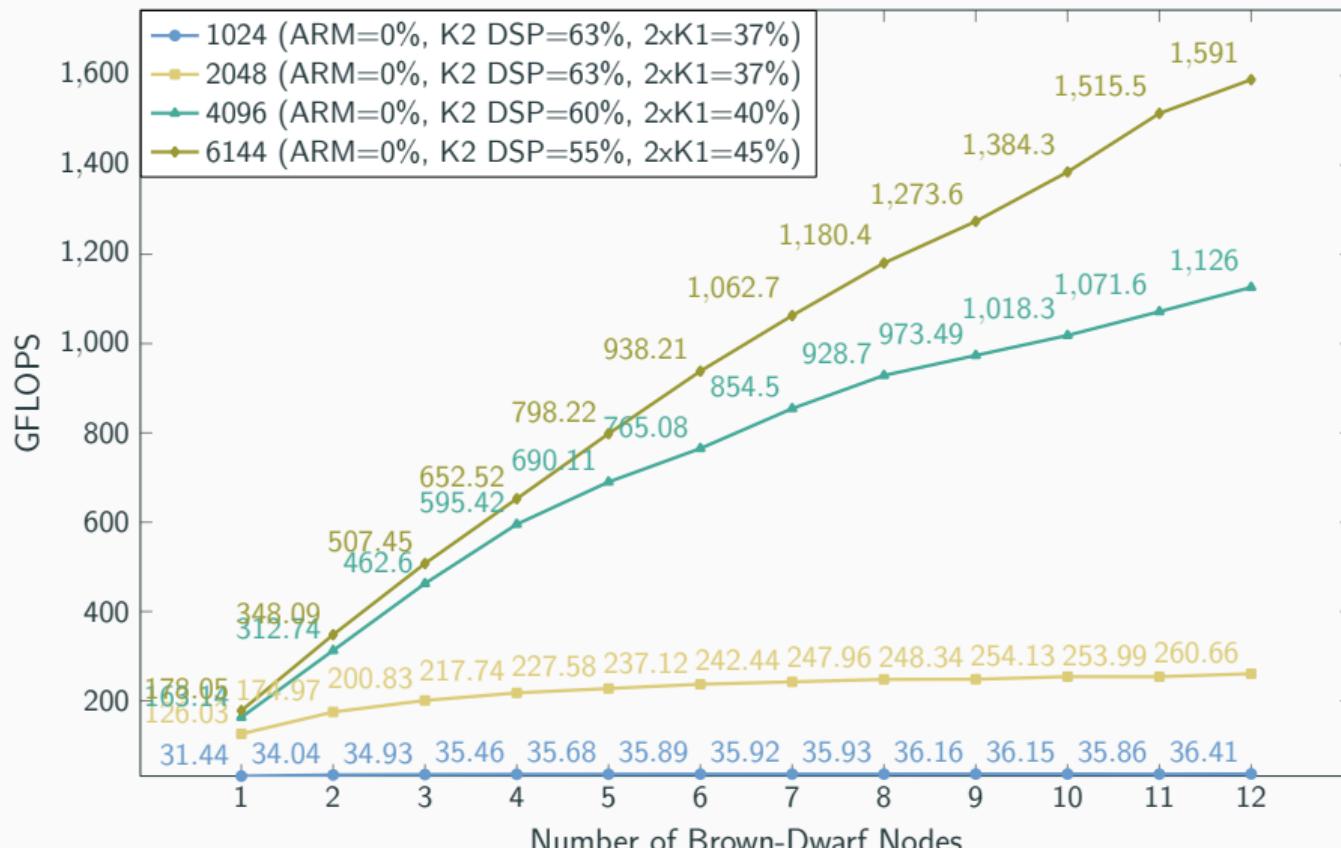
## Results: Communication Overhead



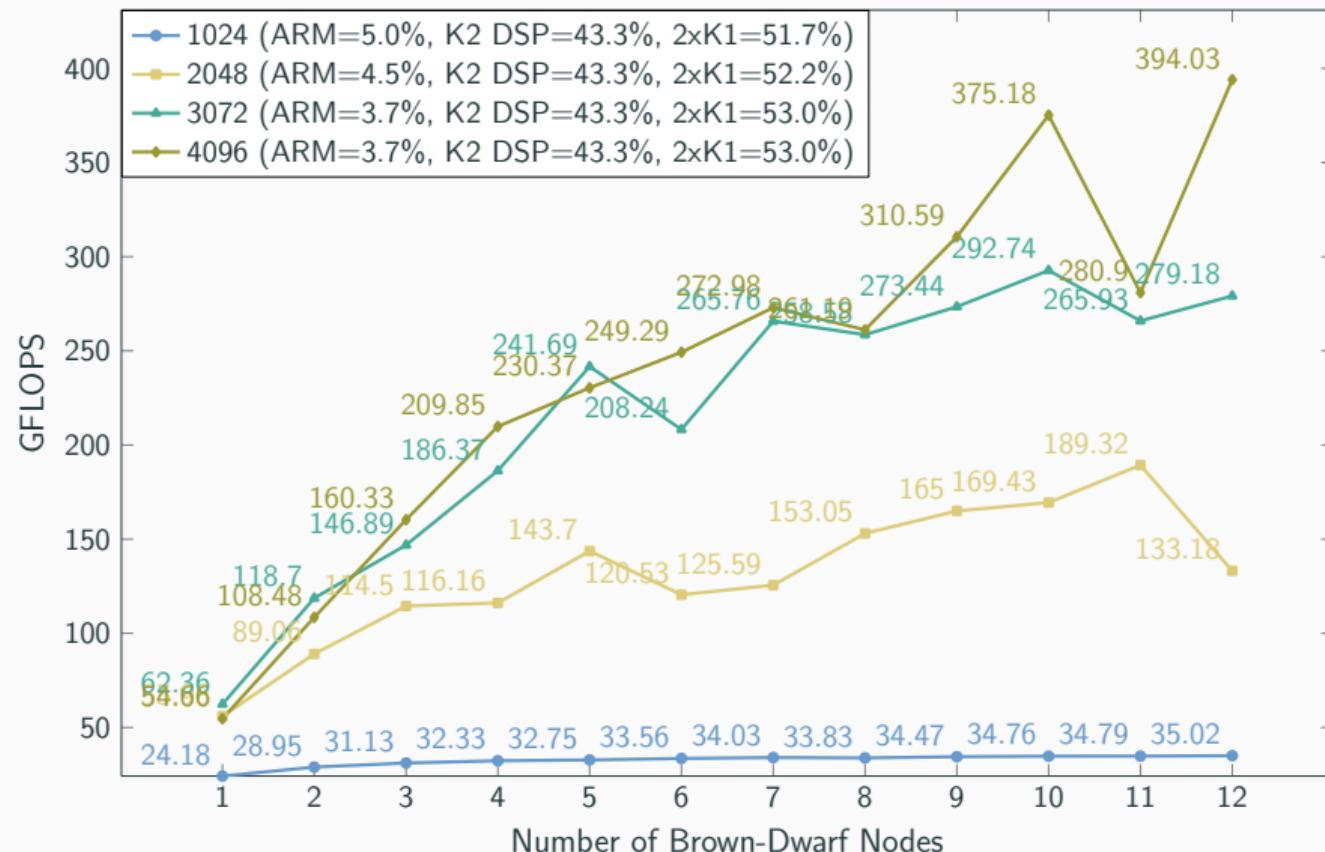
## Results: DMA Bandwidth



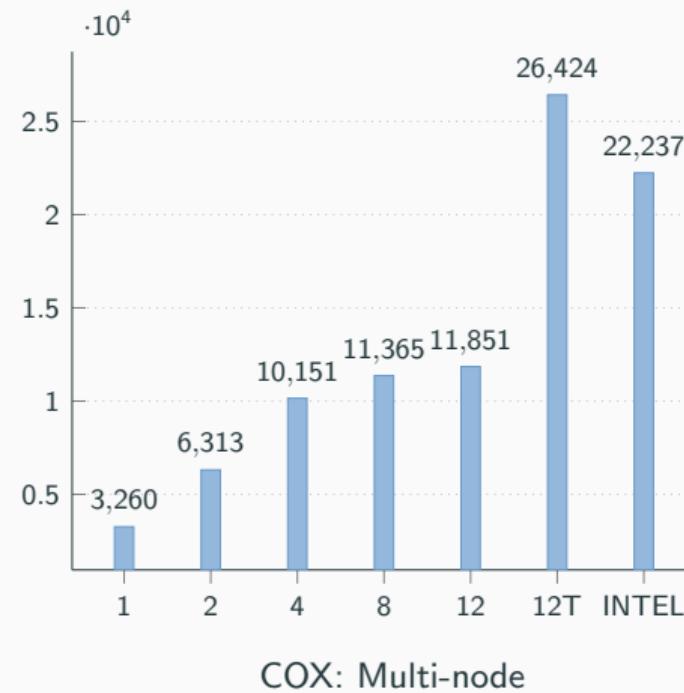
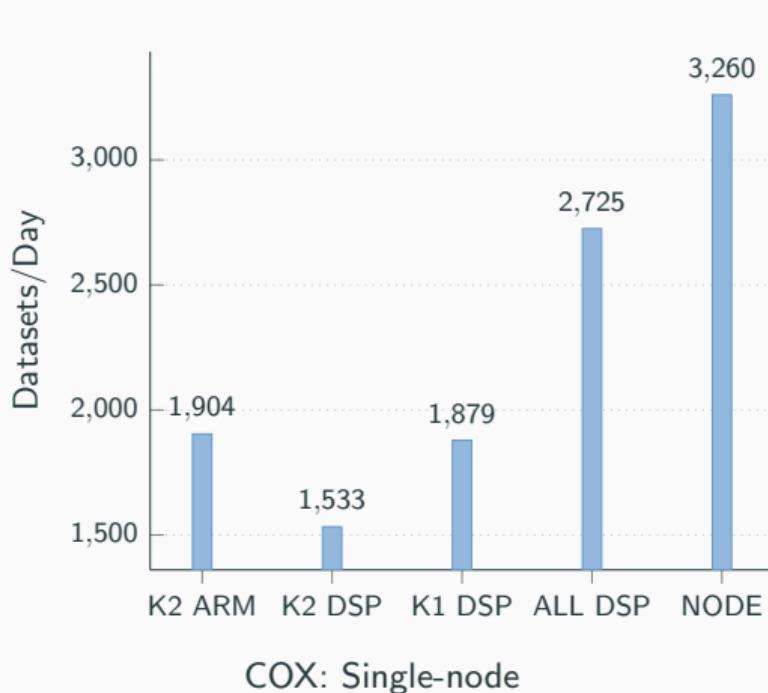
## Results: SGEMM Strong Scaling Performance



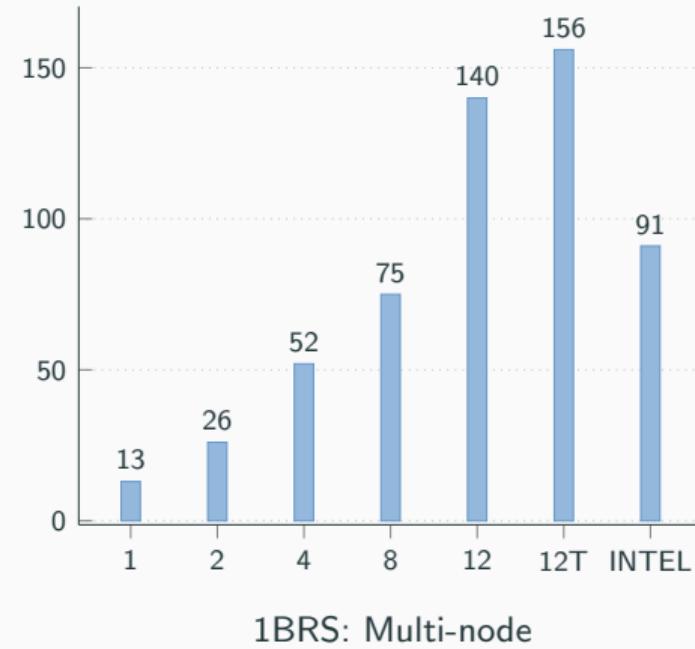
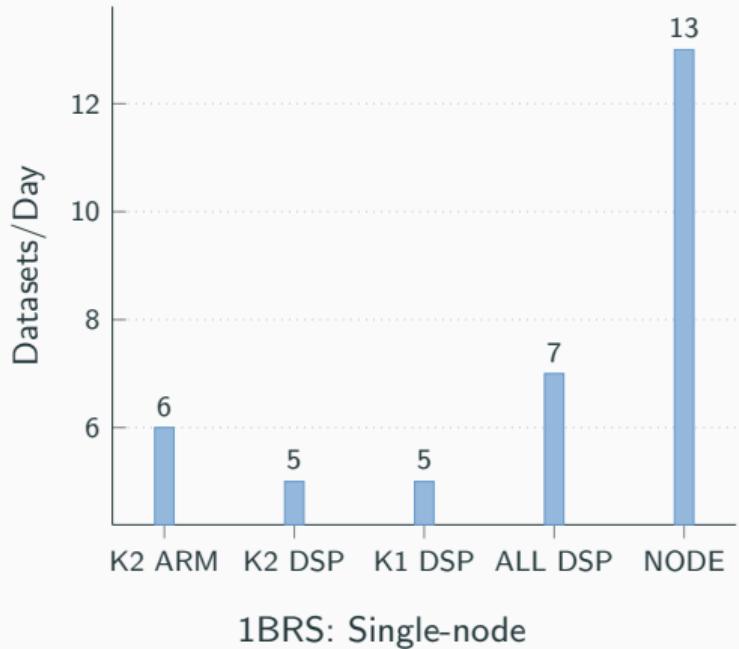
## Results: DGEMM Strong Scaling Performance



# Rhodium: Performance



# Rhodium: Performance



# Energy Efficiency Analysis

---

## Energy-efficiency: Measuring Energy

---

- 12-node Brown-Dwarf system:
  - Brown-Dwarf ATCA chassis has blade-level current sensors
  - These sensors report threshold values rather than instantaneous values
  - 6-node (1 blade) Idle Power: 252.6 Watts
  - 6-node (1 blade) Load Power: 267.1 Watts
  - **12-node (2 blade) Load Power: 534.2 Watts**
  - Does *not* include power consumption of switch blade or storage blade
- Intel+NVIDIA system
  - Component thermal design power (TDP) used for comparison
  - CPU TDP: 115 Watts, GPU TDP: 225 Watts
  - **Combined system TDP: 565 Watts**
  - Does *not* include other system components such as RAM, motherboard etc

## Energy-efficiency: GEMM & Rhodium

---

- Highest efficiency numbers measured on 12-node Brown-Dwarf system:
  - SGEMM: 2.98 GFLOPS/Watt
  - DGEMM: 0.74 GFLOPS/Watt
  - Rhodium (1BRS 12T): **0.29 Datasets/Day/Watt**
- Intel+NVIDIA system
  - Rhodium (1BRS): **0.16 Datasets/Day/Watt**
- Top system on Green500 List (PEZY-SC2 Shobu system B):
  - LINPACK: 17.009 GFLOPS/Watt

## Energy Efficiency: Analysis

---

- The 2010 Keystone I and 2012 Keystone II architectures had an energy efficiency  $1.8\times$  higher than the contemporary Intel+NVIDIA system for the 1BRS Rhodium dataset
- The Brown-Dwarf system and the TI Keystone architecture are not likely candidates for future exascale systems
- However, for real-world memory-bound datasets such as 1BRS, the Brown-Dwarf system remains competitive

## **Conclusion & Future Work**

---

# Conclusion

---

- A novel hybrid programming environment and communication framework for use with HPC applications on the nCore Brown-Dwarf ARM/DSP system
- Key aspects of a typical software stack required for such heterogeneous architectures
- A mechanism to create a hybrid self-extracting fat binary which can initiate execution on ARM and DSP cores that span SoCs across multiple nodes.
- Key Observations
  1. ARM based heterogeneous systems are suitable for real-world HPC computations
  2. Absolute performance of ARM based systems can be maximized by utilizing all PEs across multiple nodes
  3. Energy-efficiency of the Brown-Dwarf system and TI Keystone architecture remains competitive with contemporary HPC systems for memory-bound computations
- The Brown-Dwarf system at Southwest Research Institute continues to operate

## Future Work

- Implementation of other scientific application codes for Brown-Dwarf
- Use of SRIO networks on larger Brown-Dwarf systems
- Adapting this hybrid programming environment to another SoC platform to evaluate portability

Contact:

- [gmitra@fb.com](mailto:gmitra@fb.com)
- <https://www.linkedin.com/in/gaurav-mitra-770b334>

**Thank you!**