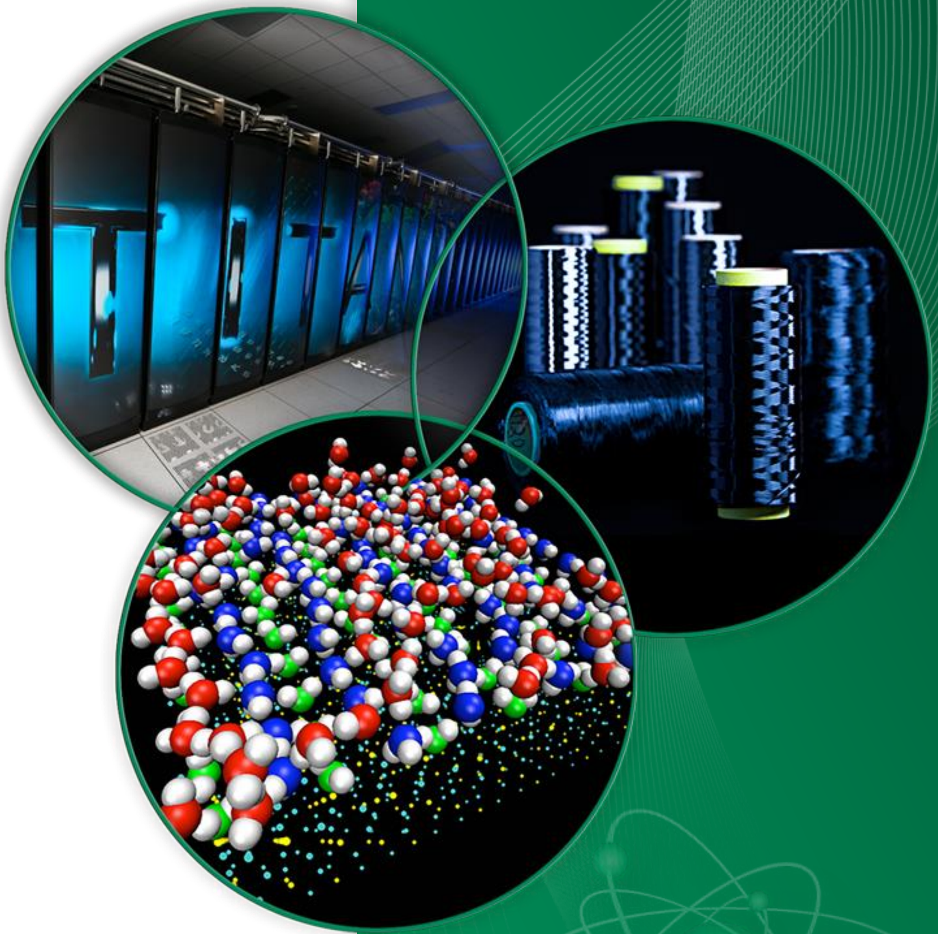
HIPATIA BoF

OpenSHMEM 2014 Workshop
March 5, 2014
Annapolis, MD



Joshua Lothian, Jonathan Schrock, Matthew Baker, Ed D’Azevedo, Stephen Poole

HIPATIA

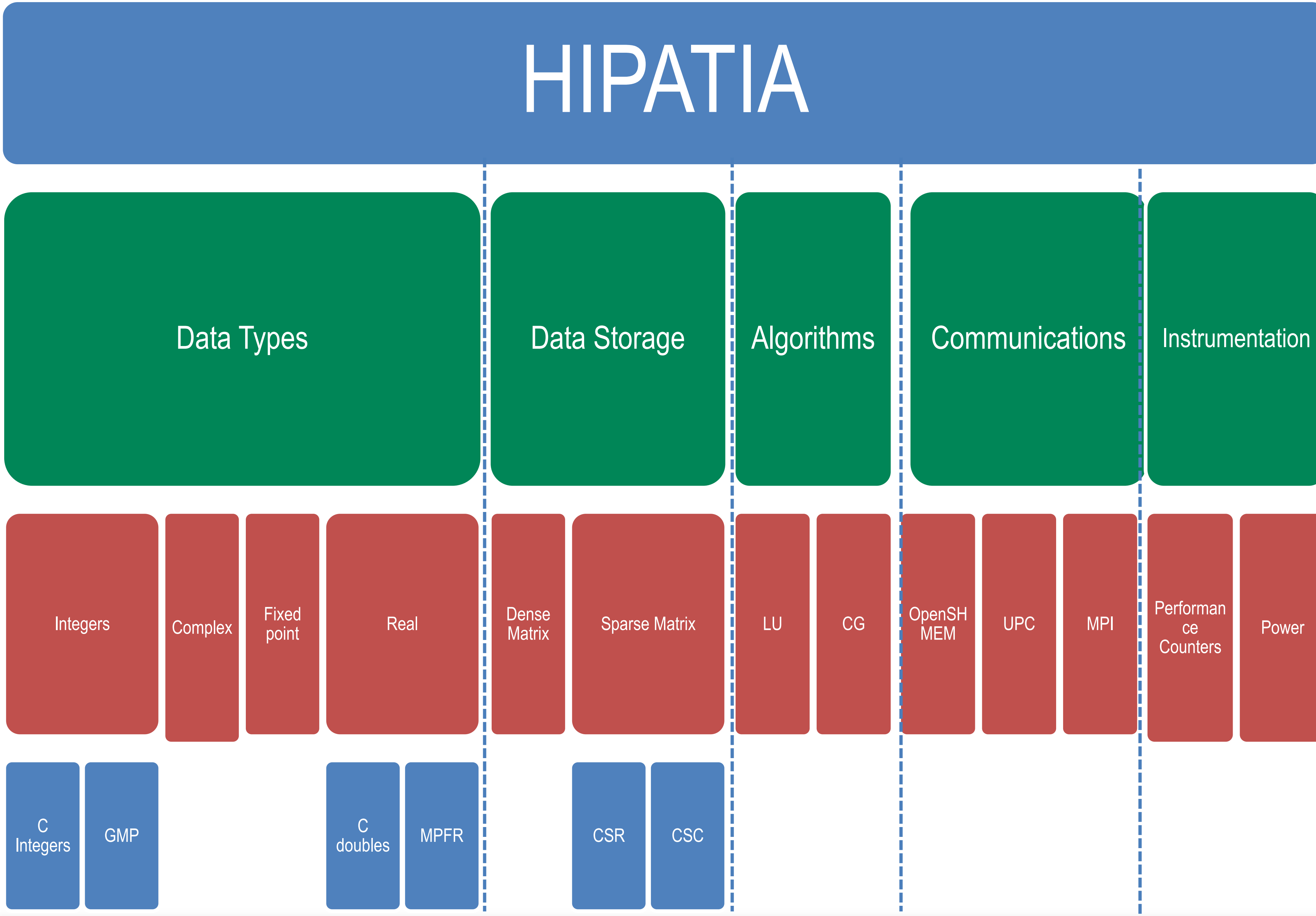
- HIPATIA is:
  - The **HI**gh **P**erformance **A**dap**T**ive **I**ntegrated Linear **A**lgebra Benchmark
  - Pluggable benchmark initially targeting OpenSHMEM communications across numerous data types, data storage, algorithms
  - Focus on areas not well represented by HPL
  - Open source (license TBD)
  - Portable, autoconf-based build system

Why HIPATIA?

- Not all applications are solving real-valued, dense matrices
  - HPCG benchmark addresses sparse matrices, but only in reals
- Desire for a good OpenSHMEM-focused benchmark
- Allow for different inputs – maybe we don't want a random matrix, but some sort of graph as input (think "Big Data")
- Industry likes having a single number to show to management that represents the problems of interest to them
  - Oil & Gas interest in fixed point

HIPATIA features

- We intend to provide a straightforward, pluggable architecture for linear algebra benchmarking
- Ability to measure power/energy will be a core capability
- Likewise, where possible, we will utilize hardware counters for performance instrumentation
  - Many processors lack counters for integer ops ☹


HIPATIA
Data Types
Data Storage
Algorithms
Communications
Instrumentation
Integers
Complex
Fixed point
Real
Dense Matrix
Sparse Matrix
LU
CG
OpenSHMEM
UPC
MPI
Performance Counters
Power
C Integers
GMP
C doubles
MPFR
CSR
CSC

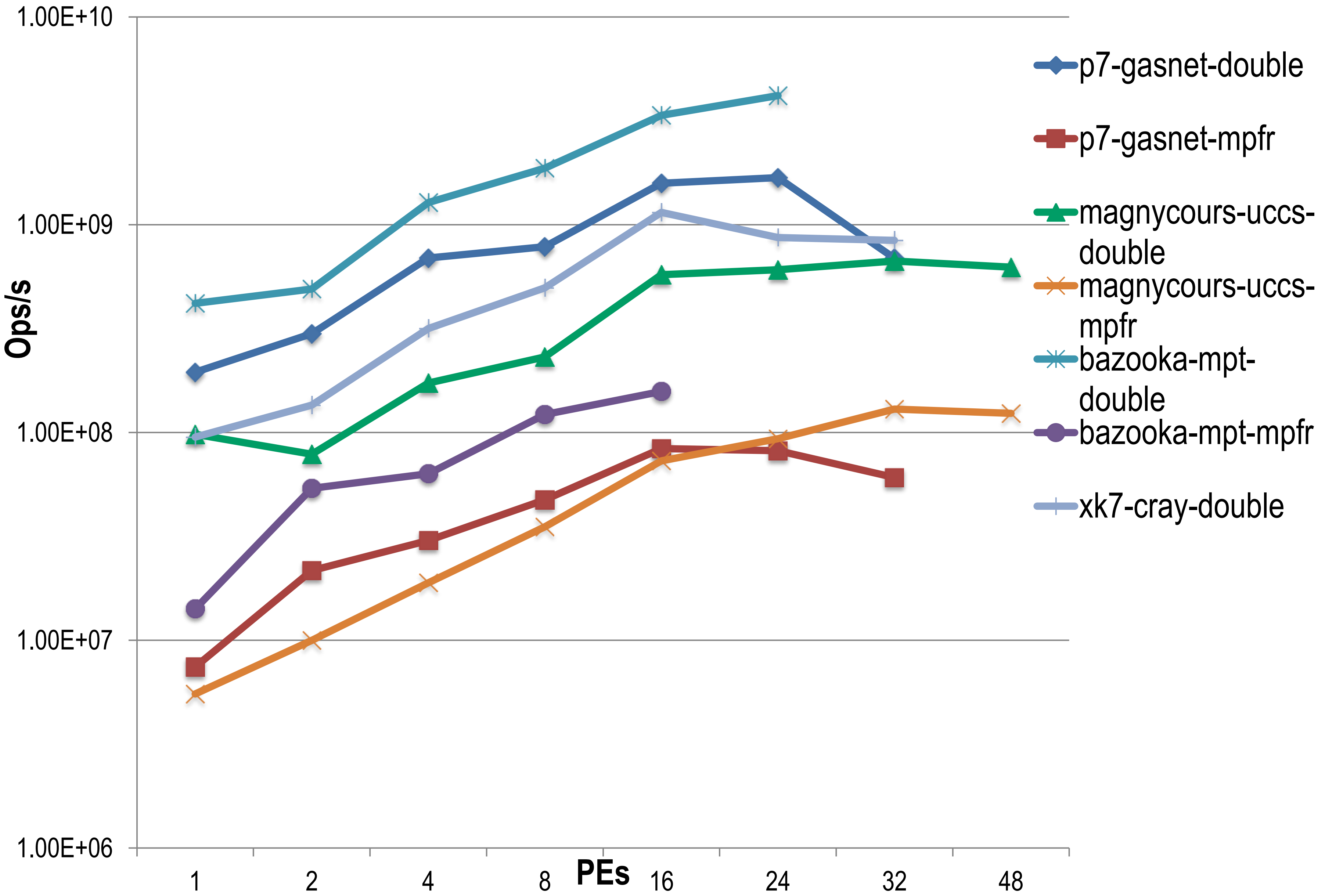
Current development status

- We initially targeted dense integer and real matrices
  - Dense, real is well studied, and relatively easy to implement
  - Dense integer is somewhat less well studied, but still more straightforward than sparse integer
- Kernels implemented but still need to be integrated
- Integer calculations performed using GNU MPFR library - http://www.mpfr.org/
  - 64-bit and 128-bit floating point **however:**
  - MPFR uses GMP in backend, which executes on integer units on rather than FP
- Multiple “true” integer algorithms have been prototyped in serial

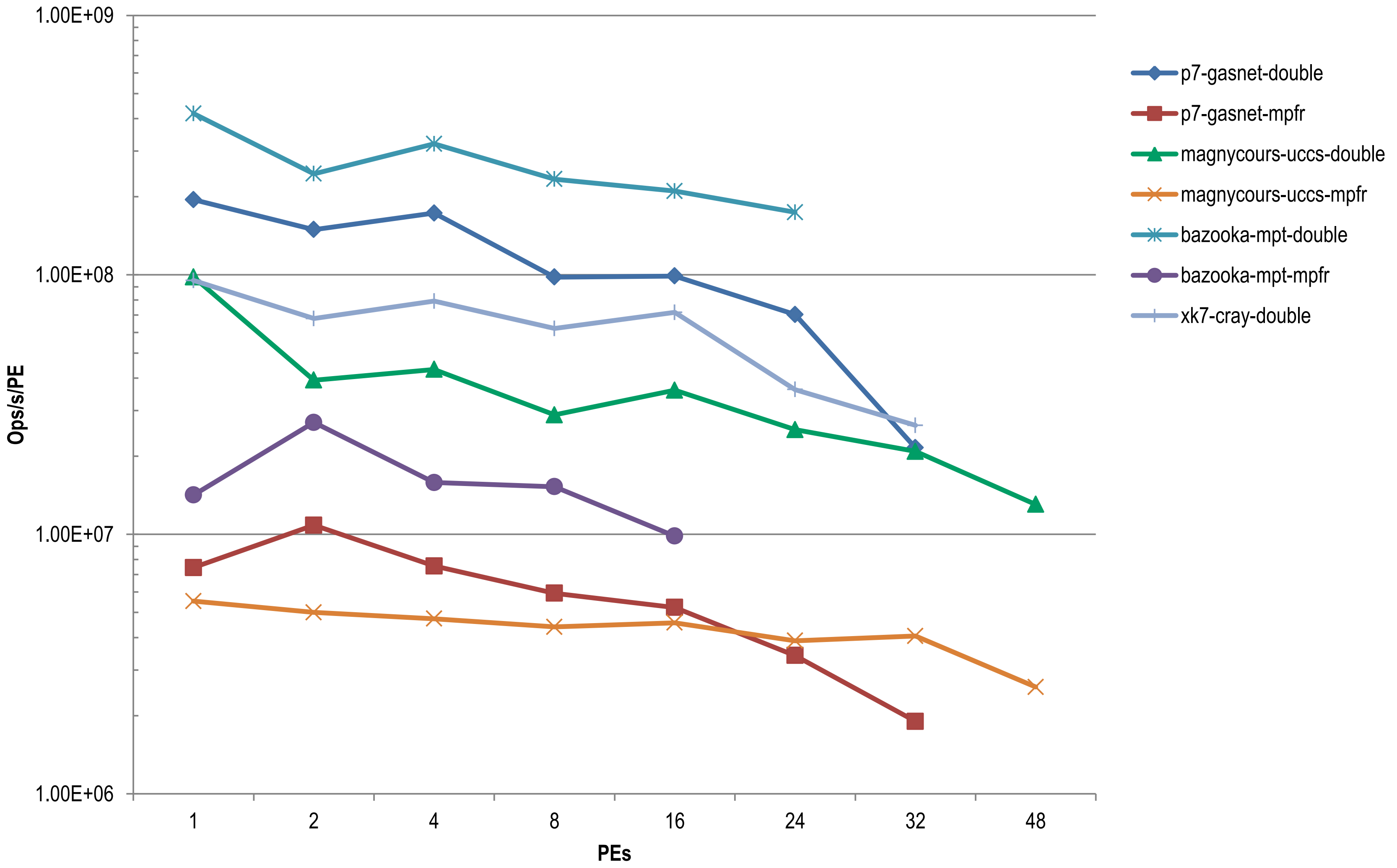
Some notes on preliminary results

- These are **very** preliminary numbers for a GEMM kernel
- Math and communication work, but have not been tuned for speed
- Upcoming additions to OpenSHMEM API will help (active sets, etc.)
- In short, we are showing ease of portability rather than meaningful performance results

Preliminary Results



Performance per PE



What's next?

- Complete user interface
- Standardize plugin API
  - Allow vendors/implementers to swap in optimized versions of solver/comms/etc
- Determine exactly how MPFR operations map to CPU operations
- Solutions of sparse matrices in the integer domain
  - Challenging to find appropriate algorithms that:
    - keep intermediate values within the integers
    - Keep intermediate values a reasonable size
    - Have consistent/deterministic storage and runtime complexity
- Open source release

Integer solvers

- We can take two different paths:
  - Native C integers
  - GMP integer
- C ints are great, easy to transfer with OpenSHMEM
  - Values are limited
- GMP ints are great, near limitless values
  - Painful to transfer with OpenSHMEM as size can vary depending on value
  - Either export to a string, or poke around in GMP internals

Integer Solvers

- A variation of the Hermite Normal Form algorithm by Kannan and Bachem.
  - Suffers from intermediate term explosion, cannot use C types
    - Single intermediate terms O(1-10MB) for highly constrained inputs
- Dixon's p-adic expansion algorithm
  - Most of the calculations performed modulo a prime, can use C types for these
- Congruence methods utilizing the Chinese Remainder Theorem
  - Most of the calculations performed modulo a prime, can use C types for these
  - Solution calculated for several different primes
- Finite Fields

Acknowledgements

This work was supported by the United States Department of Defense & used resources of the Extreme Scale Systems Center at Oak Ridge National Laboratory.