Nanostores and Beyond

Why near-data processing might be real this time

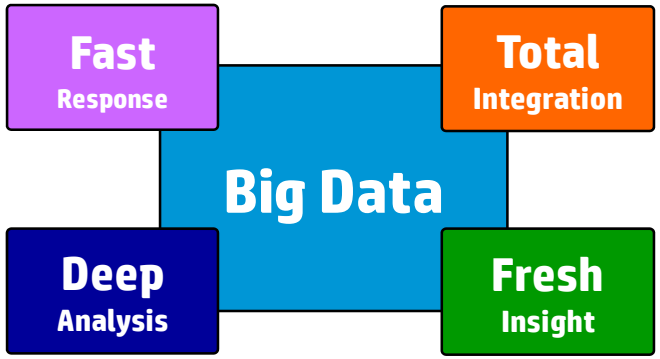
Jichuan Chang
HP Labs / Dec 8, 2013

Resource-Efficient Data-Centric Computing

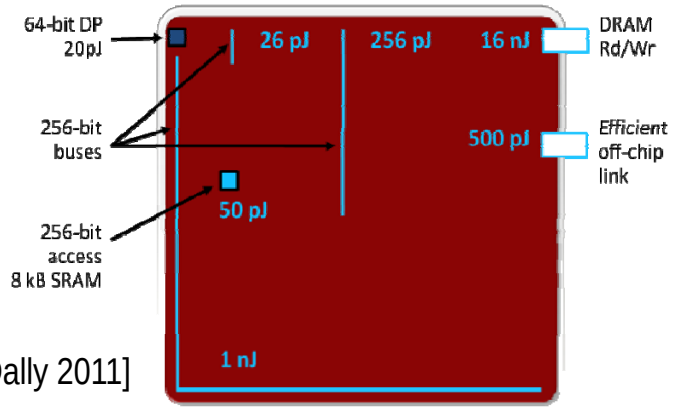
- Opportunities

16X
Moore's law

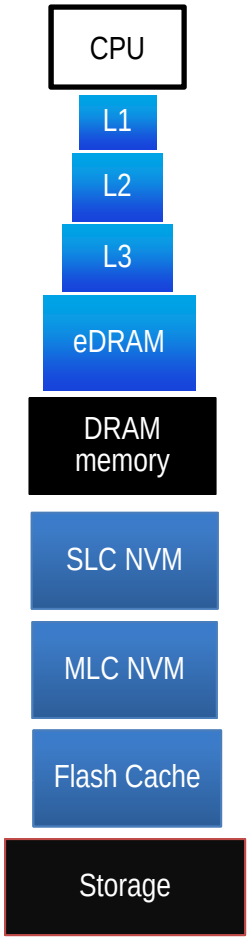
56X
online data (7-year)



- Challenges

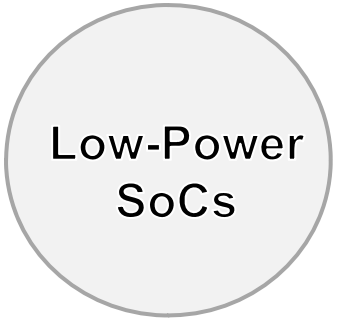


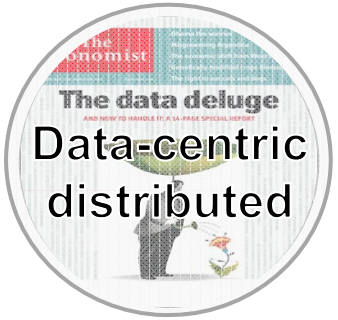
[Dally 2011]



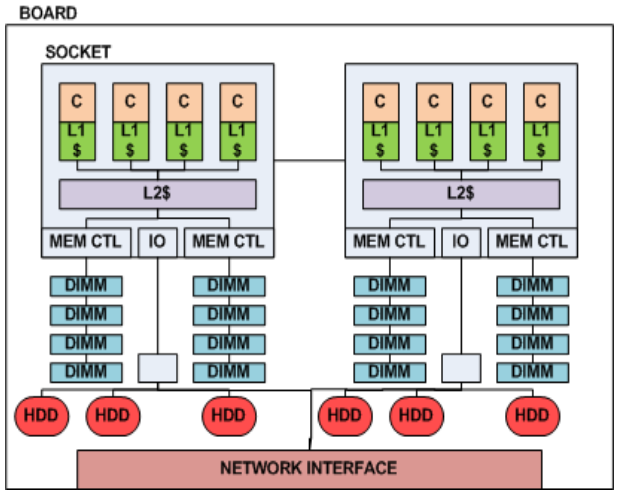
Key Design Principles

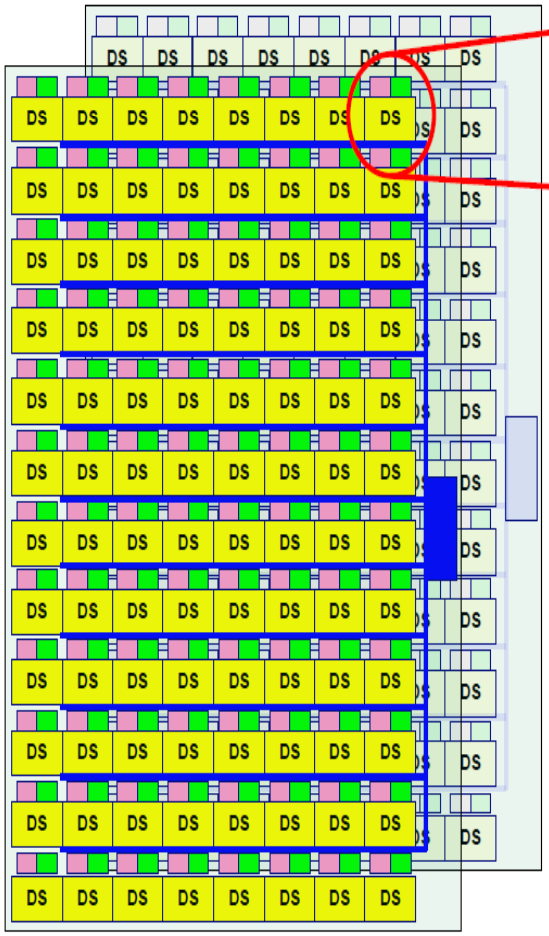
- Move and match compute to data
- Redesigning the cache/memory/storage hierarchy
- Rethink the hardware/software interfaces
- Resilience/-ilities as cross-cutting optimizations

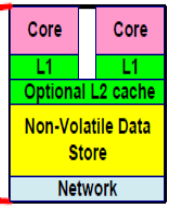


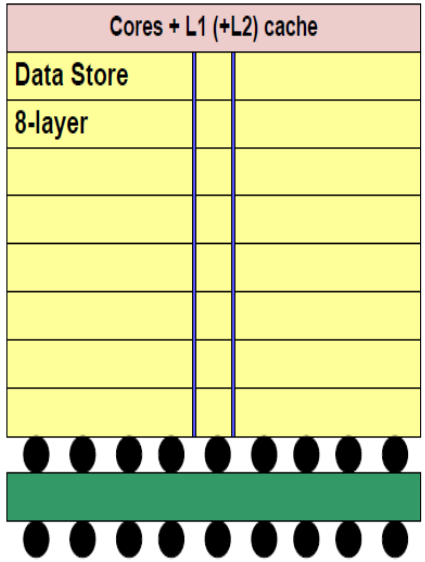


From Microprocessors to Nanostores



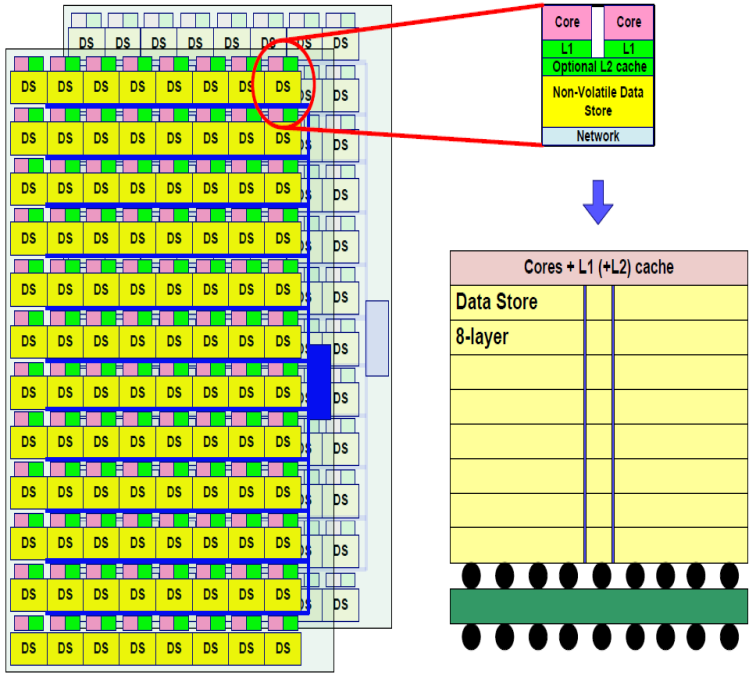




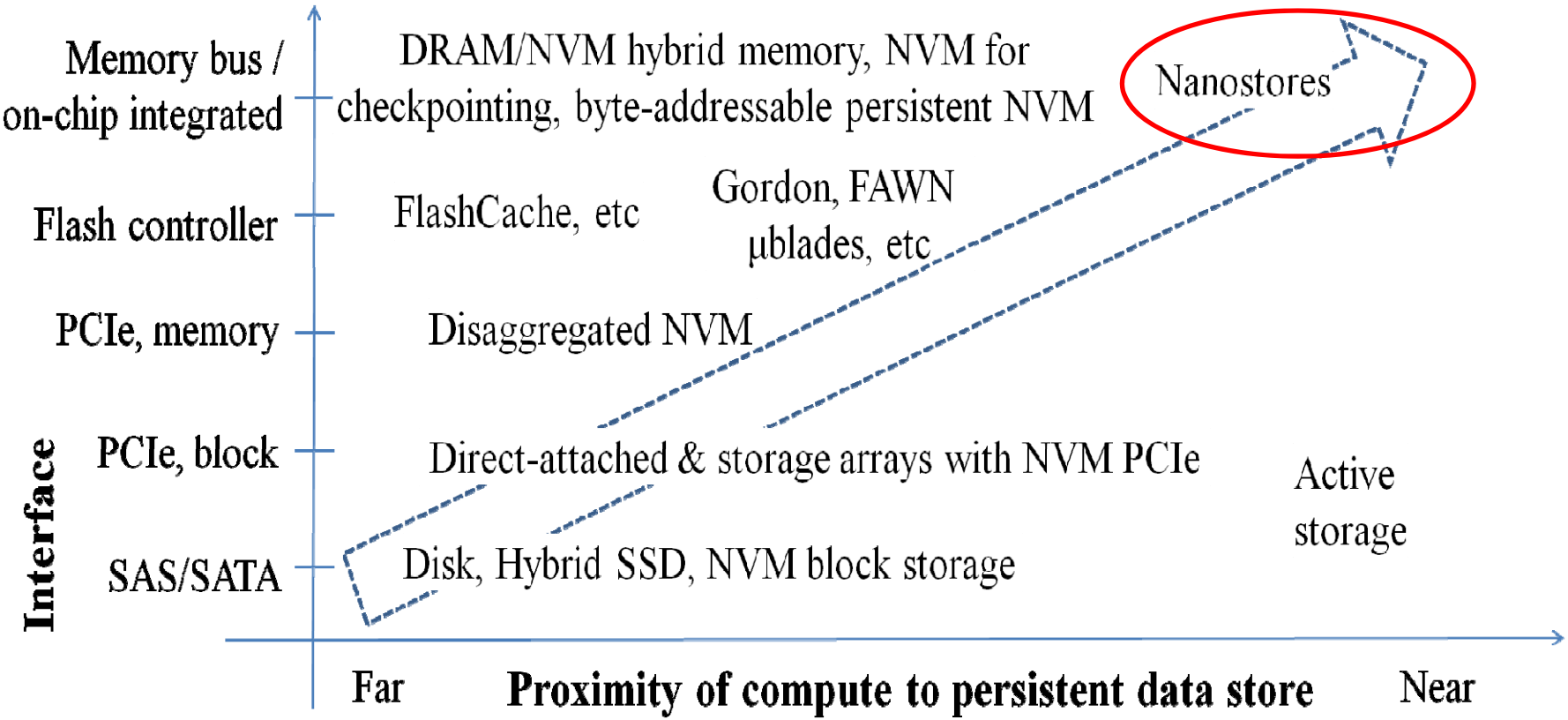


What if Data Come with Their Own Compute?

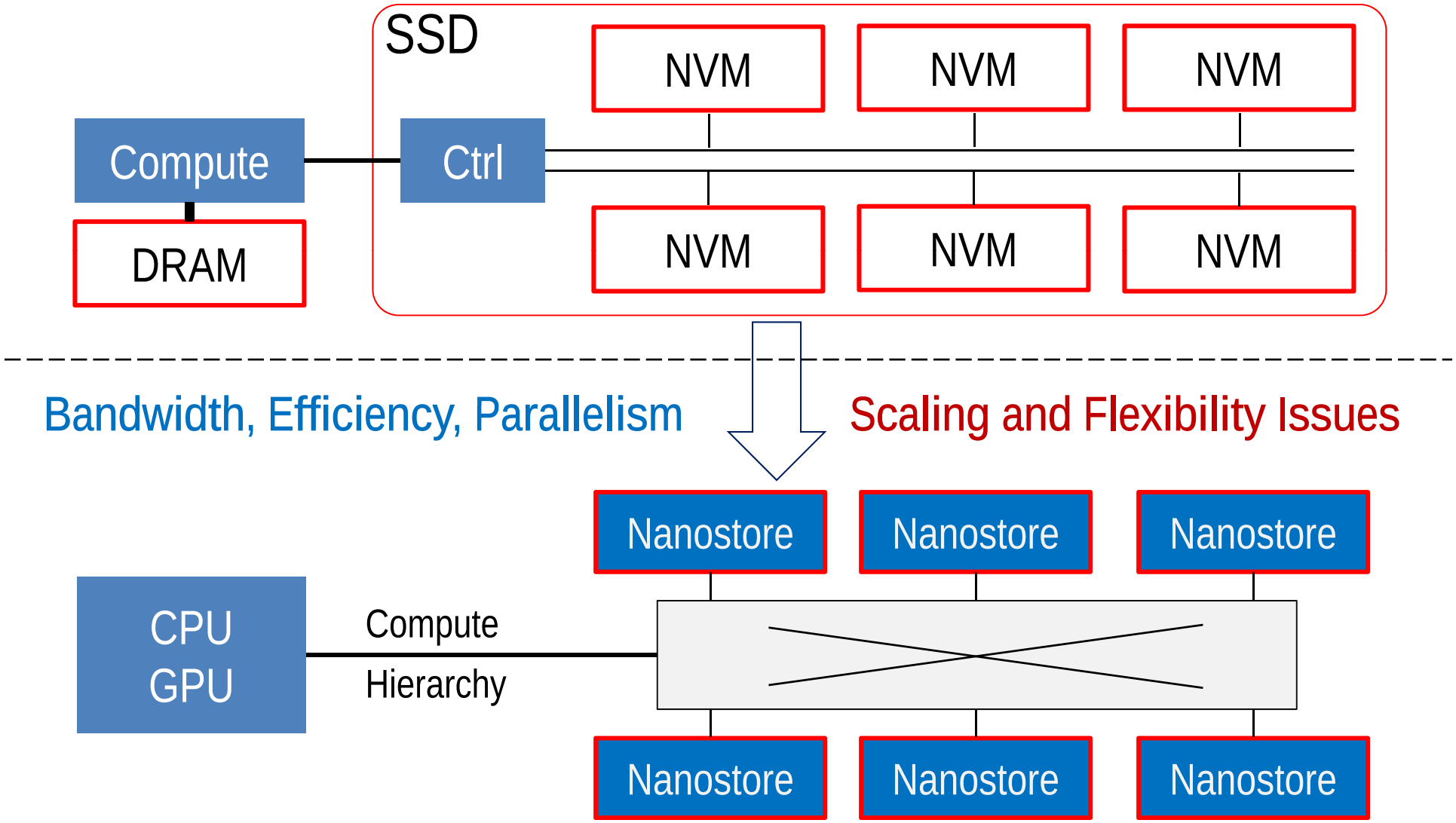
1. Leverage NVM to eliminate hard drive
2. Flatten cache / memory hierarchy
3. Co-locate compute with data stores
4. Specialize compute to match data
5. Balance compute/communicate/store
6. Compute Hierarchy and Active-*
7. Co-design software to exploit NVM
8. Match with distributed systems paradigm
9. Optimize for simplicity, volume, and efficiency



A Taxonomy of NVM-based Architectures



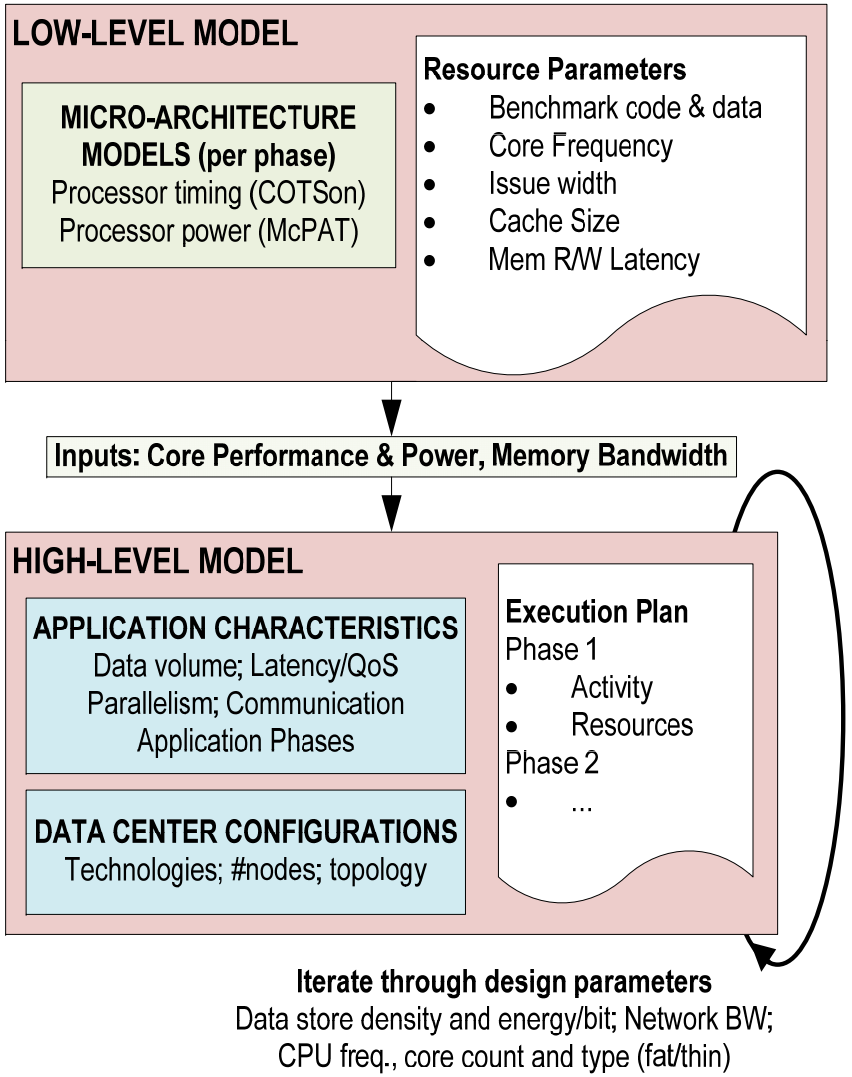
Anatomy of SSD vs. Nanostores



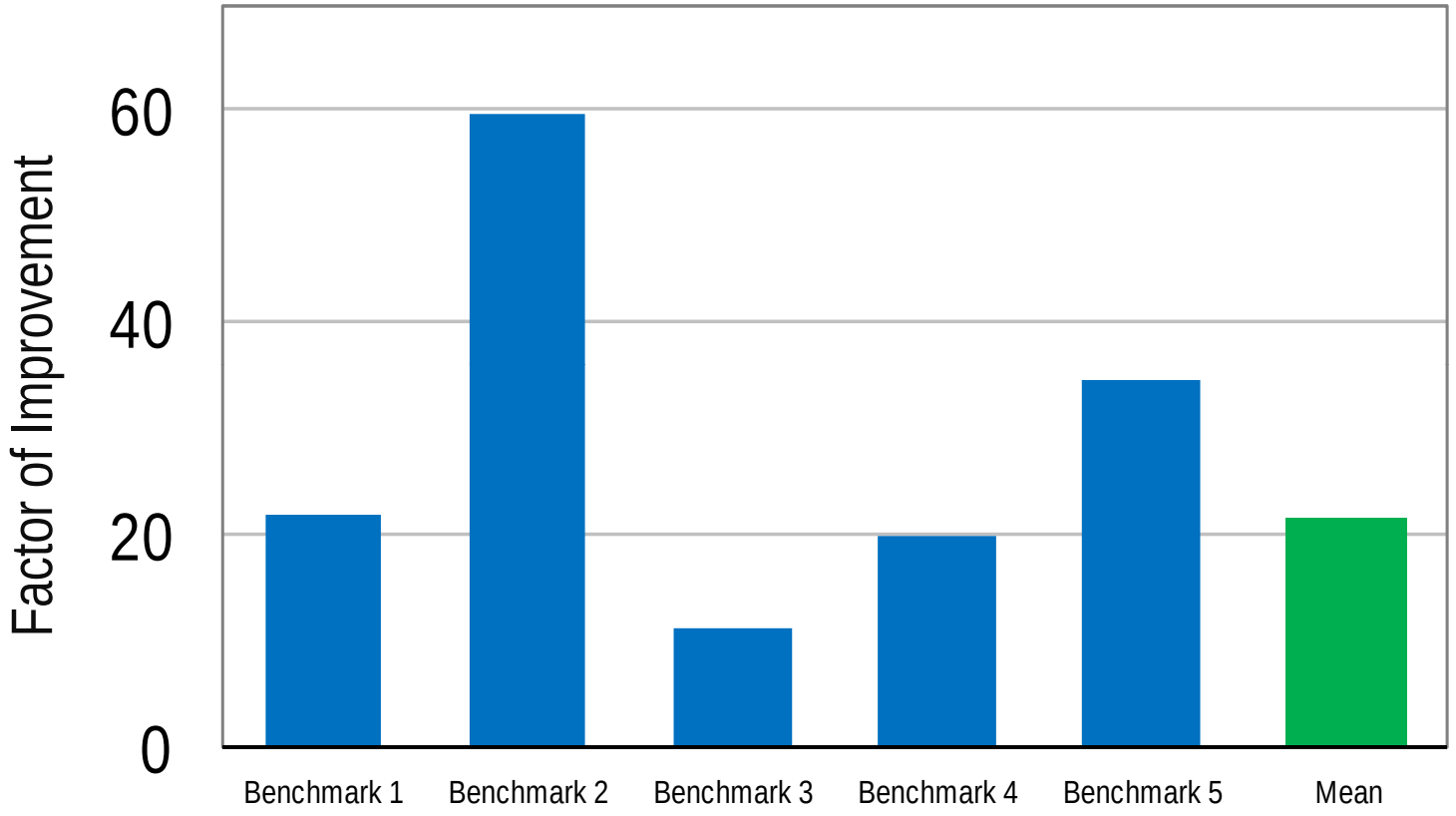
Evaluation Challenges and New Models

- Upload/download
- Search & indexing
- Transactions
- Filtering/aggregation
- Correlation
- Cubing/zoom-in
- Classification/prediction
- Social network analysis

	Access: Random	Access: Stream	Access: Permutation	Latency: Real-time	Latency: Batch	Compute: High	Compute: Low	R/W: Write heavy	R/W: Read heavy	R/W: balanced	Working Set: All	Working Set: Subset	Data: Structured	Data: Unstructured	Data: Rich media
Sort			X		X		X		X		X		X		
Search (web)	X				X	X			X		X		X	X	
Search (image)	X		X		X	X			X		X		X		
Search indexing		X			X		X		X		X			X	
Recommender	X	X	X	X			X			X		X	X		
Checksum/de-dup	X	X	X		X	X			X			X	X		
Transaction processing	X	X		X			X		X			X	X		X
Decision support	X	X		X	X	X	X	X				X	X	X	X
Video transcoding			X		X		X		X		X		X		
Mining and learning	X			X		X			X		X		X	X	X



Efficiency Benefits of Nanostores



- Balanced system design: across compute, memory, IO and network
- Future bottlenecks: network/software scalability; power/thermal

Feedbacks from Our Customers

- Positive validations on use-cases and workloads
  - Oil/gas, financial, DB/NoSQL, Storage, Events, Graph …
- Excited customers want early access to technology
  - HP Moonshot and new prototyping efforts
- More importantly: system architecture roadmap
  - Large investment in time/resource to co-design software
  - How to train programmers and what tools to use?
  - How to solve the SW vs. HW “chicken and egg” problem?

Feedbacks from Our Colleagues

- A good time to revisit PIM and Active Memory
  - Prior work: EXECUBE, PIM, IRAM, Smart Memory, DIVA, LIMA/LCMT, Active pages, FlexRAM, Pinnacle, Impulse/AMO, …
  - Funding agencies: DOE Blackcomb, DRAPA OHPC, …

- So, why it might be real this time? ☺
- And what are the new research opportunities?

Thoughts on What's Different This Time

- Assumption: both HW and SW changes are needed!

- Caveat: my personal viewpoints
- Insights from many collaborators and colleagues
  - Partha Ranganathan, Norm Jouppi, Mehul Shah, Sheng Li, Doe Hyun Yoon, Kevin Lim, Justin Meza, Greg Astfalk, John Sontag, Paolo Faraboschi, John Byrne, Laura Ramirez, Kim Keeton, Niraj Tolia, Rob Schreiber, Gilberto Ribeiro, Dwight Barron, Mitch Wright, Siamak Tavallaei, …
  - Trevor Mudge, Tom Wenisch, David Roberts, Steven Pelley, Prateek Tandon, Ron Dreslinski
  - Christos Kozyrakis, Mingyu Gao, …

Top 10 Important Reasons (1)

1. Necessity: the renewed focus on efficiency
   - Reduces data movement and cache hierarchy overhead
   - Opportunities to rebalance compute-to-memory ratios
   - Opportunities for NDP appliances and accelerators
   - Workloads and programming model are open issues
2. Technology: 3D/2.5D stacking
   - No more “merged logic/memory” dilemma
   - Enables integration, bandwidth and energy efficiency
   - Allows integration of NIC and accelerators
   - Cost and thermal issues are still open challenges

Top 10 Important Reasons (2)

3. Software: distributed software framework
   - – MapReduce as a primary example
   - – Popularized the concept of moving compute to data
   - ▪ Handles tough issues such as data layout and reliability
4. Interface: new host-memory interface support
   - – Mobile memory is the new commodity
   - – What's next? DDRx, LPDDRx, Wide I/O, HMB, HMC, …
   - – Motivations to replace strict master/slave interfaces
   - ▪ Great opportunity to add support for NDP and active-*

Top 10 Important Reasons (3)

5. Hierarchy: non-volatile memory / storage
   - Flattens the memory/storage hierarchy
   - Nanostores become a self-contained building block
     - No more choking points for getting data into the memory
6. Balance: integrated and high-BW networking
   - Renewed interest in new fabrics and photonics
   - SoC/3D enable integrated NIC and efficient stack
   - Potential new opportunity for NDP-focused networks
     - Together can enable efficient and scalable platforms

Top 10 Important Reasons (4)

7. Heterogeneity: recent trends have paved the road
   - Heterogeneity needed for specialization and flexibility
   - This use to one of the key barriers to incorporate NDP
     - GPU/APU, bigLITTLE, FPGA/SoC, … NDP ☺
8. Capacity: key to lower cost and divisibility
   - Small capacity once required high || and data movement
   - New compute-to-capacity ratios enabled by NVM, etc
     - Lower-cost memory is also important for adoption

Top 10 Important Factors (5)

9. Anchor workloads: co-designed big data systems
   - Databases: IBM Netezza, Oracle Exadata, …
   - New frameworks: MapReduce, Impala, …
   - Other possibilities: Financial/Geo/Telco, graphs, events …
   - Commercially-viable anchor market is critical
10. Ecosystem: prototypes and tools now available
   - SW: OpenCL, MapReduce, …
   - HW: Adapteva, Vinray, Micron's AP, Samsung SSD …
   - Tools: extend multicore & heterogeneous computing tools

Summary

From microprocessors to Nanostores

- Resource-efficient data center computing
- Moving and matching compute to data
- Promising results and feedbacks received

Why near-data processing might be real this time?

- Top 10 reasons (my take on this)
- Open questions remain, but exciting times ahead …
- Your insights and feedbacks are most appreciated!

Top 10 Reasons for NDP2.0

1. Necessity: the renewed focus on efficiency
2. Technology: 3D/2.5D stacking
3. Software: distributed software framework
4. Interface: new host-memory interface support
5. Hierarchy: non-volatile memory / storage
6. Balance: integrated and high-BW networking
7. Heterogeneity: recent trends have paved the road
8. Capacity: key to lower cost and divisibility
9. Anchor workloads: co-designed big data systems
10. Ecosystem: prototypes and tools now available