Opening Remarks for MCHPC'17: Workshop on Memory Centric Programming for HPC

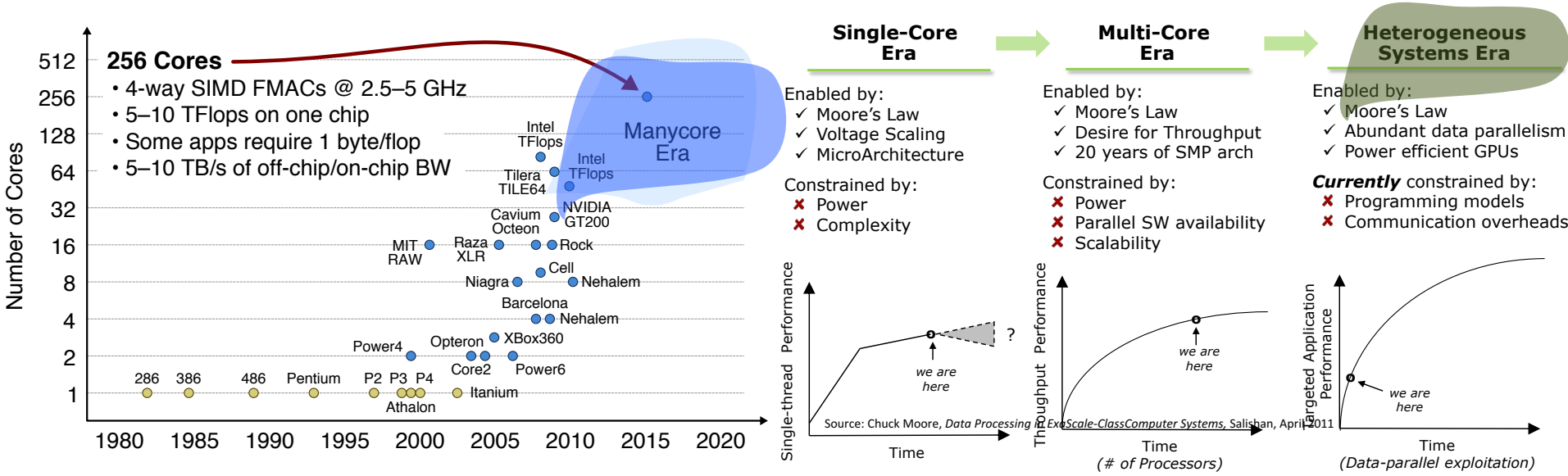
in conjunction with SC17 and in cooperation with ACM SIGHPC

9:00AM - 12:30PM, November 12, 2017

Yonghong Yan, University of South Carolina
Ronald B. Brightwell Sandia National Laboratories
Xian-He Sun, Illinois Institute of Technology

http://passlab.github.io/mchpc/mchpc2017.html

Today's Parallel Architecture



- **Massive increase in parallelism**
  - **Multicore (4 - 32) → Manycores (100s – x1000s)**
- **Heterogeneity and domain-specific processors**
  - **System-level (accelerators) vs chip level (embedded)**
- **Power and memory wall challenges**

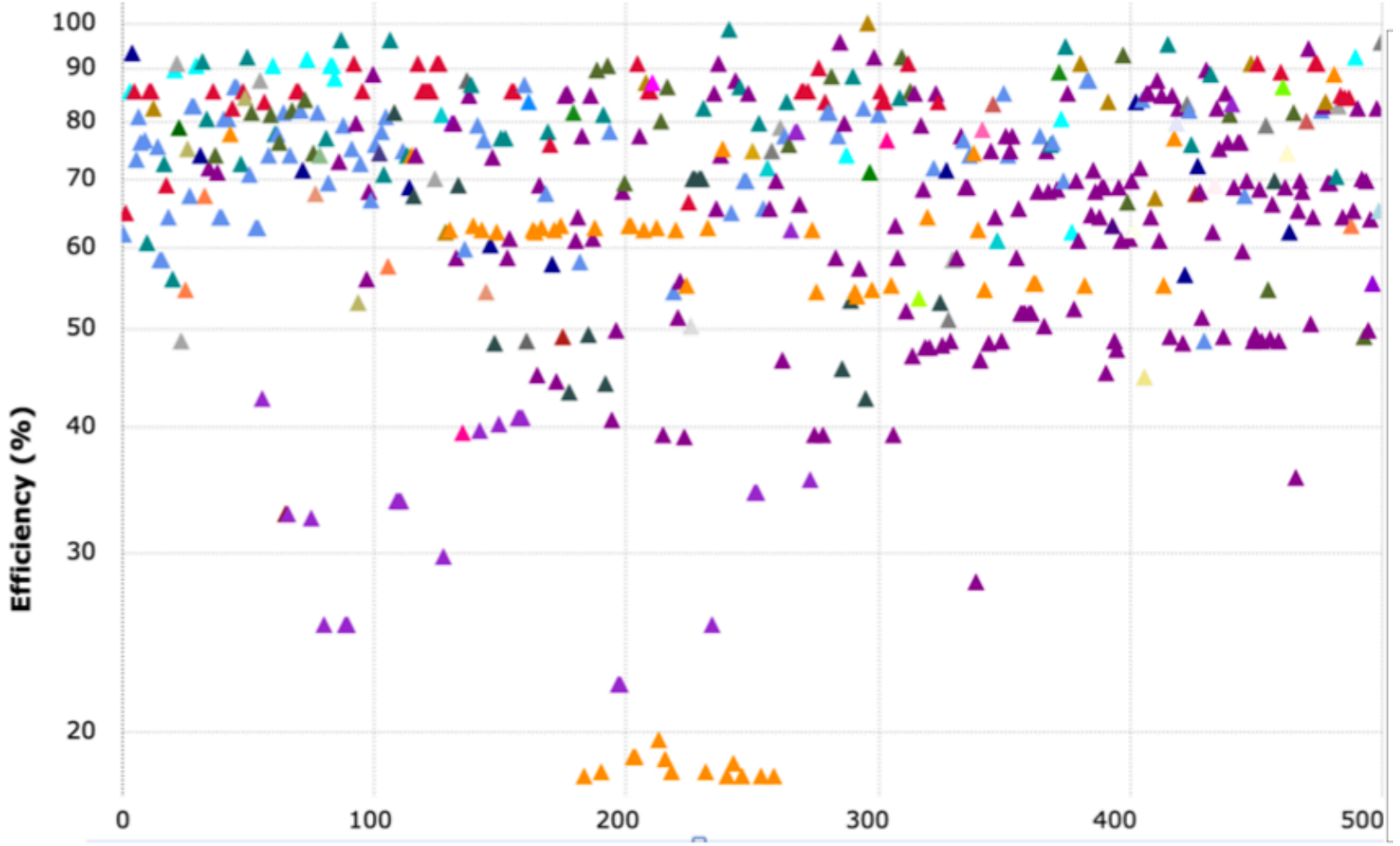
Increasing Memory Complexity

- **The memory wall challenge is becoming much tougher**

Memory Types	Read Latency (ns)	Write Latency (ns)	Bandwidth (GB/s)
SRAM and Cache (L1, L2 and L3)	2 - 8	2 - 8	
DRAM	50 - 200	50 - 200	25
Stacked DRAM (HMC and HBM)	40 - 90	40 - 90	400
NAND NVRAM	~100us	~2-3ms	1GB/s for read and 10MB/s for write
3D XPoint (Intel and Micron)	2-3x slower than DRAM	4-6x slower than DRAM	

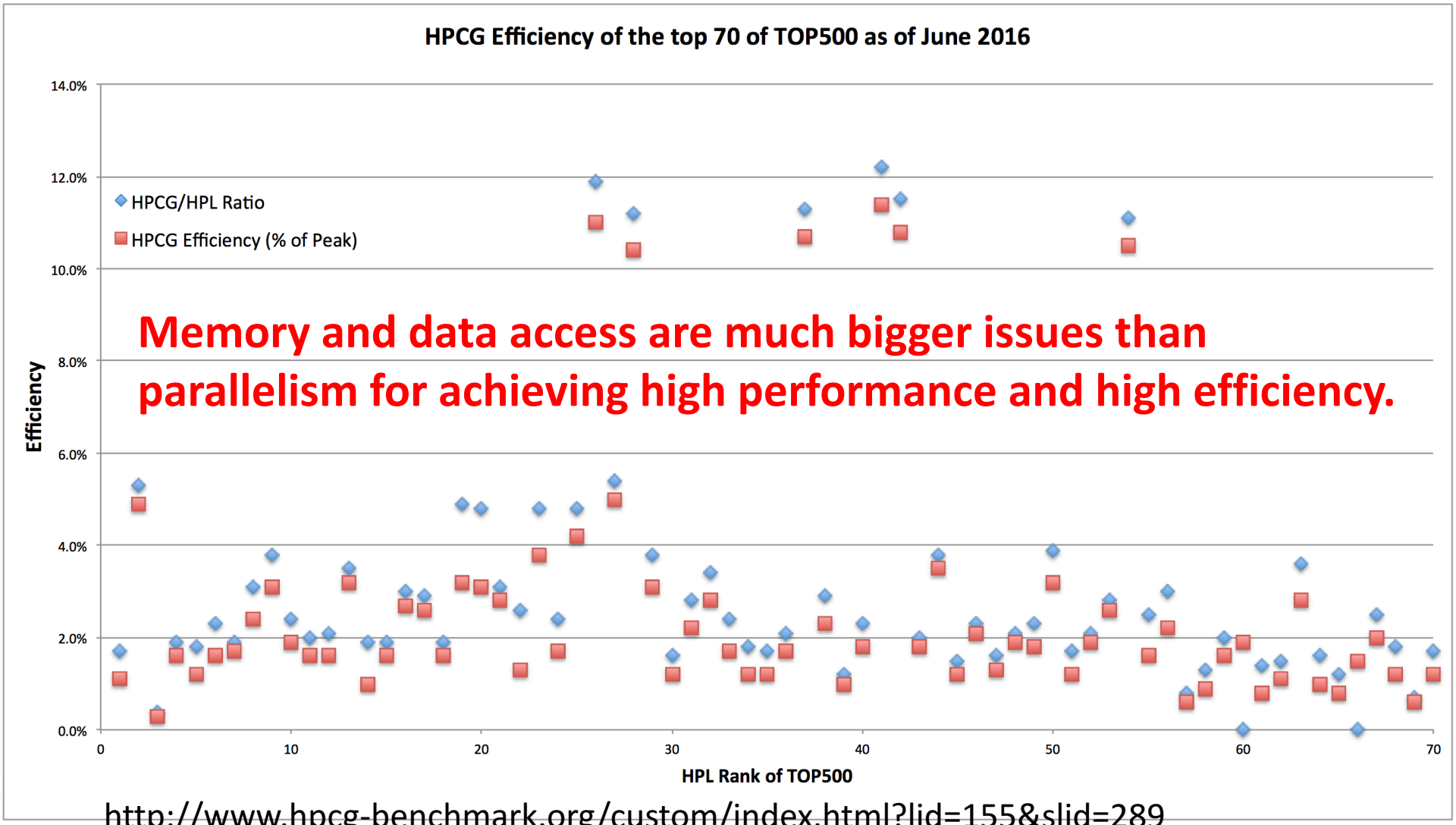
Reality: Performance Efficiency of HPL (Computation Intensive)

- Ratio of app performance to peak performance



Efficiency of High Performance Conjugate Gradients (HPCG) (Comp/Data ratio are more balanced)

- Ratio of sustained performance to peak



http://www.hpcg-benchmark.org/custom/index.html?lid=155&slid=289

Memory Centric Programming for HPC

The notion and techniques of exposing the hardware memory system and its hierarchy to the programmer via portable programming abstractions and APIs

- **DRAM and NUMA regions, shared and private caches, scratch pad, 3-D stacked memory, and non-volatile memory**

- **Examples:**
  - ***place* in OpenMP and X10, and *locale* in Chapel to represent memory regions in a system;**
  - ***shared* modifier in CUDA and *cache* modifier in OpenACC for representing GPU scratch pad SRAM;**
  - ***memkind* library for specifying memory type ;**
  - **Recent OpenMP memory management constructs;**
  - **PMEM library for persistent memory programming**

Goals of MCHPC Workshop

- **Elevate the notion of memory centric programming to utilize the unprecedented and ever-elevating modern memory systems**

- **Discuss state-of-the art and challenges of programming models, compiler and runtime systems for working with existing and emerging memory systems**

MCHPC'17 Program

- 09:00 - 09:10 Opening Remarks, Yonghong Yan, University of South Carolina
- 09:10 - 10:00 Session 1: Keynote Talk: Compiler and Runtime Challenges for Memory Centric Programming, Vivek Sarkar (Georgia Tech)
- Session Chair: Ron Brightwell, Sandia National Laboratories
- 10:00 - 10:30 Break
- 10:30 - 11:10 Session 2: Invited Talk: Persistent Memory: The Value to HPC and the Challenges, Andy Rudoff (Intel)
- Session Chair: Yonghong Yan
- 11:10 - 12:28 Session 3: Paper Presentations, Session Chair: Yonghong Yan
- 11:10 - 11:35 Bit Contiguous Memory Allocation for Processing In Memory, John Leidel
- 11:35 - 12:00 Beyond 16GB: Out-of-Core Stencil Computations, Istvan Reguly, Gihan Mudalige and Mike Giles
- 12:00 - 12:15 NUMA Distance for Heterogeneous Memory, Sean Williams, Latchesar Ionkov and Michael Lang
- 12:15 - 12:28 Evaluating GPGPU Memory Performance Through the C-AMAT Model, Ning Zhang, Chuntao Jiang, Xian-He Sun, and Shuaiwen (Leon) Song
- 12:28 - 12:30 Closing

Appreciation and Acknowledgement

- All authors who submitted papers to this workshop
- Program committee members for providing us with high-quality reviews under tight deadlines.
- Thankful to our Keynote speaker, Vivek Sarkar from Georgia Tech and our Invited speaker Andy Rudoff from Intel Corporation.
- Special thanks to SIGHPC for publishing the proceedings of the workshop.
- SC17 Workshop chair: Almadena Chtchelkanova and Luiz DeRose (SC ’17 Workshop Chairs)
- Sincere thanks are due to the attendees, without whom this conference would not be a success

Sponsorship

- **The workshop is part of a CAREER project sponsored by NSF under contract # 1652732**