Hierarchies, Clouds, and Specialization

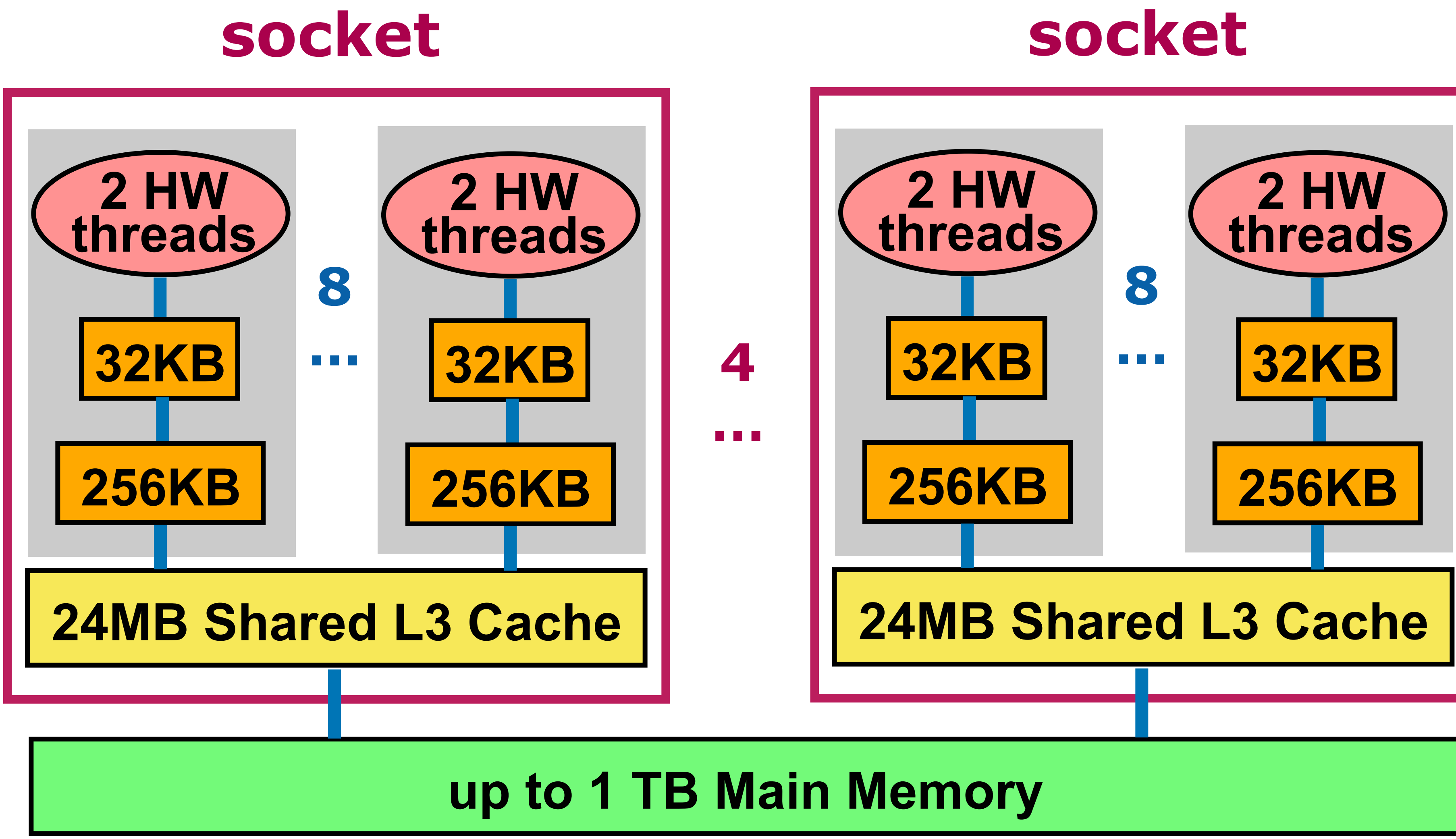
Phillip B. Gibbons
Intel Labs Pittsburgh

June 28, 2012

Intel Science & Technology
Center for Cloud Computing

NSF Workshop on Research Directions in the Principles of Parallel Computation

Research Direction #1: Hierarchies



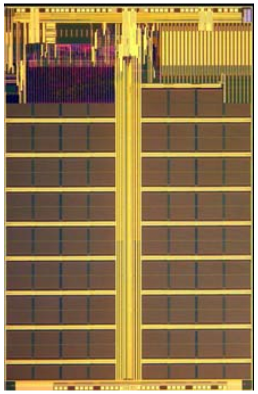
Attach: Magnetic Disks & Flash Devices

Xeon 7500 Series (Nehalem)

Hierarchy Trends

- **Good performance [energy] increasingly requires effective use of hierarchy**

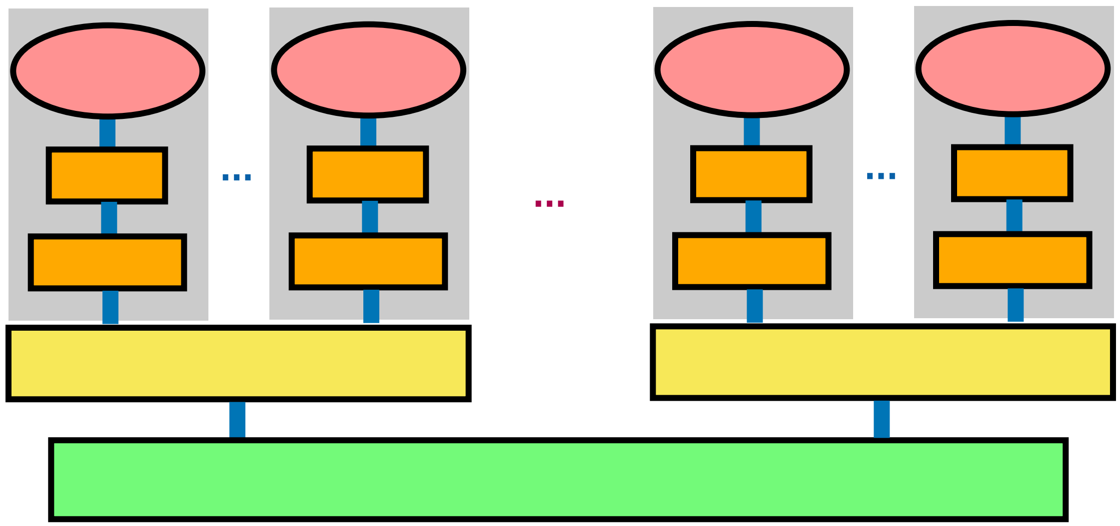
- **Hierarchy getting richer**
  - More levels of cache
  - More cores
  - New memory/storage technologies
    - Flash/SSDs, emerging PCM
    - Bridge gaps in hierarchies – can't just look at last level of hierarchy



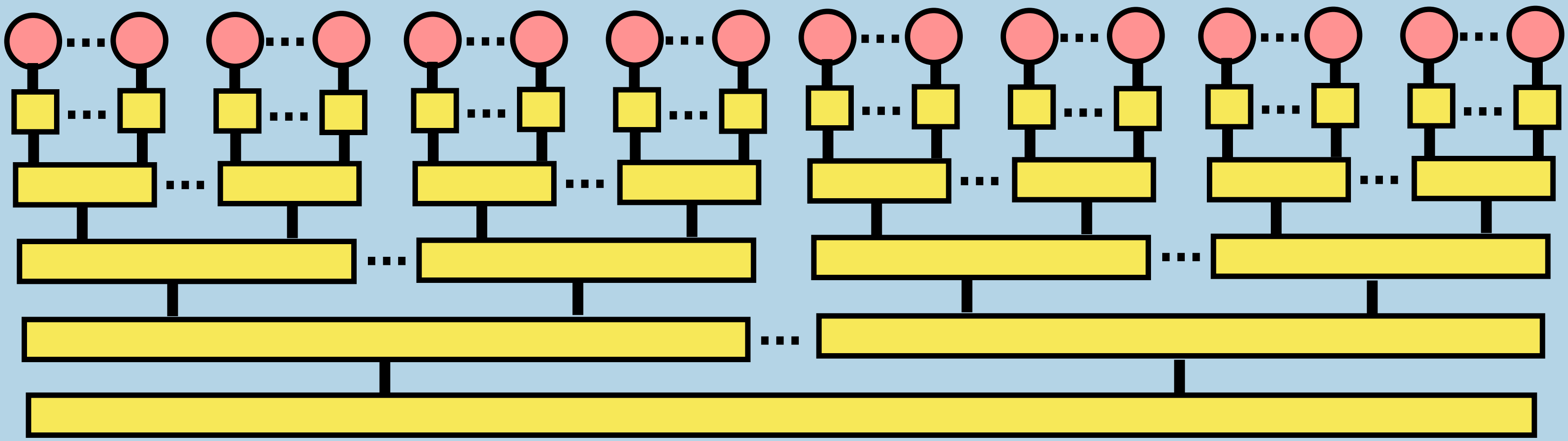
How to Model the Hierarchy?

Specific Example:

Xeon 7500



General Abstraction: Tree of Caches

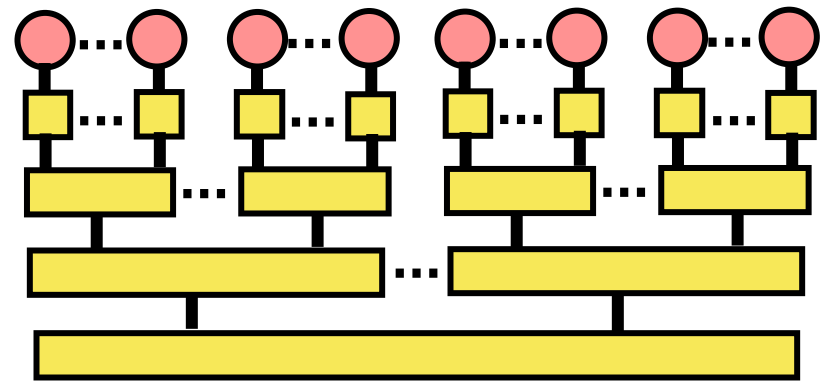


PMH model [ACF93]

How to Design Algorithms?

Design to Tree-of-Caches abstraction:

- **Multi-BSP Model** [L.G. Valiant, ESA'08]
  - 4 parameters/level: cache size, fanout, latency/sync cost, transfer bandwidth
  - Bulk-Synchronous



Our Goal:

- **~ Simplicity of Cache-Oblivious Model**
  - Handles dynamic, irregular parallelism
  - Co-design with smart thread schedulers

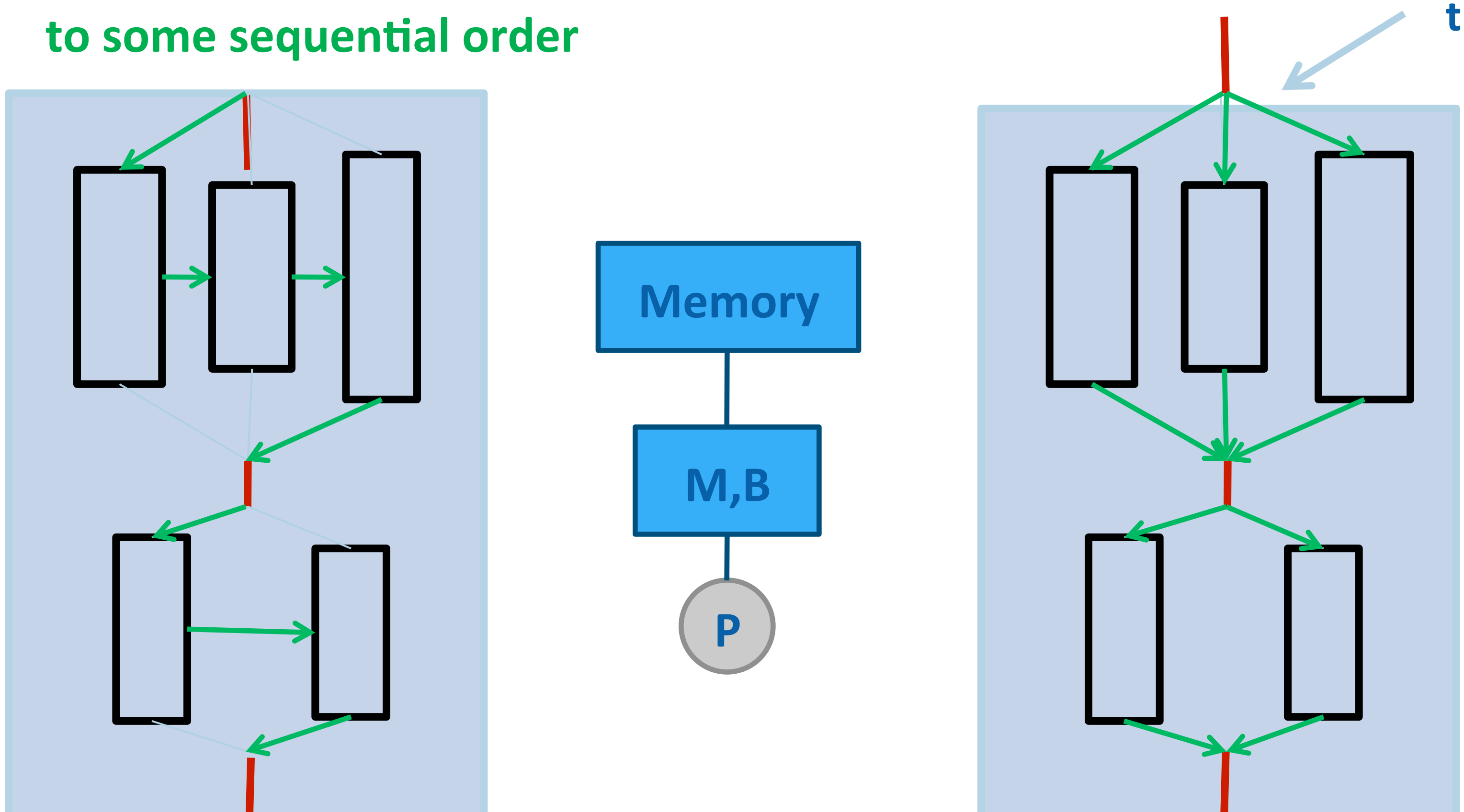
Fundamentals: Spatial locality, Temporal locality, Constructive sharing, Minimize communication

What is Algorithm Designer's Model?

Parallel Cache-Oblivious Model

Differs from CO model in how cache state is carried forward

Carry forward cache state according to some sequential order



Assuming this task fits in cache

All three subtasks start with same state

Merge state and carry forward

If task does not fit, subtasks start with empty cache

What is the Right Scheduler?

- **Following [CSBR10], we study "Space-Bounded Schedulers"**
  - Pin tasks to cache where fits
  - Allocate work to cores sharing that cache

- **Can prove good cache bounds, but overheads in current implementation too high:**
  - Faster to use Work-Stealing Scheduler

What algorithm model + thread scheduler can rule the world?

Research Direction #2: Clouds

- **Trend: Most computing is moving to the Cloud**
- **Trend: Large-scale Parallel Computing done using MapReduce/Hadoop or follow-ons**
  - Bulk-synchrony is more popular than ever ☺
- **Trend: Computations span client to cloud**
  - Moving towards tiered clouds
- **Trend: Multi-tenancy using virtual machines**
  - Impact on alg. design / schedulers?

Intel Science & Technology Center for Cloud Computing

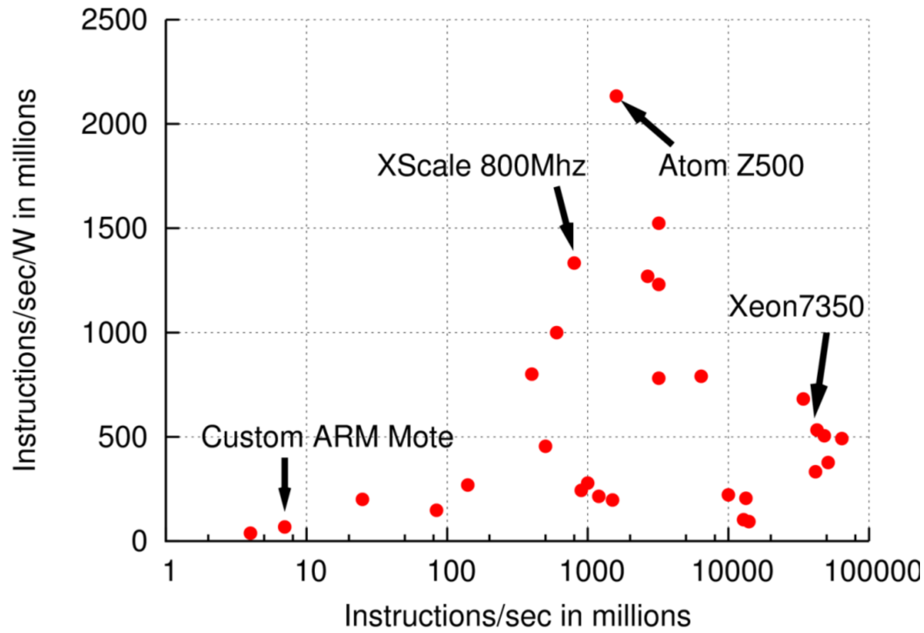
Dealing with Multi-tenancy & VMs?

- **Algs must be suited for dynamic resources**
  - Avoid HPC-think: don't know network topology
- **Schedulers must provide throughput + fairness**
  - Failed steal attempts not useful work
  - Yielding at failed steal attempts leads to unfairness
  - BWS [Eurosys'12] decreases average unfairness from 124% to 20% and increases thruput by 12%
- **How is VM an obstacle to high level performance abstractions?**
  - What hooks are needed to remove the obstacles?
  - Algs/PL community has vision of goal—can inform

Research Direction #3: Specialization

Specialization is key to cloud efficiency

- Data center with mix of server architectures
- Heterogeneity within each processor



Algorithm designers should study specialization

- For algorithms to be run in the cloud
- For algs that support cloud infrastructure

BACK-UP SLIDES: REFERENCES

References - I

Smart thread schedulers can enable simple, hierarchy-savvy abstractions

- PDF scheduler for shared caches [SPAA'04]
- Scheduling for constructive sharing [SPAA'07]
- Controlled-PDF scheduler [SODA'08]
- Work stealing overheads beyond fork-join [SPAA'09]
- Hierarchy-savvy parallel algorithms [SPAA'10]
- Parallel cache-oblivious model & scheduler [SPAA'11]

Tools, Hooks, Determinism simplify programming

- Memory-block transactions [SPAA'08]
- Semantic space profiling /visualization [ICFP'08, JFP2010]
- Efficient internal determinism [PPoPP'12]

References - II

Flash/PCM-savvy (database) systems maximize benefits of Flash/PCM

- **Flash-savvy algorithms [VLDB'08, PVLDB 2010]**
- **Flash-based OLTP transactional logging [SIGMOD'09]**
- **Non-blocking joins for Data Warehouses [SIGMOD'10]**
- **Efficient online updates for Data Warehouses [SIGMOD'11]**
- **PCM-savvy database algorithms [CIDR'11]**

Cloud Computing / Specialization

- **Scheduling for multi-tenancy [EuroSys'12]**
- **Cloud ISTC whitepaper** [www.istc-cc.cmu.edu/publications/papers/2011/ISTC-Cloud-Whitepaper.pdf]

See www.pittsburgh.intel-research.net/people/gibbons/gibbons-cv.pdf for authors & paper titles of each reference