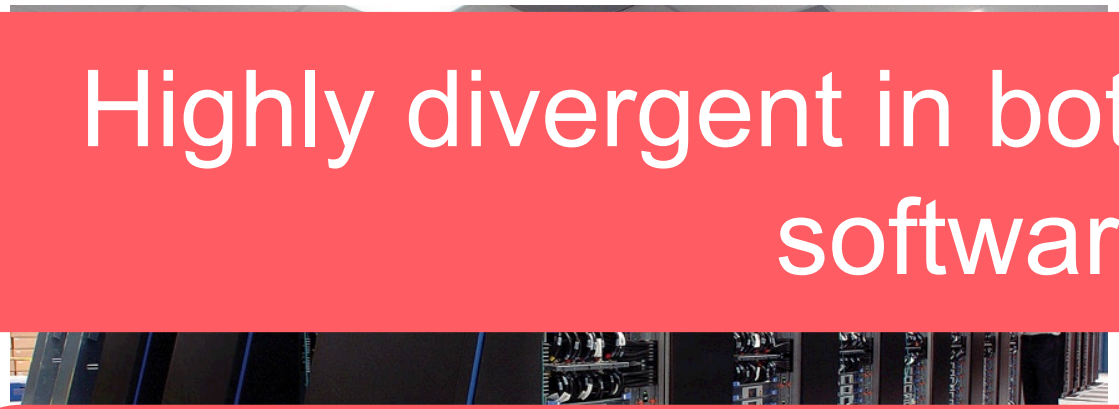
Exploring HPC and Big Data Convergence: a Graph Processing Study on Intel KNL

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

a.uta@vu.nl

Vrije Universiteit Amsterdam

HPC and Big Data Infrastructure



Highly divergent in both hardware and software!

Divergence is expensive and unsustainable: energy, computation, human resources!

Divergence - unsustainable and expensive!

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

Most big data stacks are unable to take advantage of (HPC) hardware features.

Addressing the 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? What are the trade-offs?

Big Data on HPC-capable Many-cores

Representative:

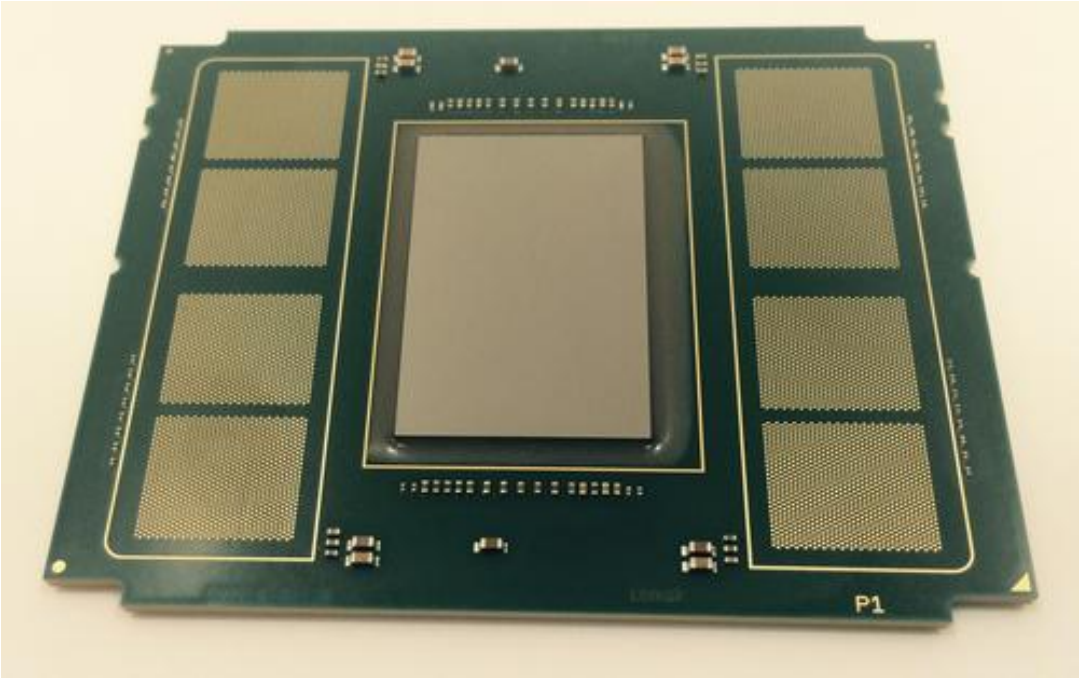
- Intel KNL – 2nd generation Xeon Phi

Can run Big Data:

- Accelerator-like self-booting CPU
- Full x86_64 compatibility

HPC Features:

- (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

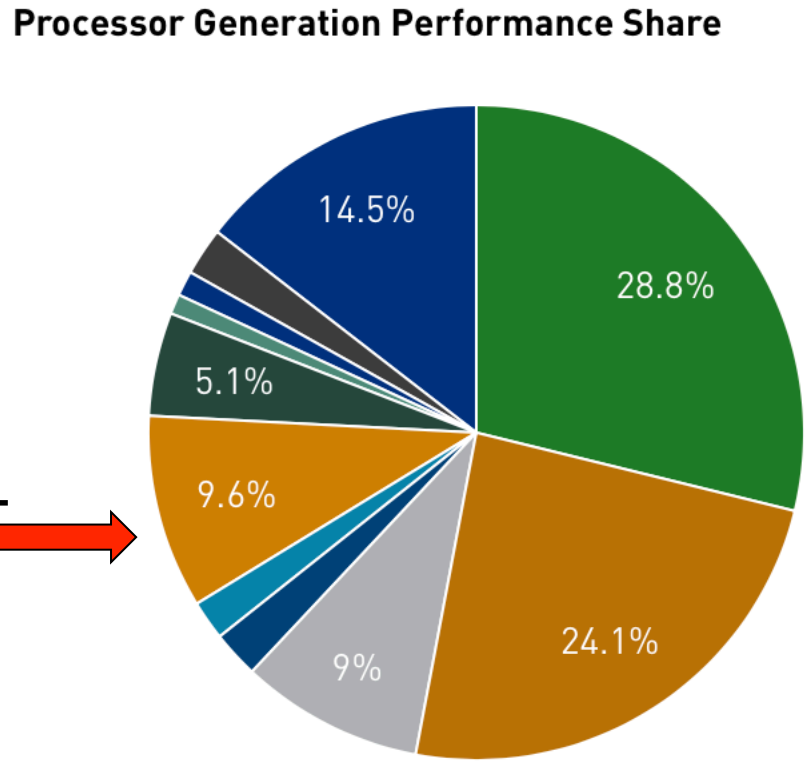
Representative for Top500:

- 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

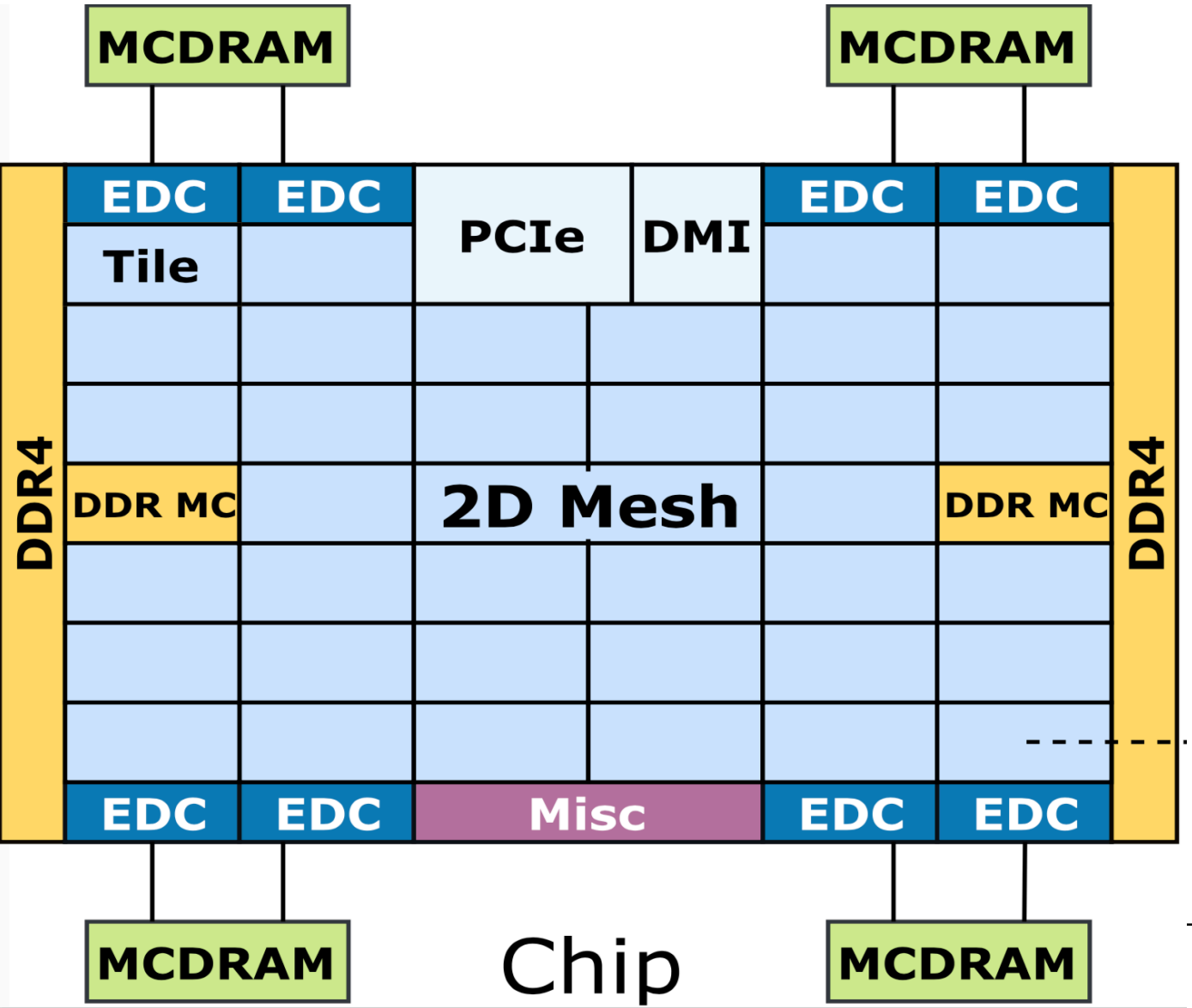
Many performance facets:

- Highly configurable at boot time
- Works as many different machines (due to configurable clustering and memory modes)

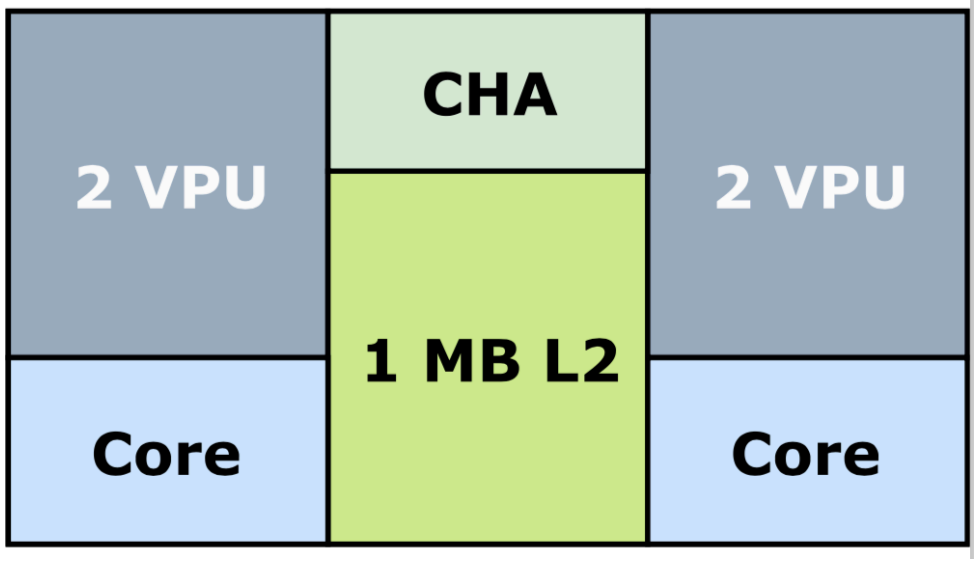
KNL



KNL Architecture



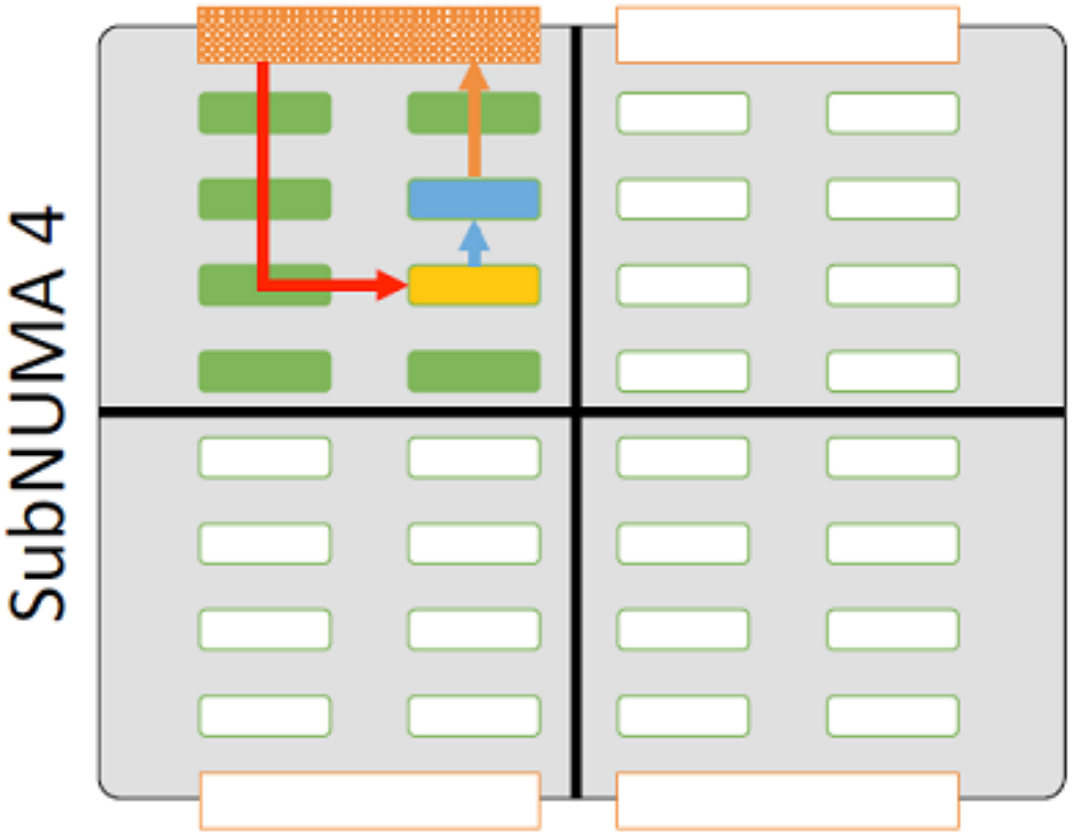
Chip

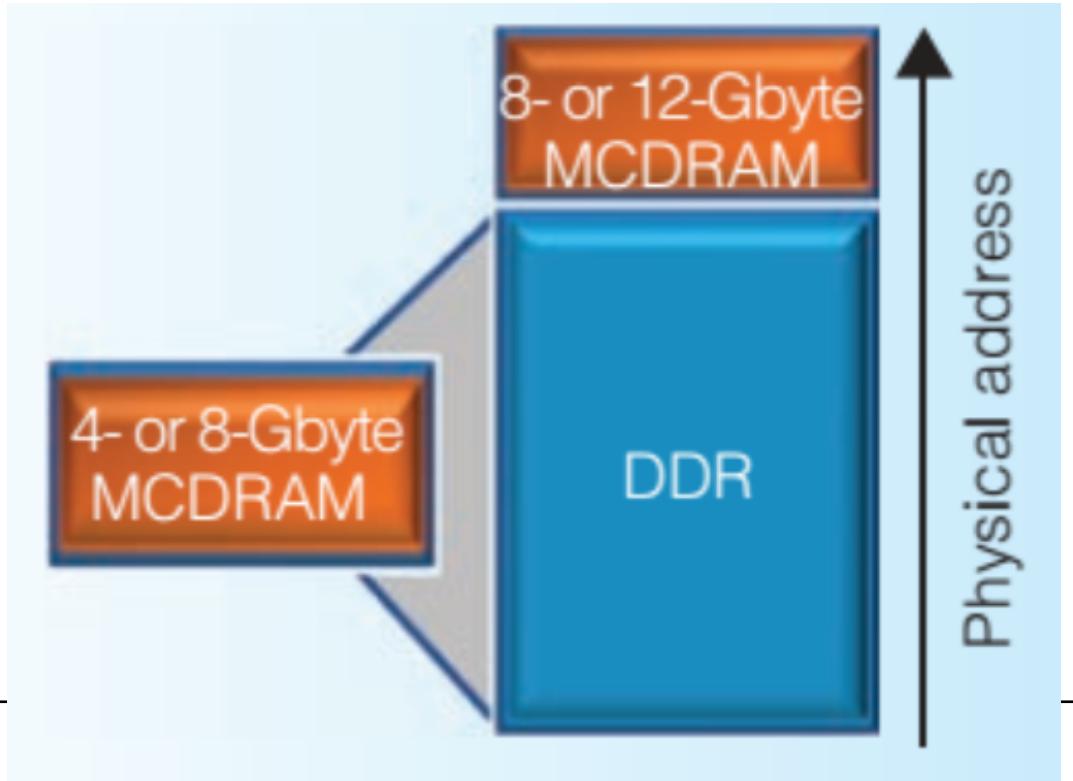


Tile

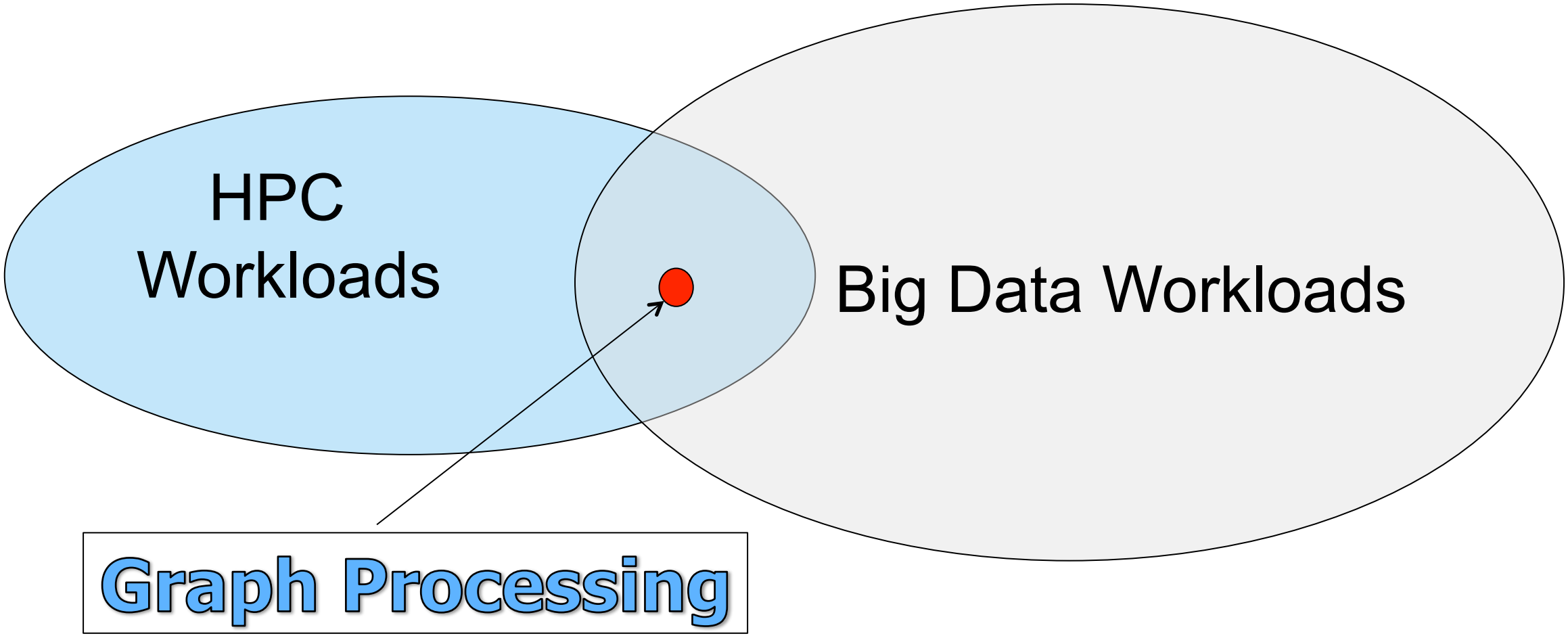
KNL – Hardware Parameter Space

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



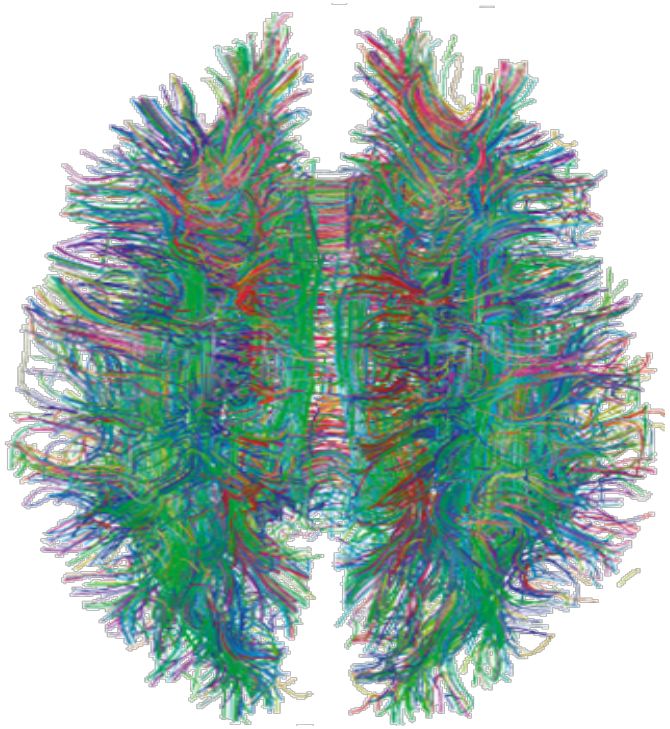


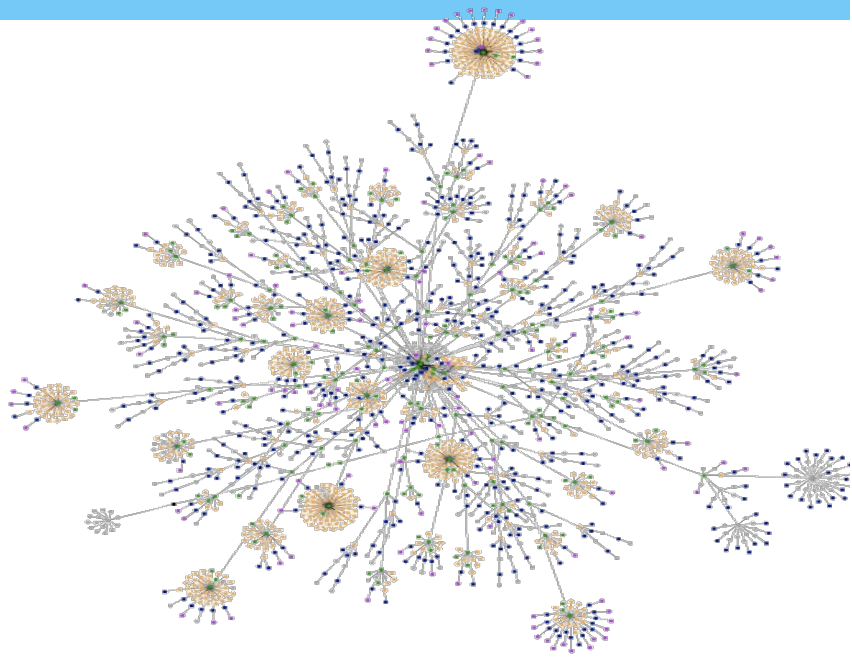
Graph Processing – HPC and Big Data



Graph Processing – High-impact Domain

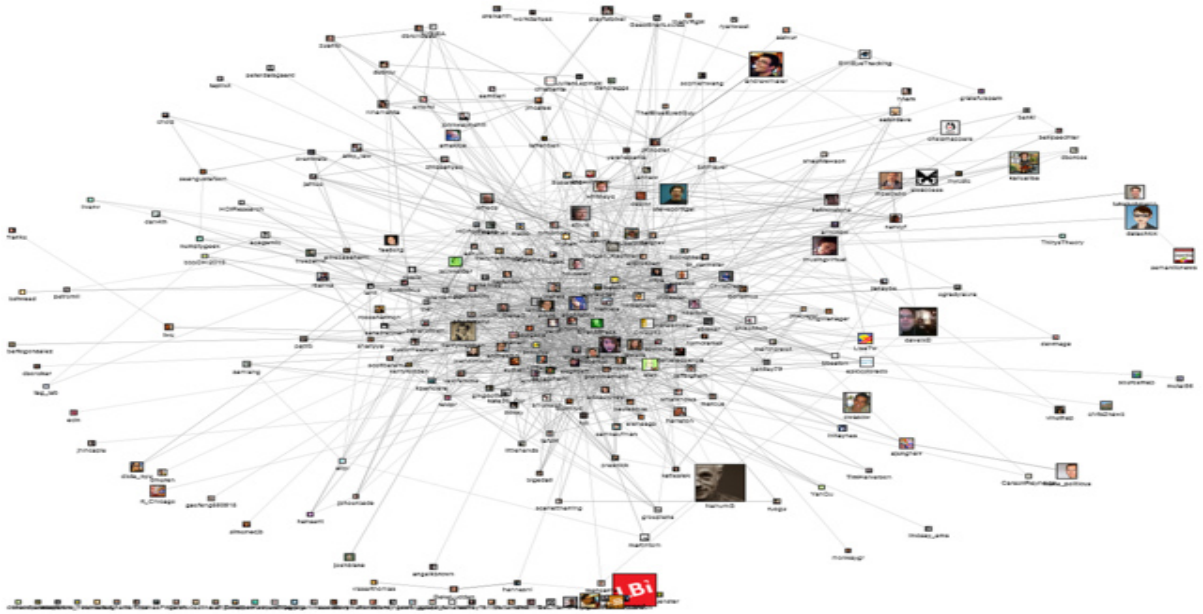
- Social networks
- Drug discovery
- Monitoring wildfires
- Combating human-trafficking
- Studying the human brain





Graph Processing – Highly Challenging

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

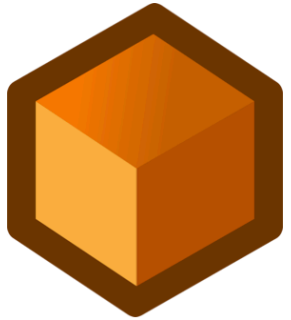


Performance = f(platform, algorithm, dataset)

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

How to study the convergence?

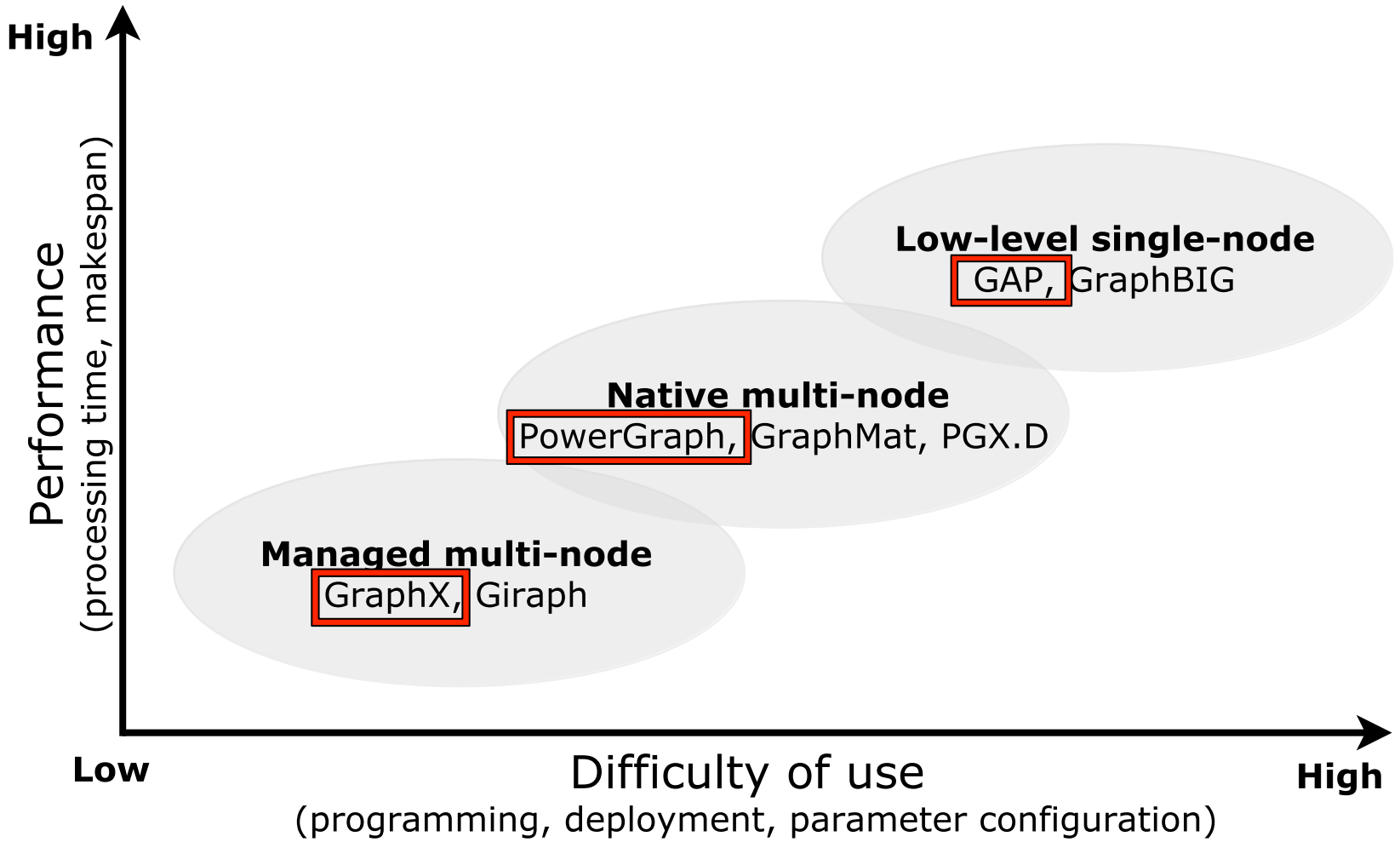
- Benchmark using Graphalytics
- Multiple classes of algorithms
- Multiple datasets (scale-free and non-scale free)
- Multiple classes of graph analytics platforms
- Comparison between KNL and de-facto big data hardware (Intel Xeon family)



Graphalytics

Open-source Graph Processing Benchmark Suite

Graph Analytics Platforms



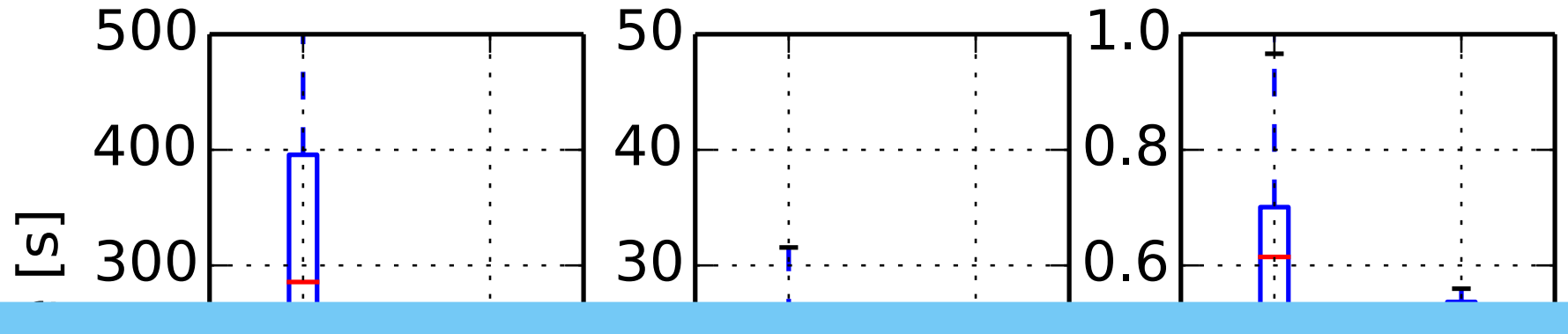
Quantifying the Convergence

- Large-scale study – over 300,000 compute core-hours
- Experiments run in DAS-5, Cartesius cluster*, Intel Academic cluster*
- **Q1: How does the KNL parameter space influence performance?**
- **Q2: How (difficult it is) to tune the platforms on KNL?**
- **Q3: Is KNL faster than Xeon?**
- **Q4: Does it scale?**

	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

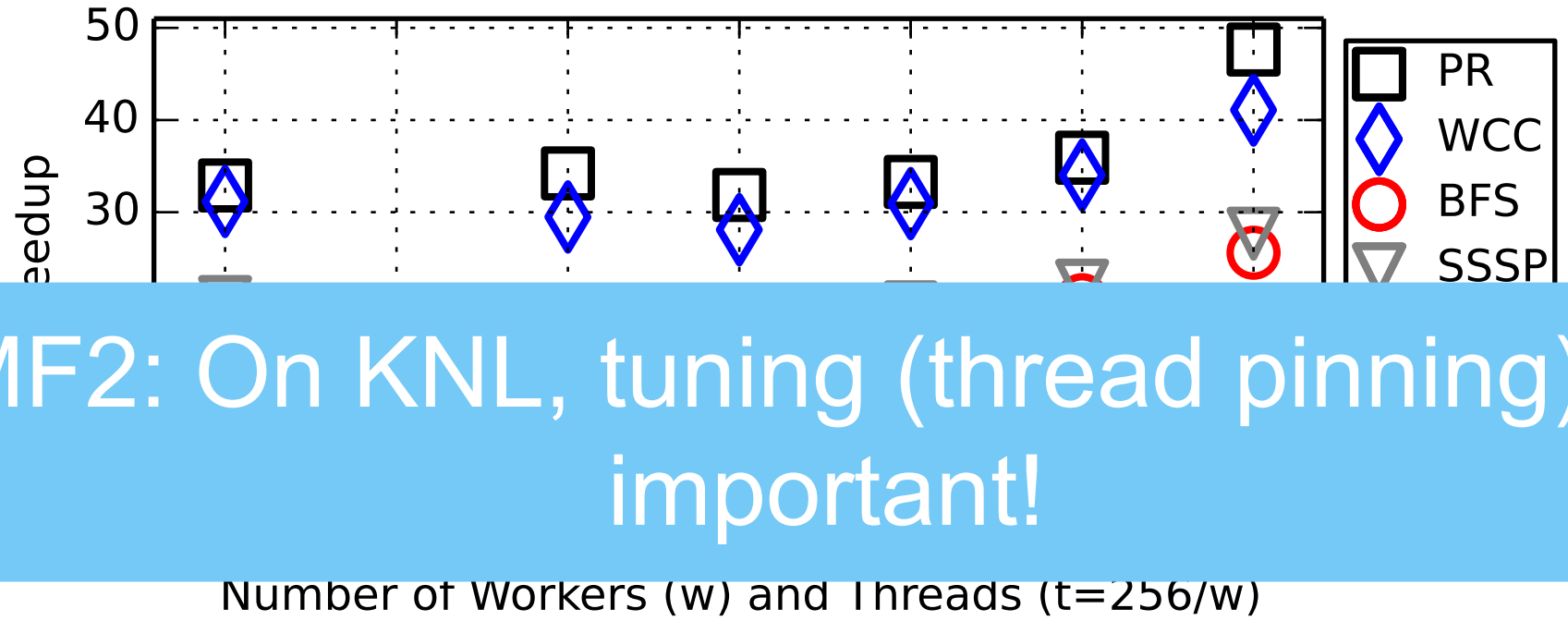
Hardware + Software Parameters



MF1: Much larger performance range due to KNL configurability and interactions with software!

(a) GraphX (b) Powergraph (c) GAP

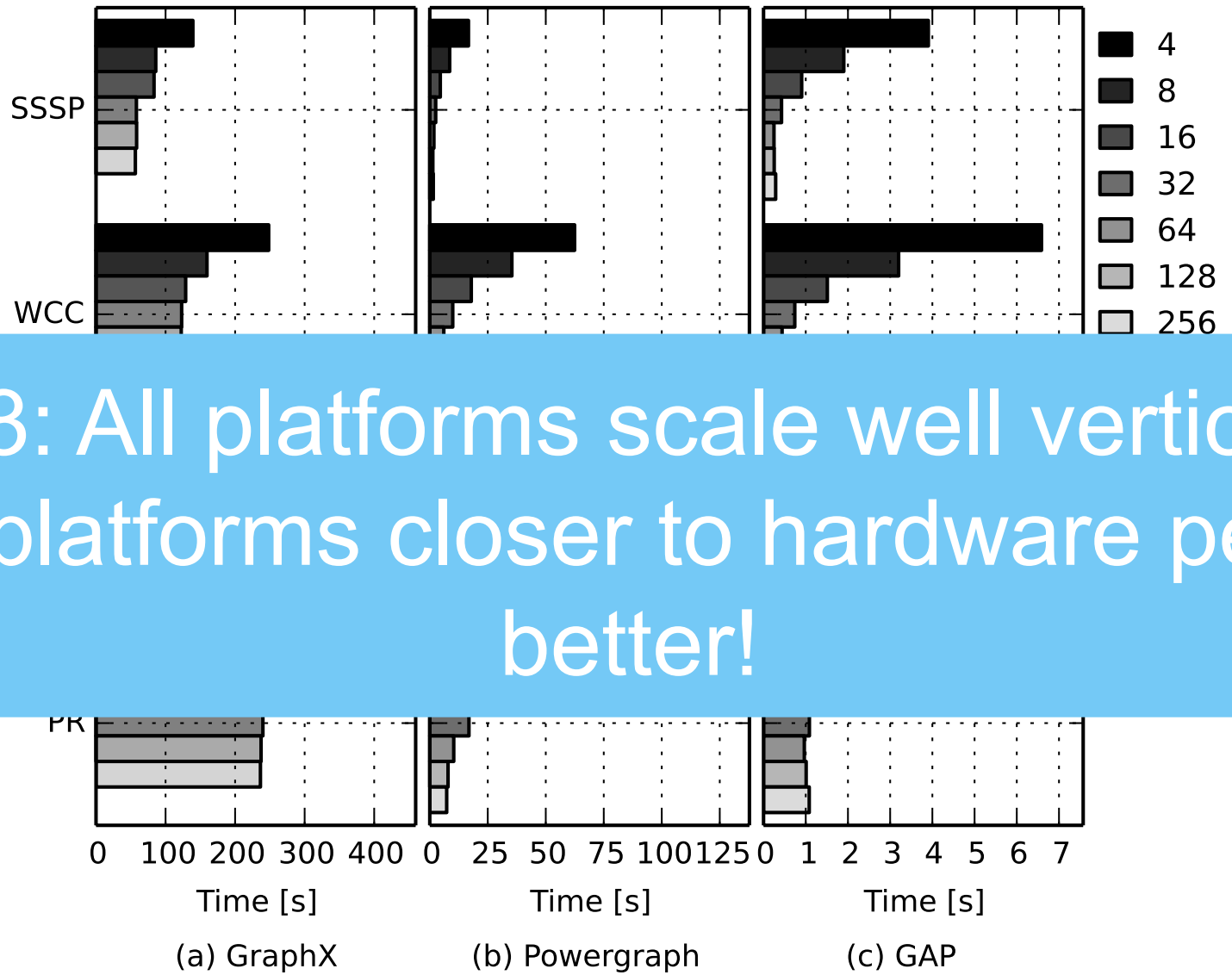
KNL Hardware + Platform Interaction and Tuning



MF2: On KNL, tuning (thread pinning) is important!

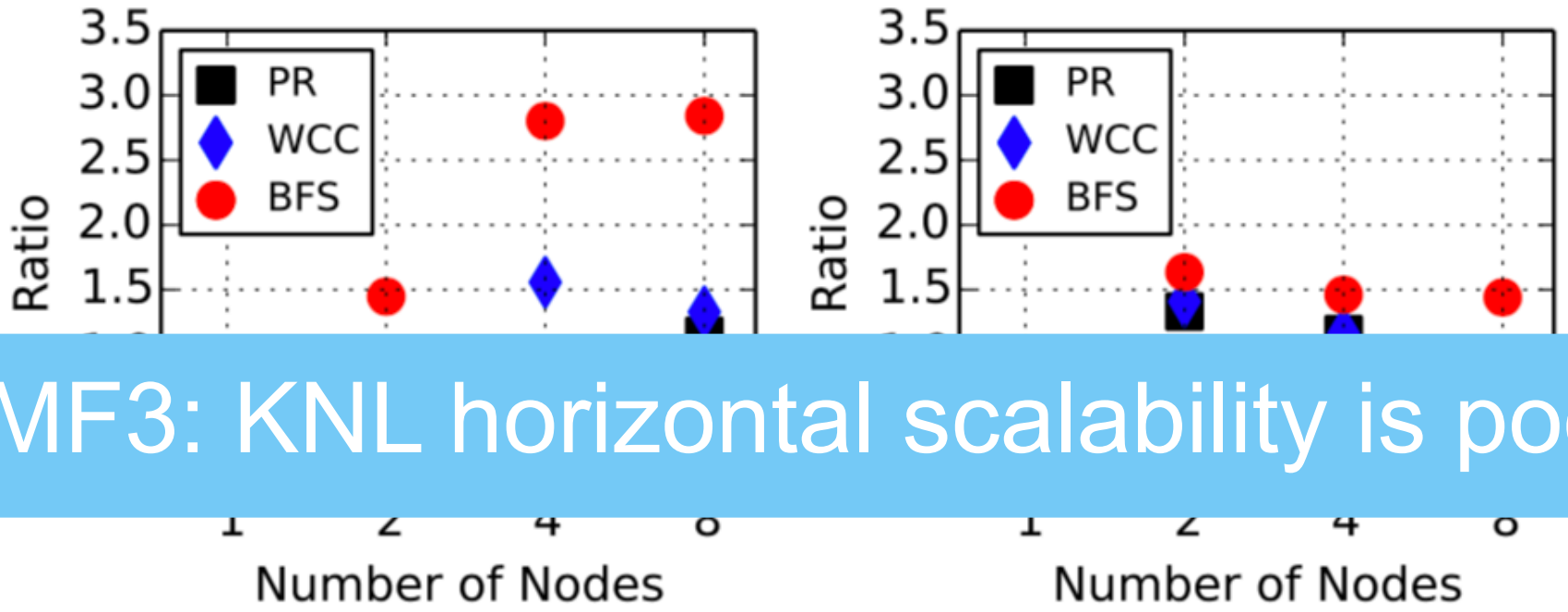
Powergraph, Datagen_7-9 – thread pinning speedup
(pinning on Xeon – 5% improvement)

KNL Vertical Scaling



MF3: All platforms scale well vertically!
MF4: platforms closer to hardware perform better!

Horizontal Scaling



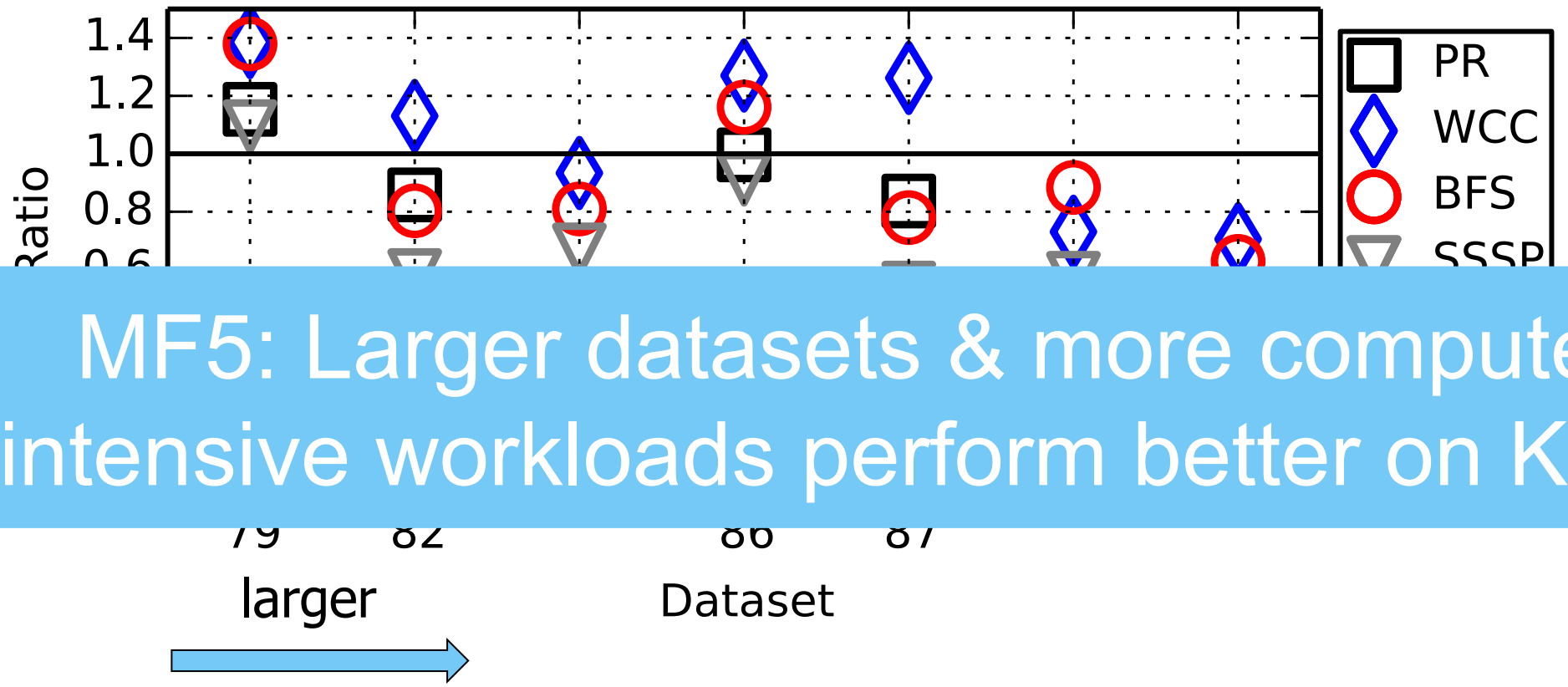
(a) GraphX.

(b) Powergraph.

MF3: KNL horizontal scalability is poor!

- KNL vs. Xeon Speedup on 1-8 nodes
- KNL single-thread networking is slow! (3X bandwidth, 8X latency)

KNL outperforms Xeon



MF5: Larger datasets & more compute-intensive workloads perform better on KNL!

GAP, KNL vs. Xeon Speedup

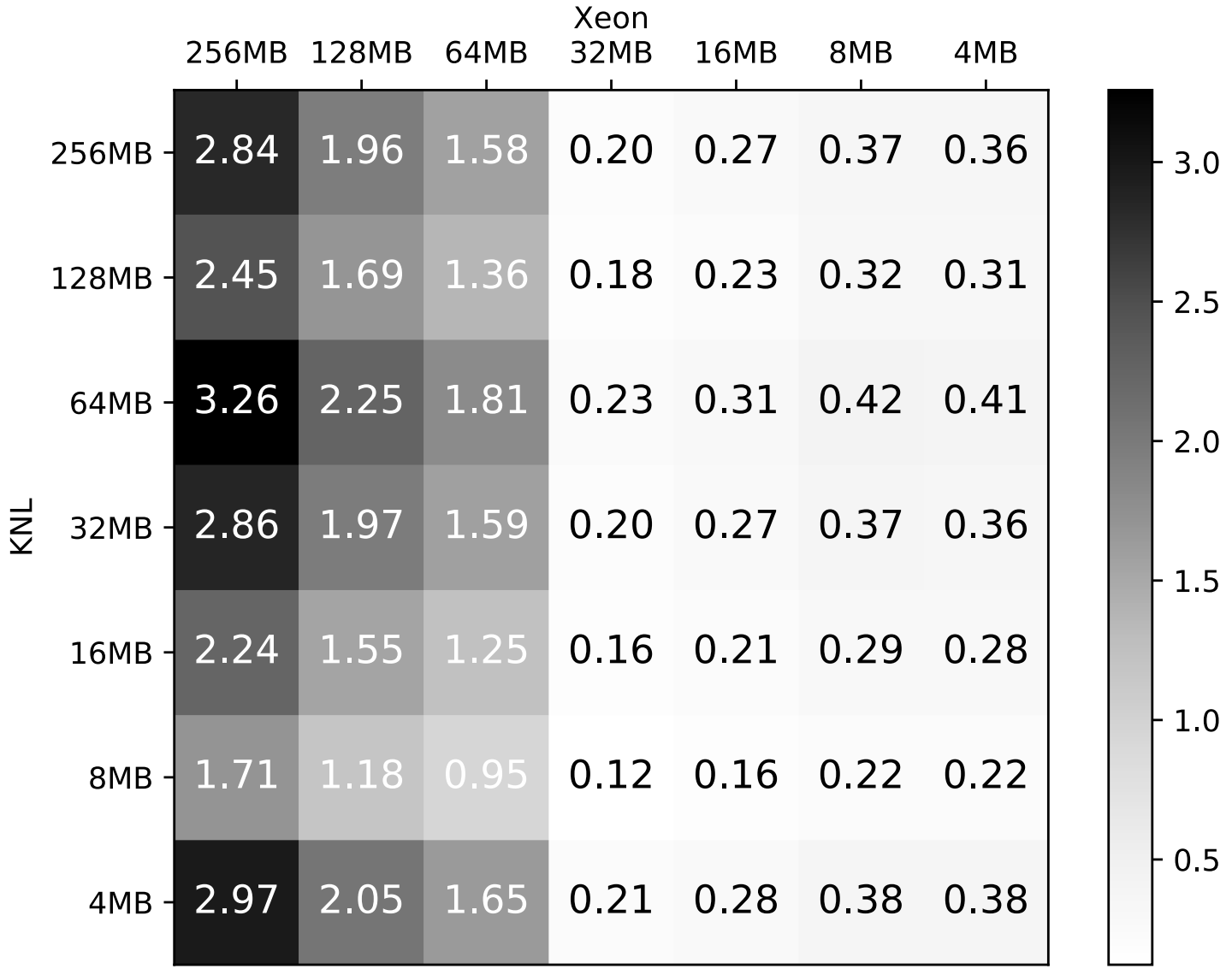
Take-home Message: Main Findings

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

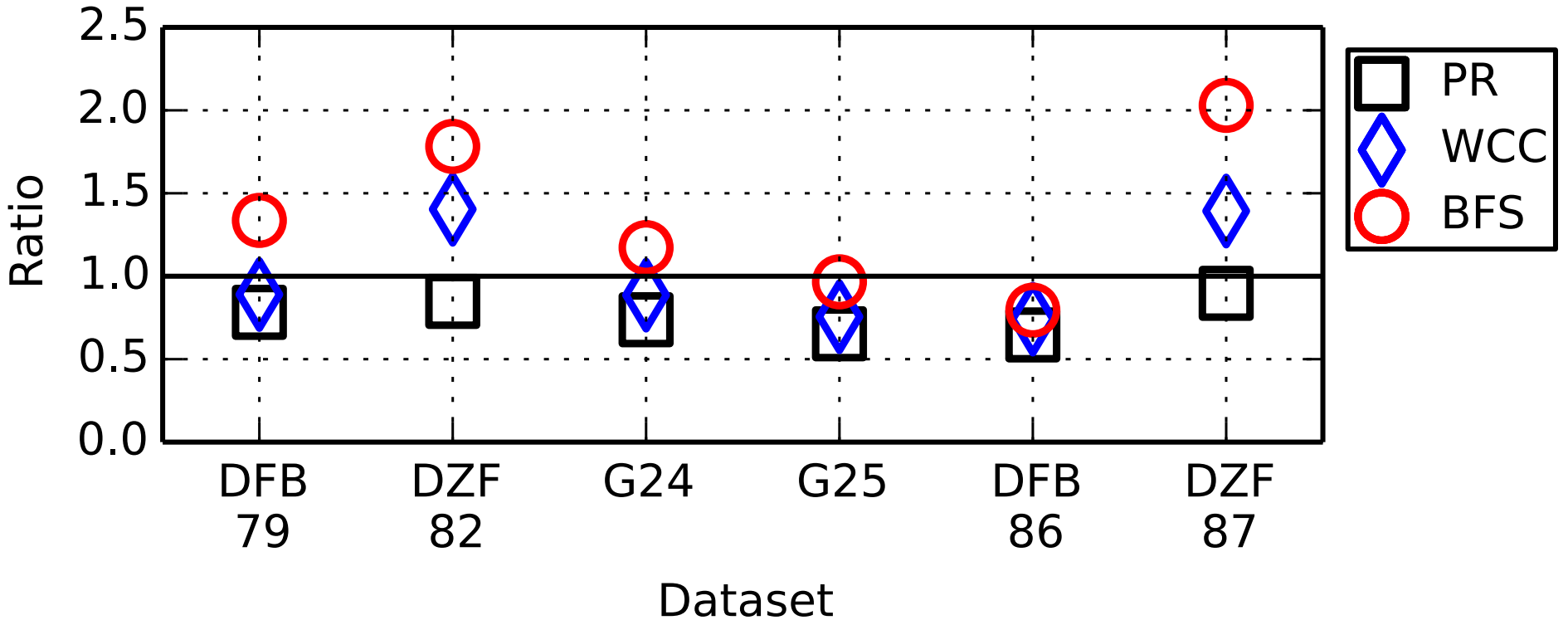
Alexandru Uta
a.uta@vu.nl

Extra Slides

Tuning GraphX



KNL vs. Xeon on Powergraph



KNL – Modes Analysis