Exploring HPC and Big Data Convergence: a Graph Processing Study on Intel KNL

Alexandru Uta, Ana Varbanescu, Ahmed Musaafir, Chris Lemaire, Alexandru Iosup

a.uta@vu.nl

Vrije Universiteit Amsterdam

HPC and Big Data Infrastructure

How does the hardware and software landscape look for these paradigms?

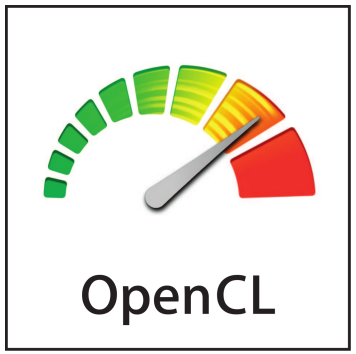
HPC Infrastructure

- Large numbers of (thinner, low-power) cores
- Intricate NUMA topologies
- Fast interconnects (InfiniBand, 40+ Gb Ethernet)
- Accelerators (GPUs, FPGAs, TPUs)
- Compute-intensive workloads (simulations)

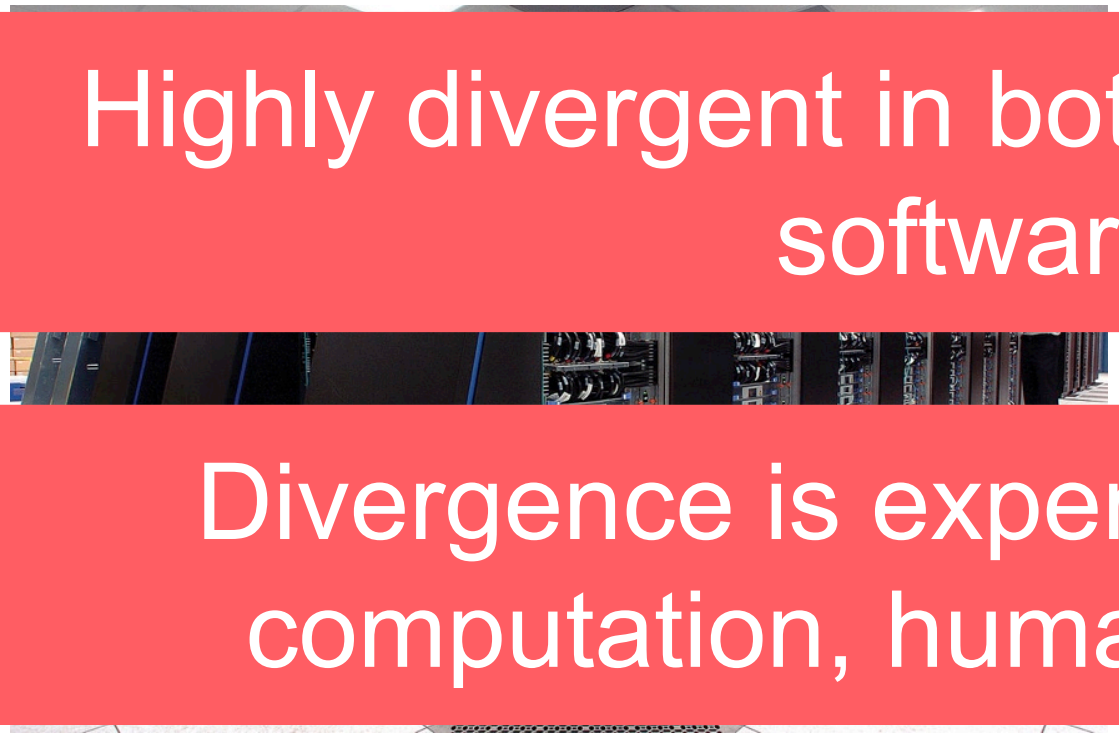
Big Data Infrastructure

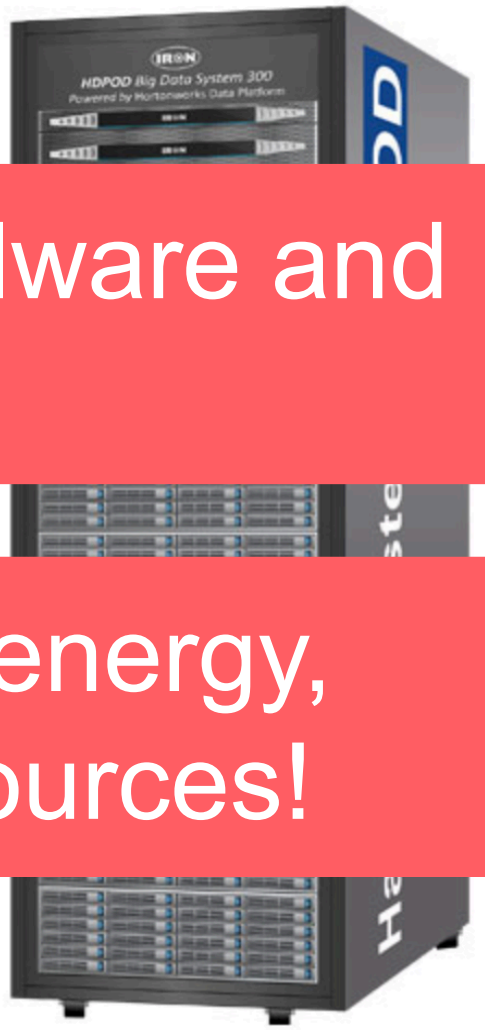
- (generally) commodity hardware
- Fat-core CPUs
- large memory (and caches) per core
- Large storage
- Less emphasis on fast networks
- Often virtualized clusters (cloud)
- Data-intensive workloads

HPC vs. Big Data Software



HPC and Big Data Infrastructure





Highly divergent in both hardware and software!

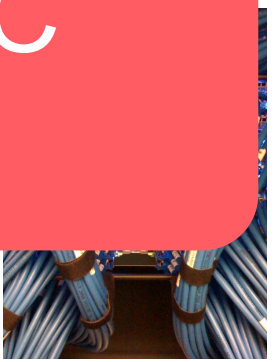
Divergence is expensive: energy, computation, human resources!

HPC and Big Data Convergence

- Only in software: porting big data to HPC hardware

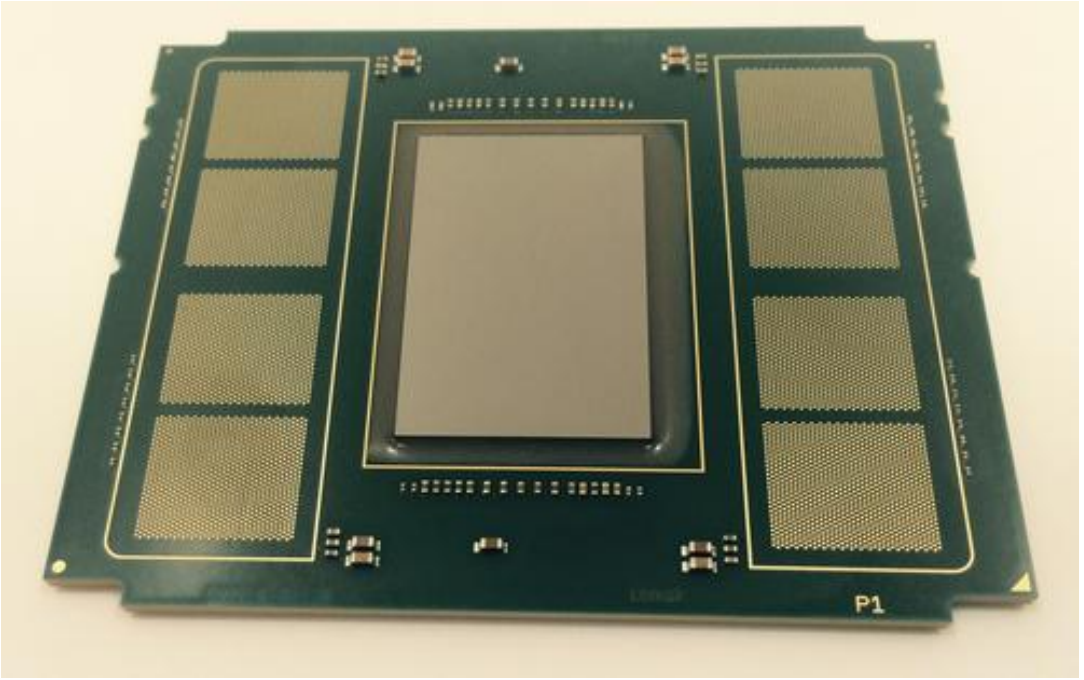
Significant effort in porting and tuning!

Can we run big data directly on HPC hardware?



Big Data on Intel Knights Landing

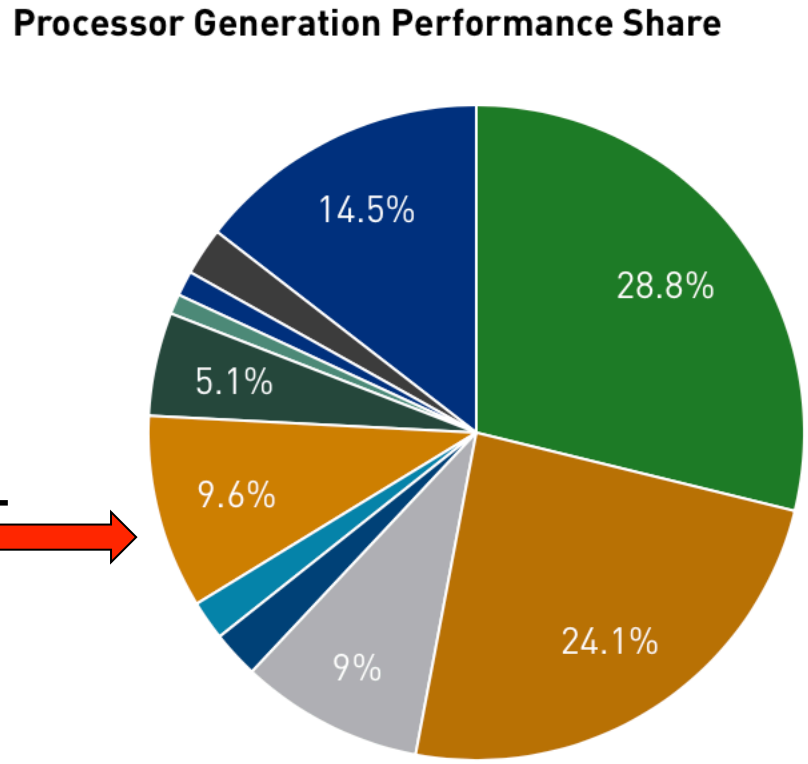
- Intel KNL – 2nd generation Xeon Phi
- Accelerator-like self-booting CPU
- Full x86_64 compatibility
- (up to) 72 low-power Intel Atom cores
- Wide vector instructions (512B)
- 16GB high-bandwidth on-chip memory
- **3 TFLOPS + 400 GB/s (on-chip) memory bandwidth**



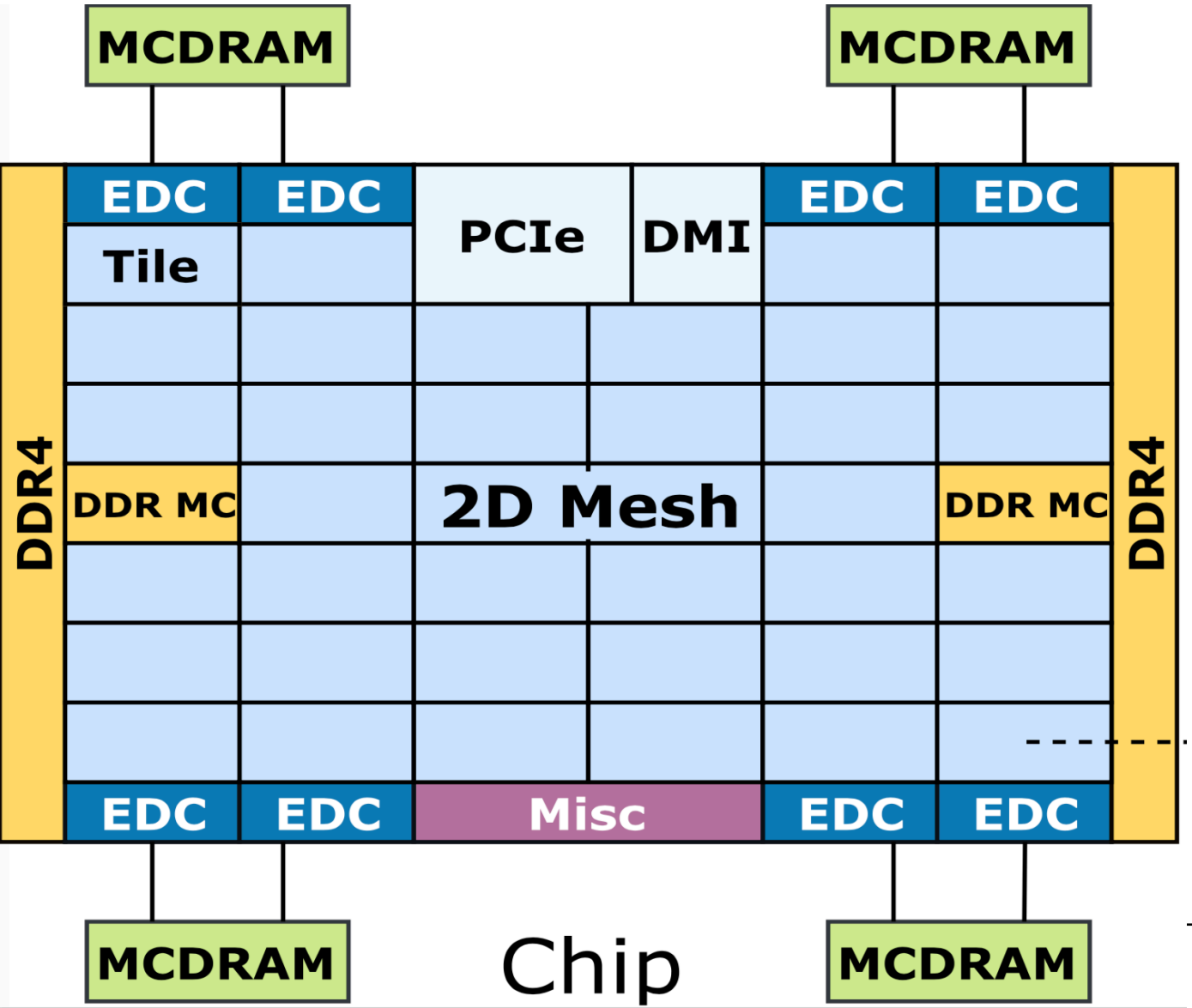
Intel KNL – Highly Representative for HPC

- 3 clusters in top 10 of top500.org contain KNL
- ~3% of the share of CPUs in top500
- ~10% of the performance share of top500
- Highly configurable at boot time
- Works as many different machines (due to configurable clustering and memory modes)

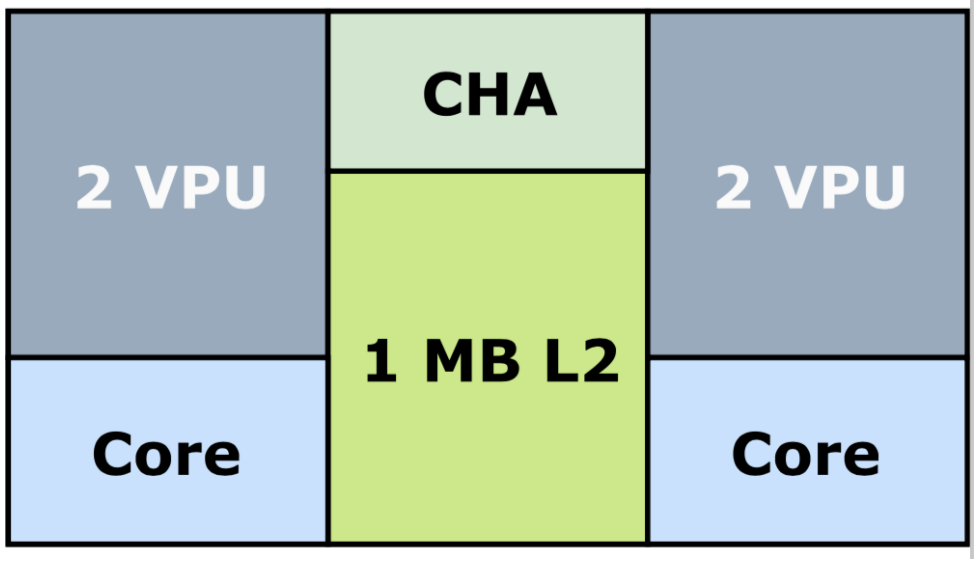
KNL



KNL Architecture



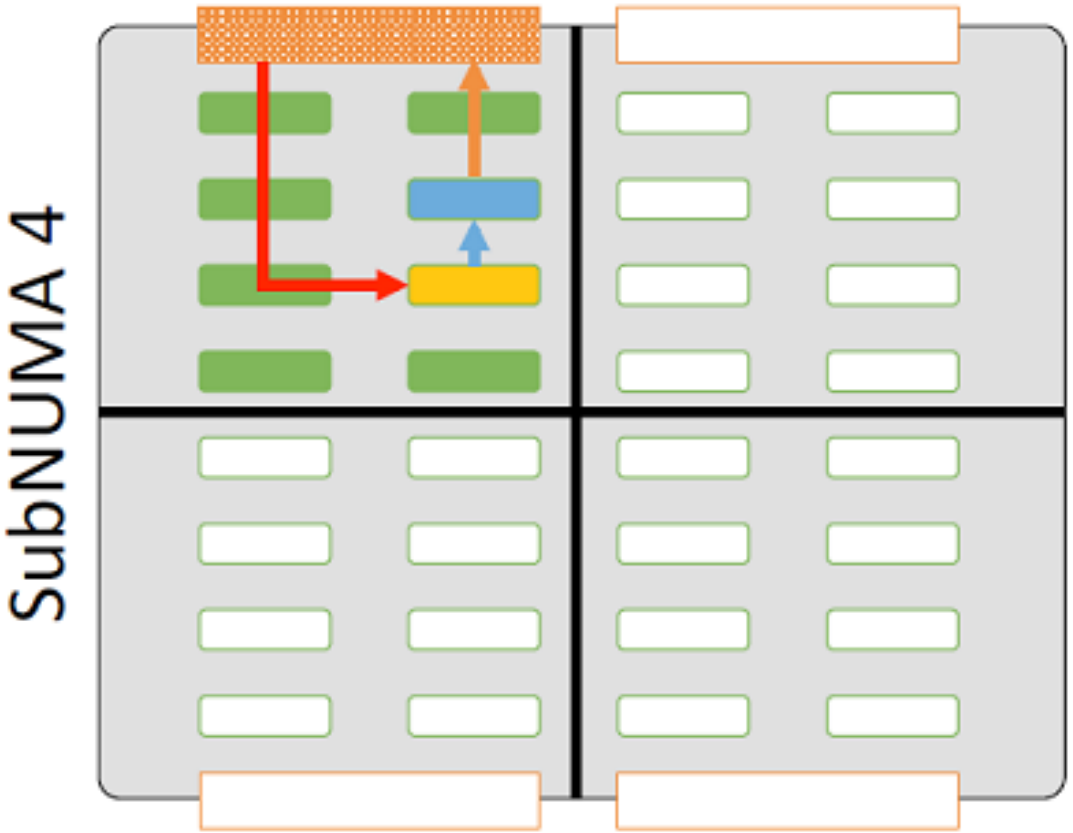
Chip

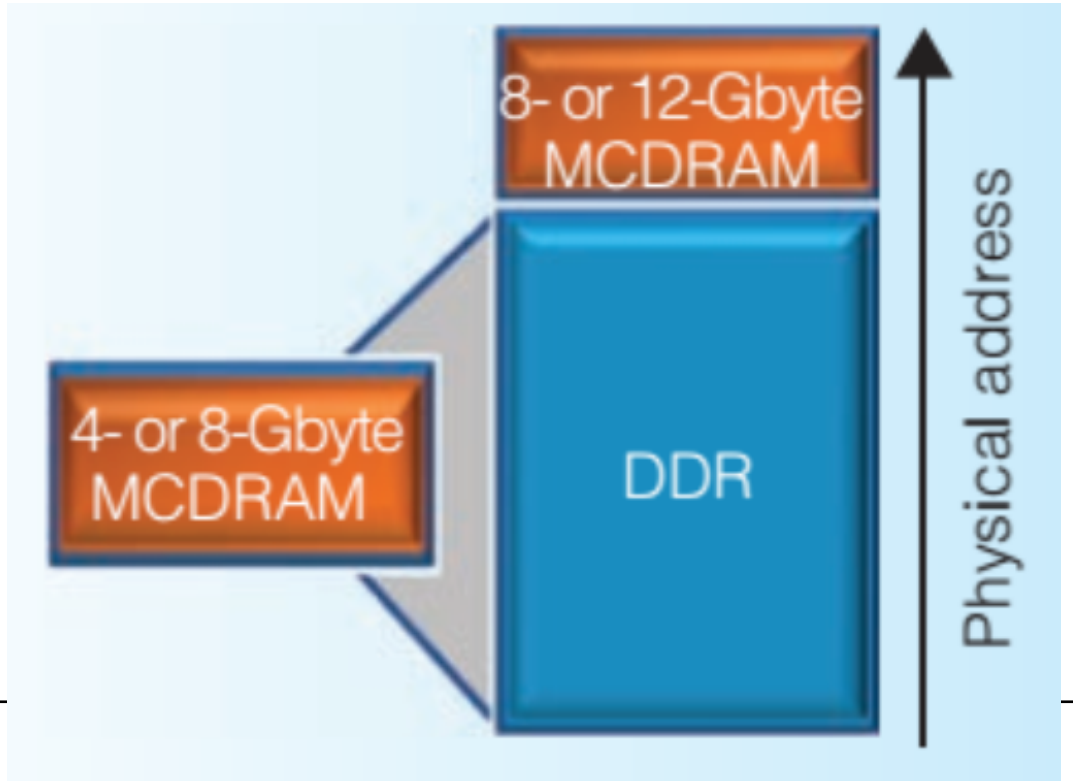


Tile

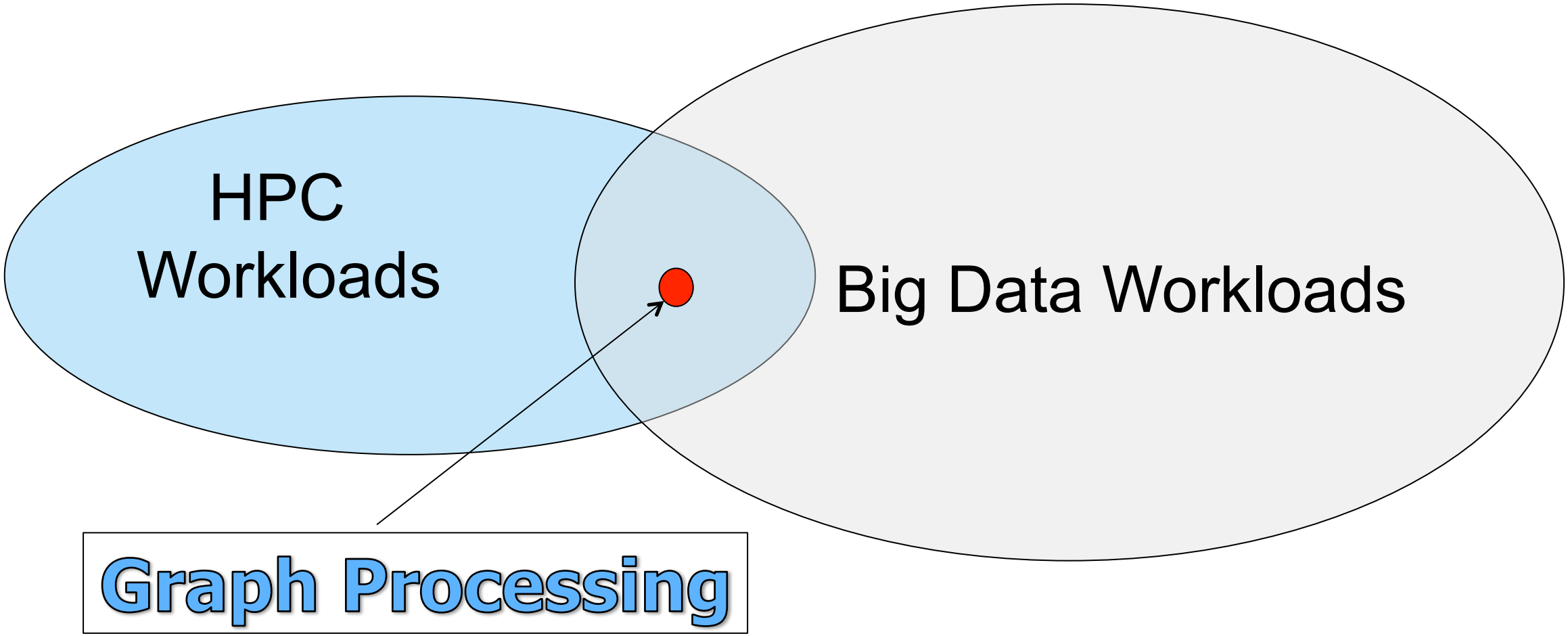
KNL – Hardware Parameter Space

- Clustering modes: (L2 cache miss latency)
  - All2All
  - Quadrant/Hemisphere
  - NUMA
- Memory modes: (on-chip memory)
  - Flat
  - Cache
  - hybrid



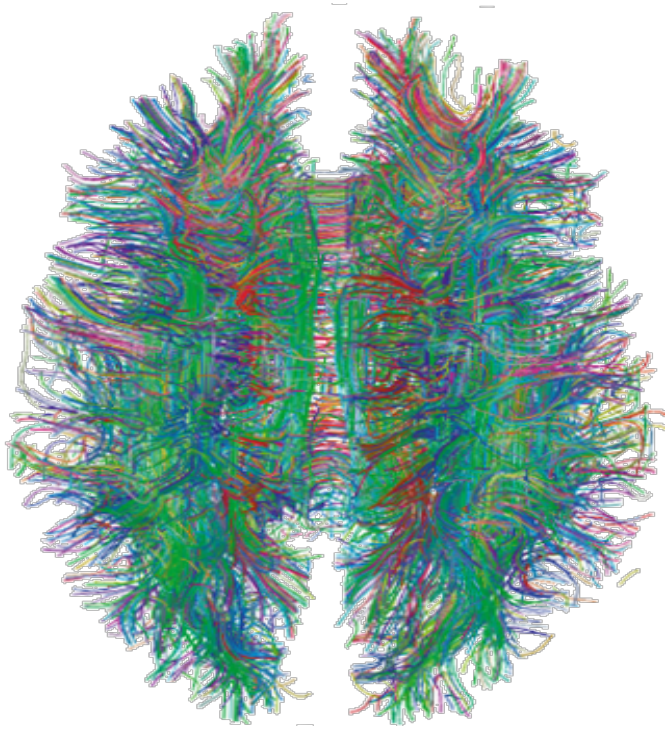


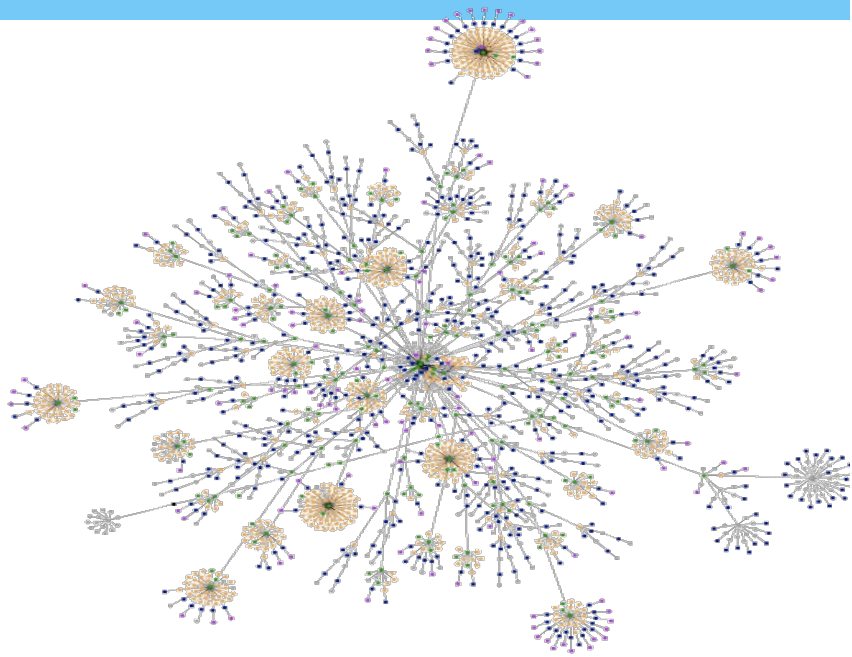
Graph Processing – HPC and Big Data



Graph Processing – High-impact Domain

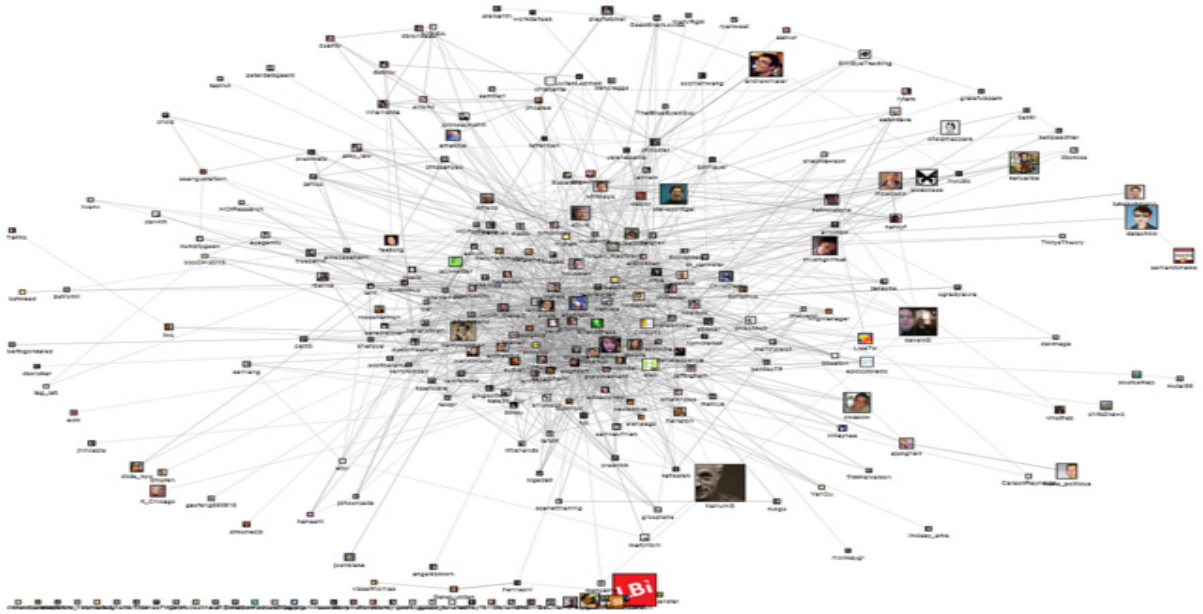
- Social networks
- Drug discovery
- Monitoring wildfires
- Combating human-trafficking
- Studying the human brain





Graph Processing – Highly Challenging

- Mostly traversing links between entities
- Little computation
- Mostly memory bound
- Highly irregular workloads
- Cache misses
- PAD Triangle [1,2]

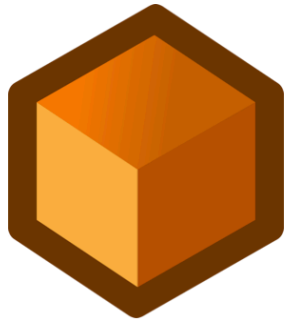


Performance = f(platform, algorithm, dataset)

[1] Guo et al., IPDPS ’14 ; [2] Iosup et al., VLDB ‘16

How to study the convergence?

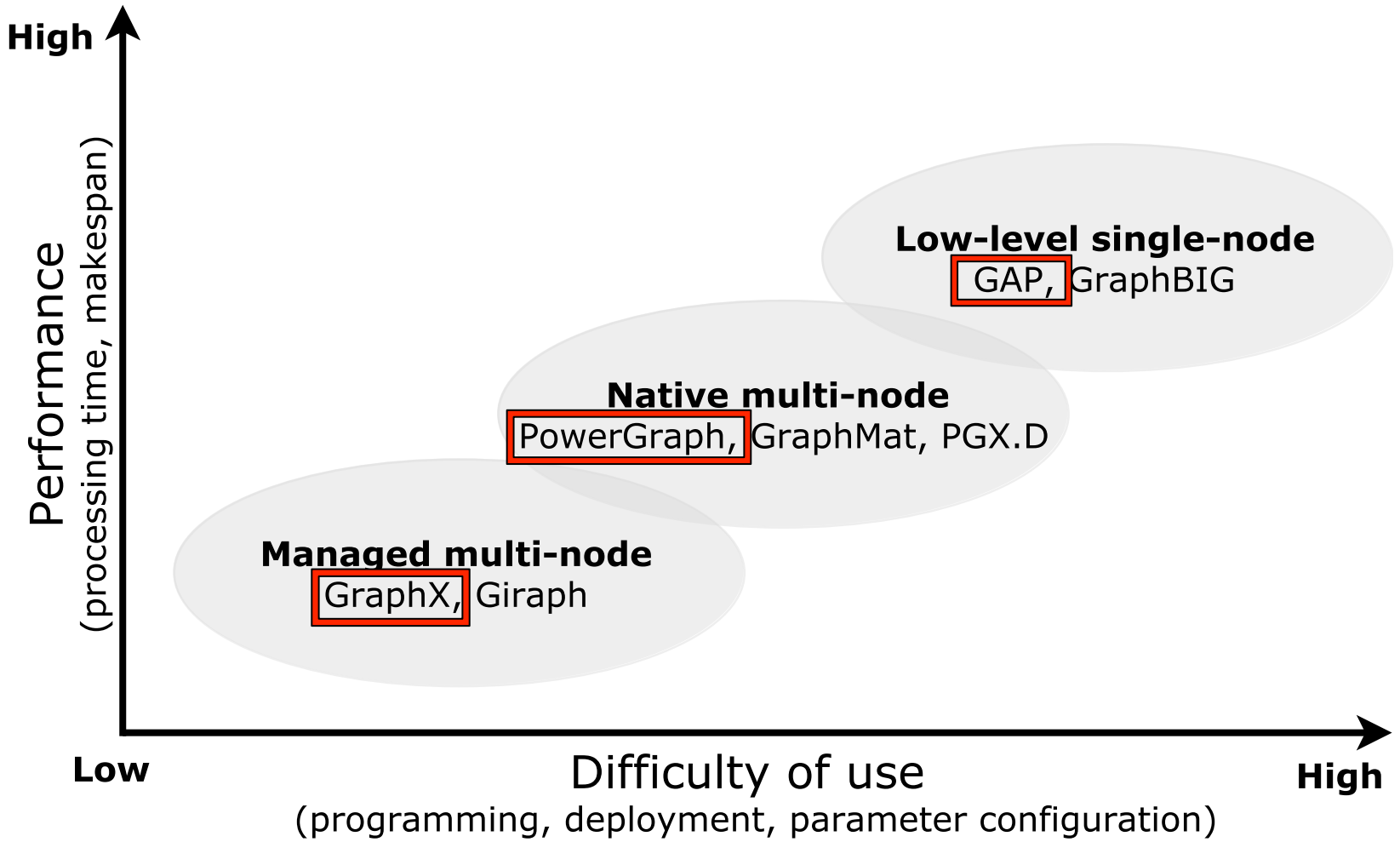
- Benchmark using Graphalytics
- Multiple classes of algorithms
- Multiple datasets (scale-free and non-scale free)
- Multiple classes of graph analytics platforms
- Comparison between KNL and de-facto big data hardware (Intel Xeon family)



Graphalytics

Open-source Graph Processing Benchmark Suite

Graph Analytics Platforms



Quantifying the Convergence

- Large-scale study – over 300,000 compute core-hours
- Experiments run in DAS-5, Cartesius cluster*, Intel Academic cluster*

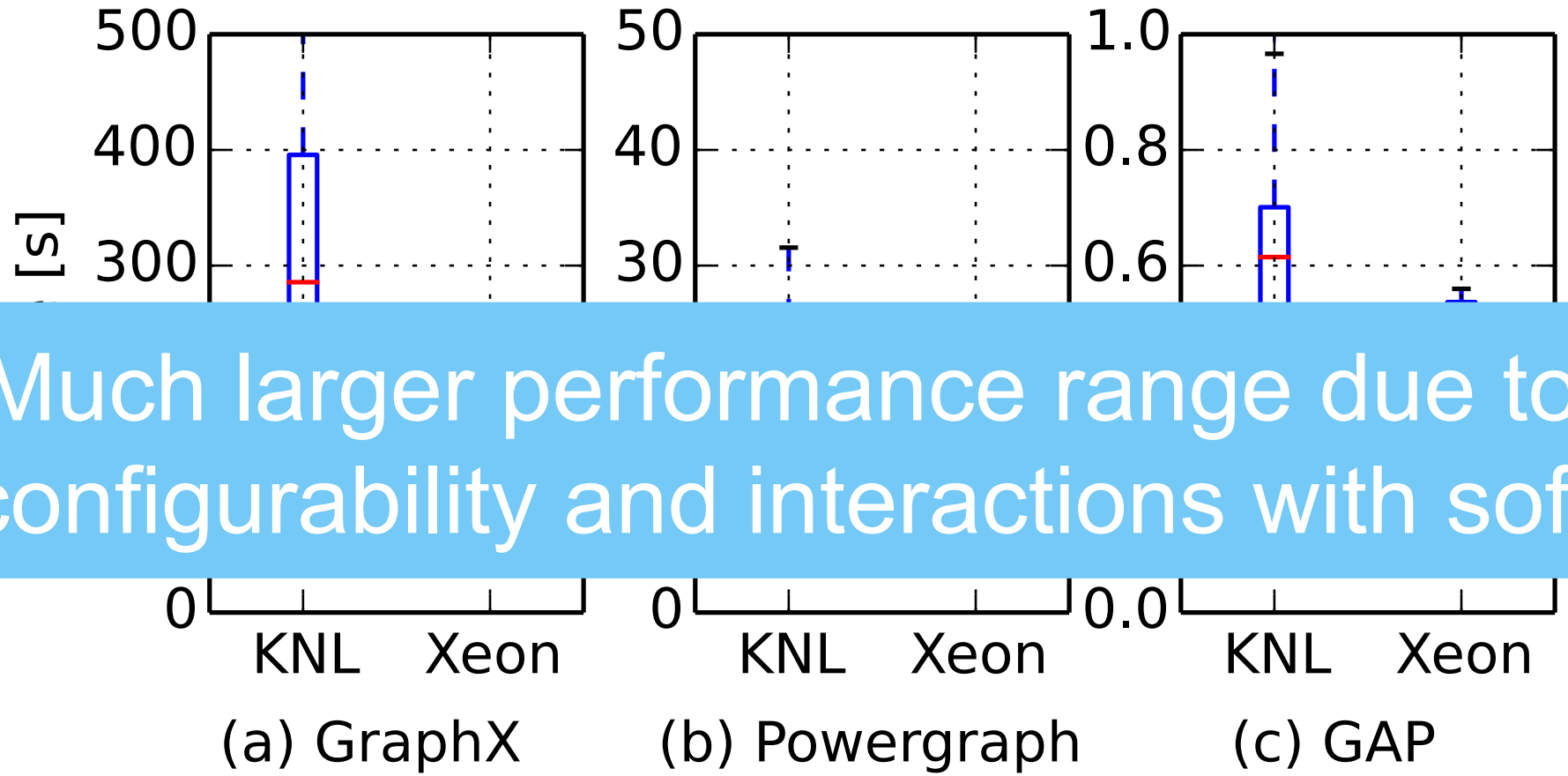
	Xeon E5-2630v3	**Xeon Phi 7230**
Cores	16 (32 hyperthreads)	64 (256 hyperthreads)
Frequency (GHz)	2.4	1.3
Network	56Gbit FDR InfiniBand	56Gbit FDR InfiniBand
Memory	64GB DDR4	96GB DDR4
OS	Linux 3.10.0	Linux 3.10.0

* Thanks to grants from NWO and Intel

What to assess?

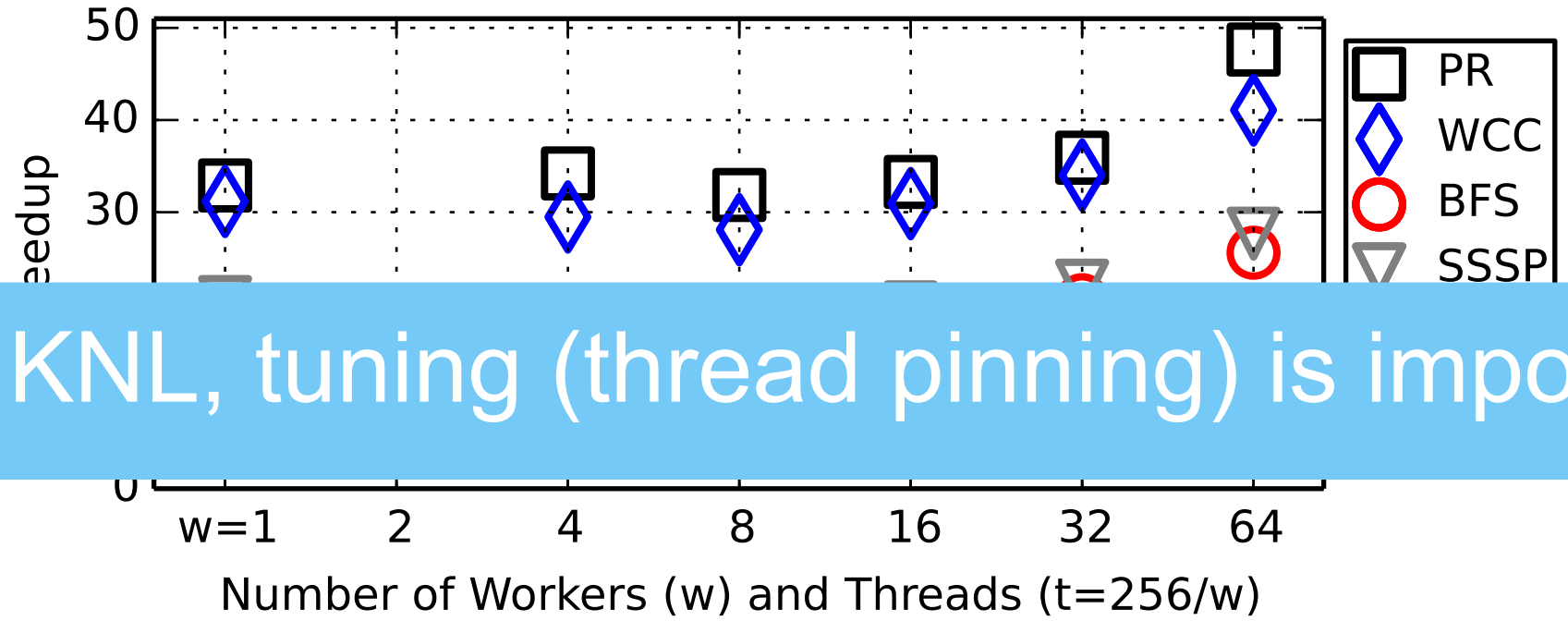
- How does the KNL parameter space influence performance?
- How (difficult it is) to tune the platforms on KNL?
- Is KNL faster than Xeon?
- Does it scale?

Hardware + PAD



Much larger performance range due to KNL configurability and interactions with software!

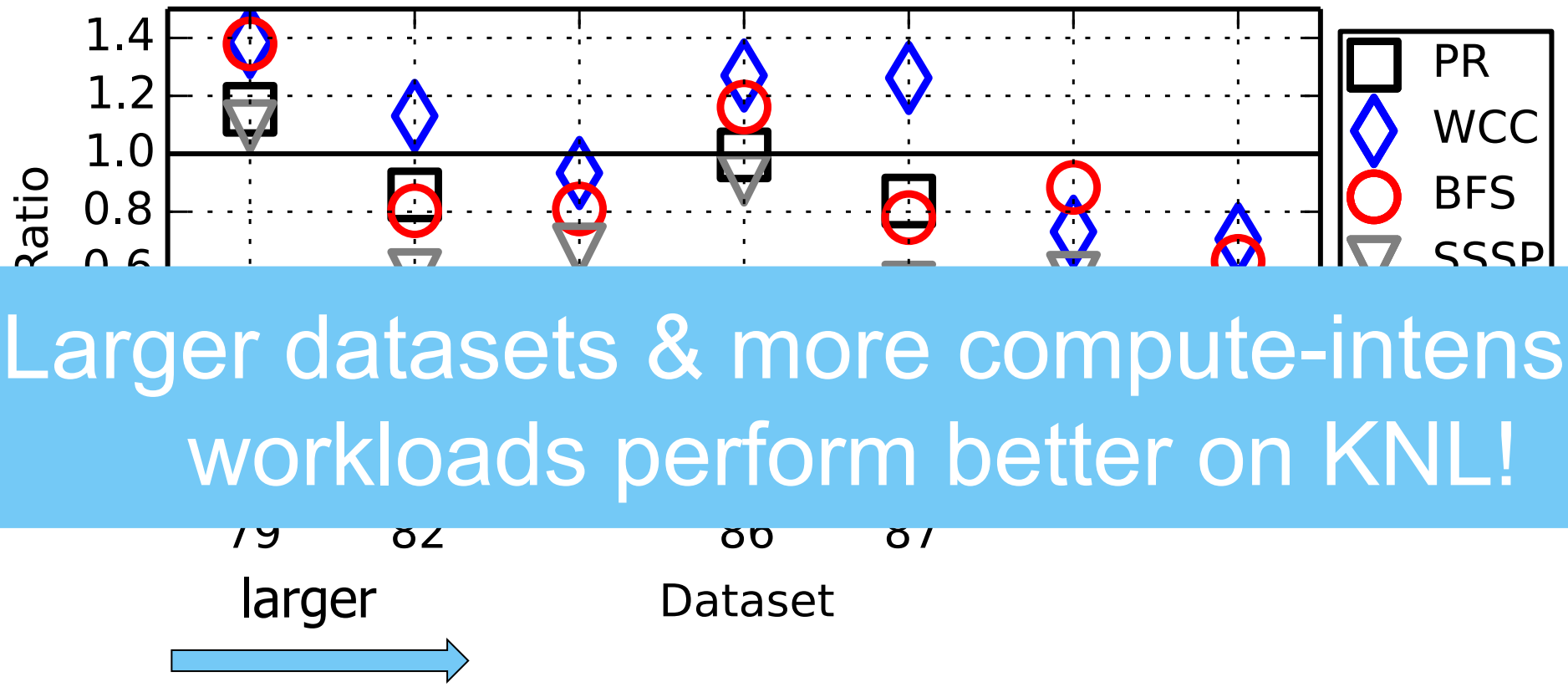
KNL Hardware + Platform Interaction and Tuning



On KNL, tuning (thread pinning) is important!

Powergraph, Datagen_7-9 – thread pinning speedup
(pinning on Xeon – 5% improvement)

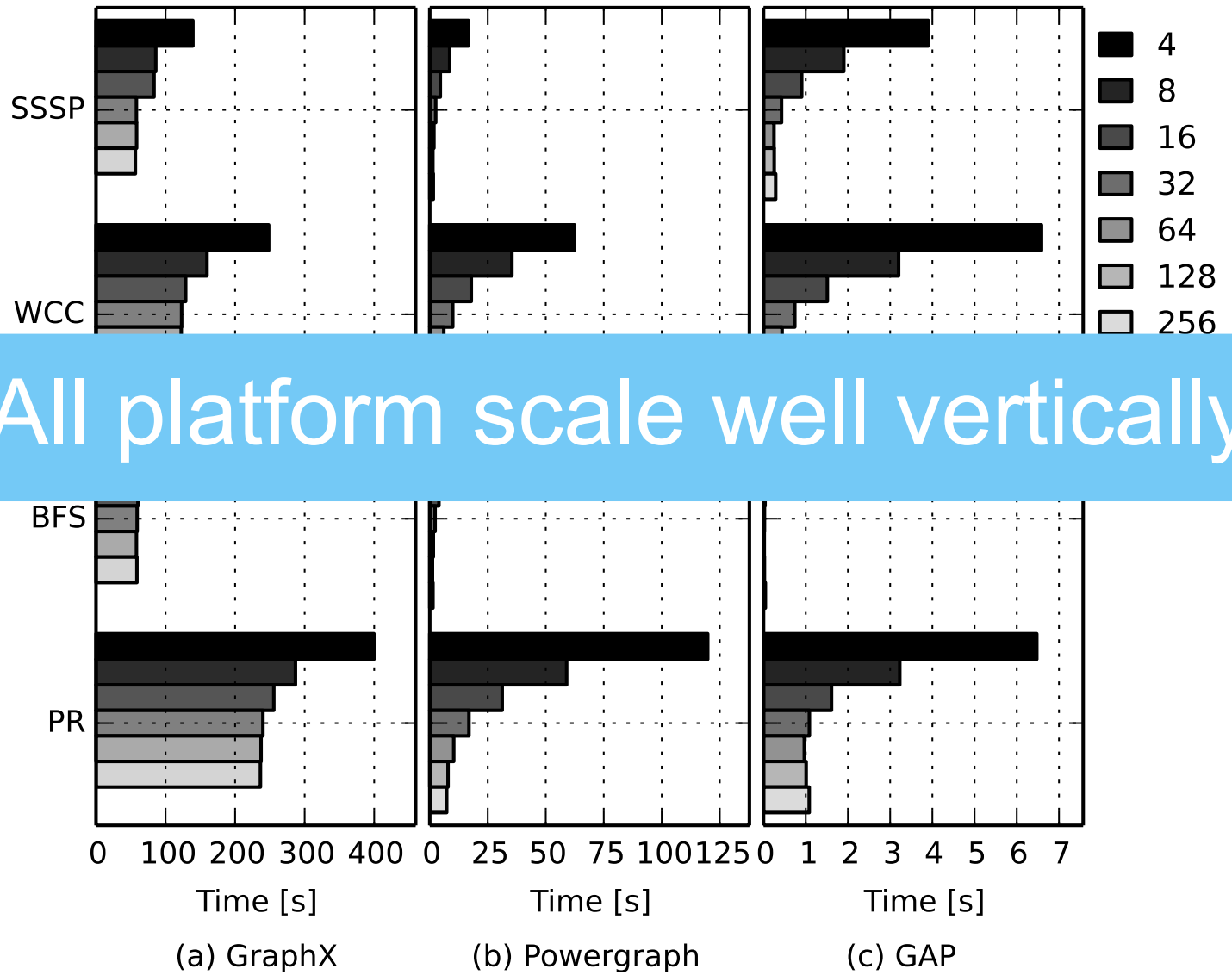
KNL outperforms Xeon



Larger datasets & more compute-intensive workloads perform better on KNL!

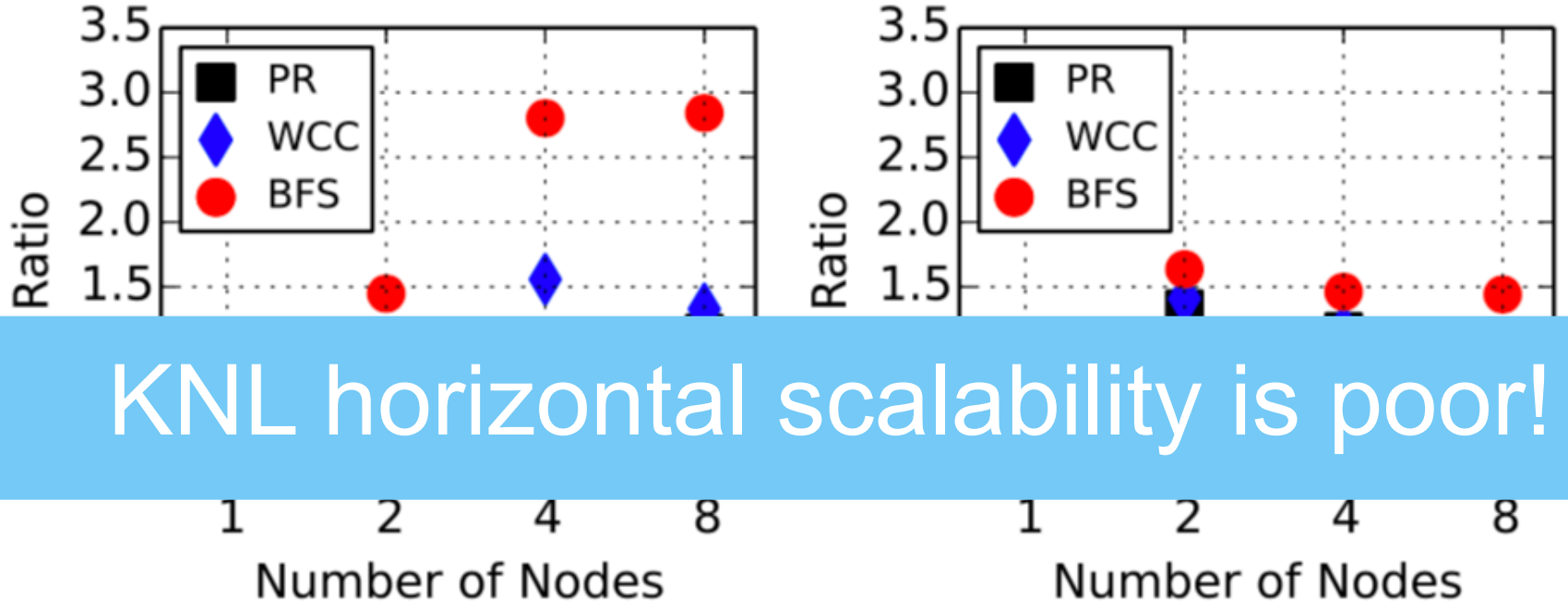
GAP, KNL vs. Xeon Speedup

KNL Vertical Scaling



All platform scale well vertically!

Horizontal Scaling



(a) GraphX. (b) Powergraph.

KNL horizontal scalability is poor!

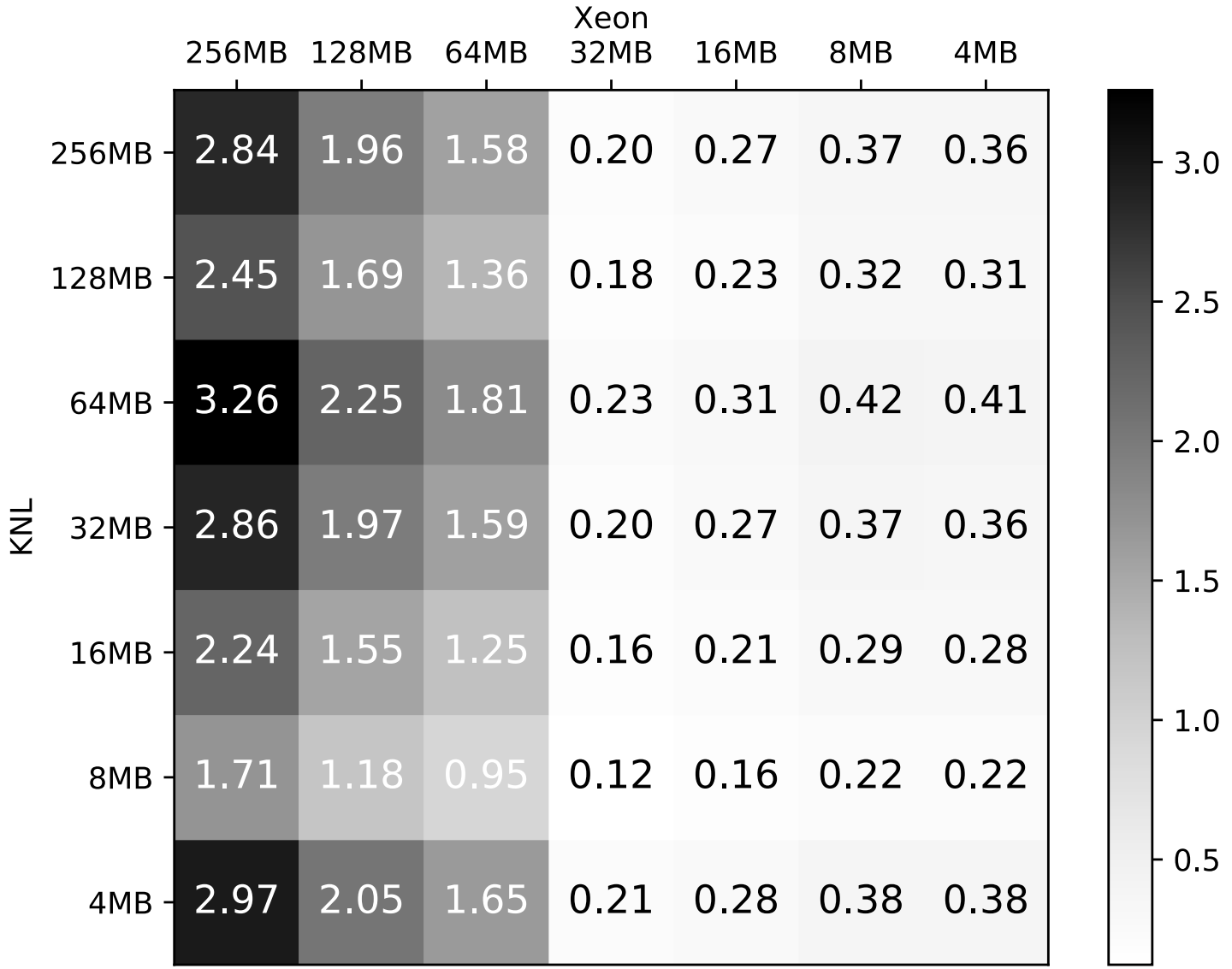
KNL vs. Xeon Speedup on 1-8 nodes
KNL single-thread networking is slow!

Take-home Message: Main Findings

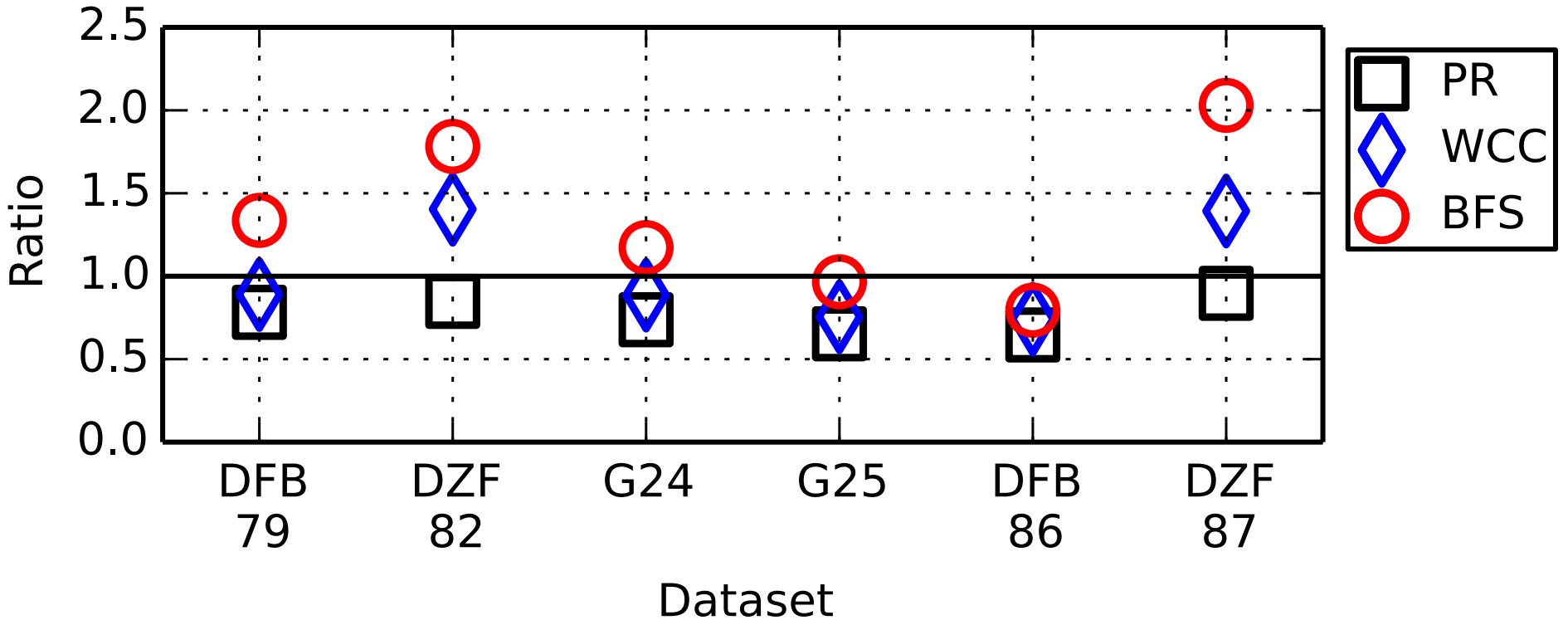
- MF1: **Convergence** – KNL outperforms Xeon
- MF2: **HPAD** – hardware adds an extra complexity layer
- MF3: **H-P interaction** – platforms closer to the hardware perform better on KNL
- MF4: **Tuning** – good performance entails significant tuning for KNL
- MF5: **Scaling** – KNL scales well vertically, but cannot scale horizontally
- Future work: adapt software to KNL
  - Use wide vectors
  - Use the on-chip memory
  - Multithreaded I/O and networking

Extra Slides

Tuning GraphX



KNL vs. Xeon on Powergraph



KNL – Modes Analysis