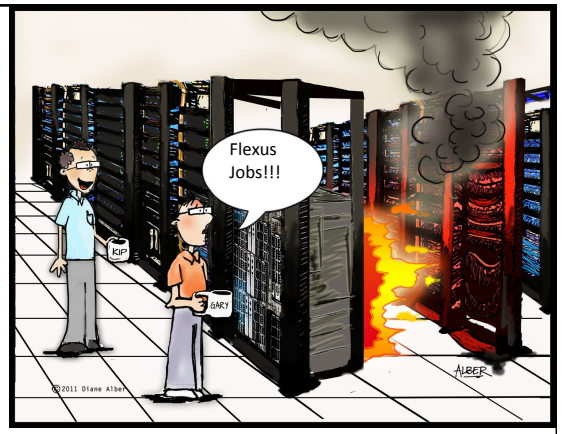
ACA

Servers

Fall 2016



Pejman Lotfi-Kamran

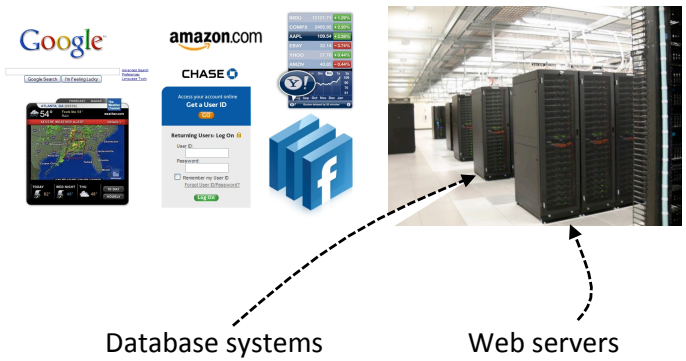
Adapted from slides originally developed by Profs. Hill, Hoe, Falsafi and Wenisch of CMU, EPFL, Michigan, Wisconsin

Where Are We?

Fr	Sa	Su	Mo	Tu
	27-Shahrivar		29-Shahrivar	
	3-Mehr		5-Mehr	
	10-Mehr		12-Mehr	
	17-Mehr		19-Mehr	
	24-Mehr		26-Mehr	
	1-Aban		3-Aban	
	8-Aban		10-Aban	
	15-Aban		17-Aban	
	22-Aban		24-Aban	
	29-Aban		1-Azar	
	6-Azar		8-Azar	
	13-Azar		15-Azar	
	20-Azar		22-Azar	
	27-Azar		29-Azar	
	4-Dey		6-Dey	

- This Lecture
  - Server
- Next Lecture:
  - Data Center

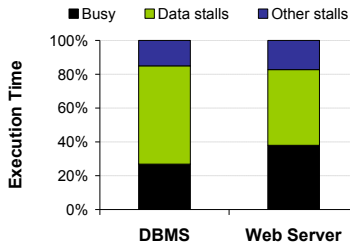
Modern Computing



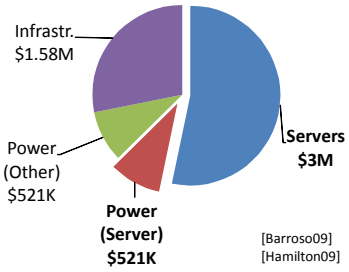
➡ Server applications: the backbone of computing

But, Servers Are Wasting $$$

Inefficient computation



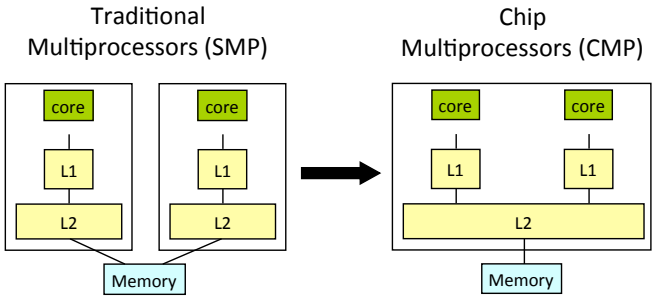
Servers are expensive



[Barroso09]
[Hamilton09]

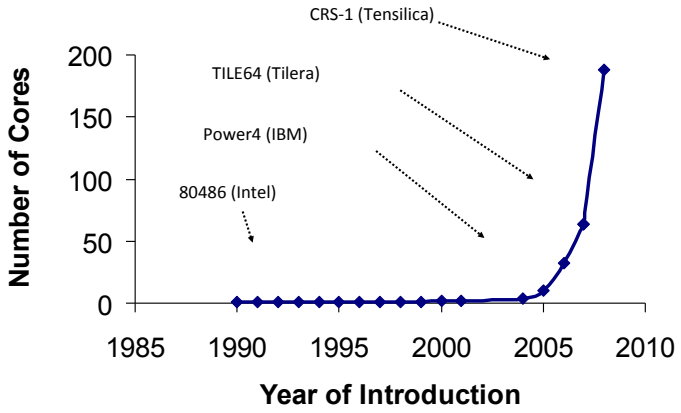
➡ Must optimize processors for server workloads

Hardware Integration Trends

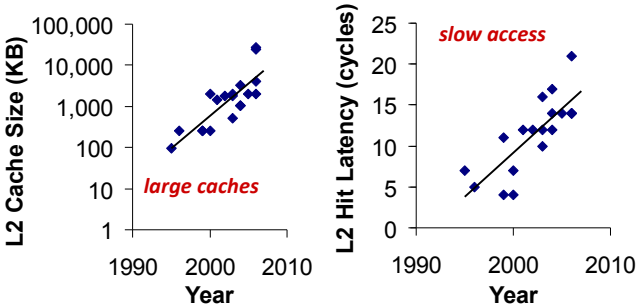


- Moore's Law: 2x trans. = 2x cores, 2x caches
- Trends to use larger but slower caches

Hardware parallelism grows



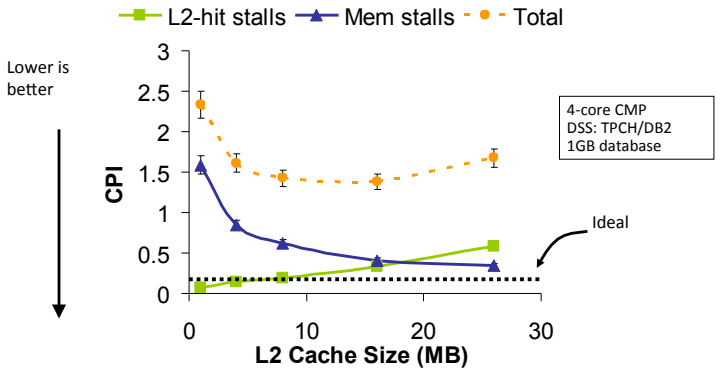
Cache sizes grow



- Larger cache → lose 50% performance!

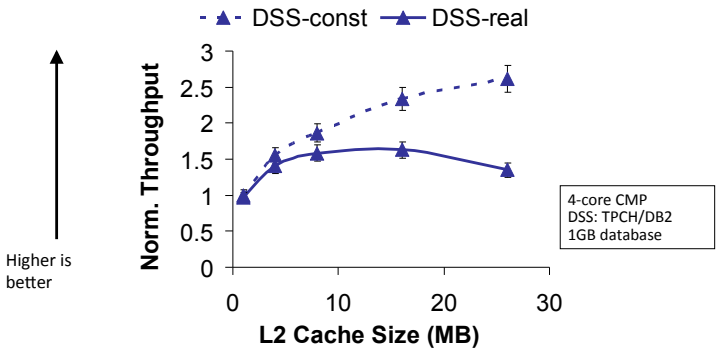
Cache Accesses Dominate Execution

[CIDR'07]



- Bottleneck shifts from memory to L2-hit stalls

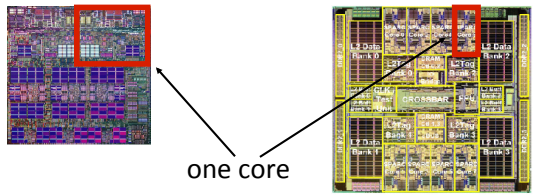
Lose Half the Potential Throughput



➧ Increasing latency forces caches to become distributed

Parallelism in Modern CMPs

- **Fat Camp (FC)**
  wide-issue, OOO
  e.g., IBM Power5
- **Lean Camp (LC)**
  in-order, multi-threaded
  e.g., Sun UltraSparc T1

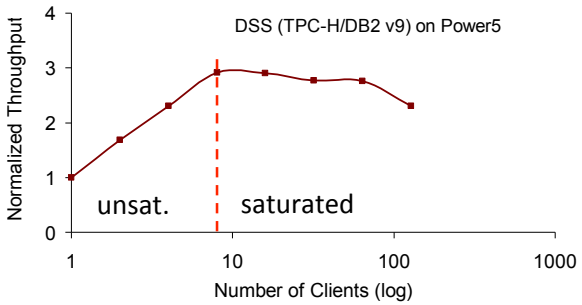


➧ FC: parallelism within thread, LC: across threads

Processor Designs Taxonomy

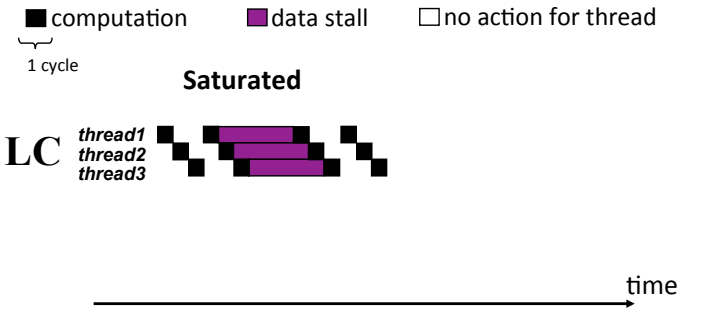
Technology	Fat Camp (FC)	Lean Camp (LC)
Issue Width	Wide (4+)	Narrow (1-2)
Execution Order	Out-of-Order	In-order
Pipeline Size	Deep (14+)	Shallow (5-6)
HW Threads	Few (1-2)	Many (4+)
Core Size	Large (3×LCSize)	Small (LCsize)

Workload Taxonomy



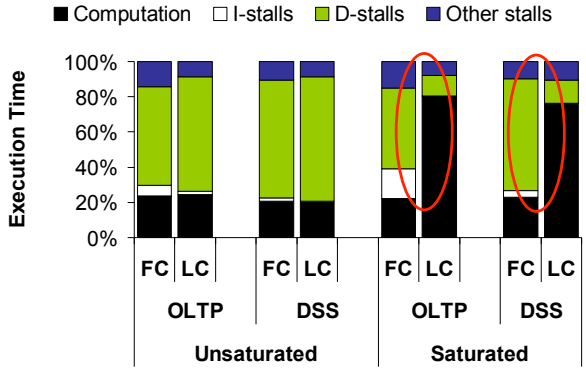
- **Unsaturated**: there exist idle hardware contexts
- **Saturated**: core always finds user threads to run

How Camps Reduce Stalls



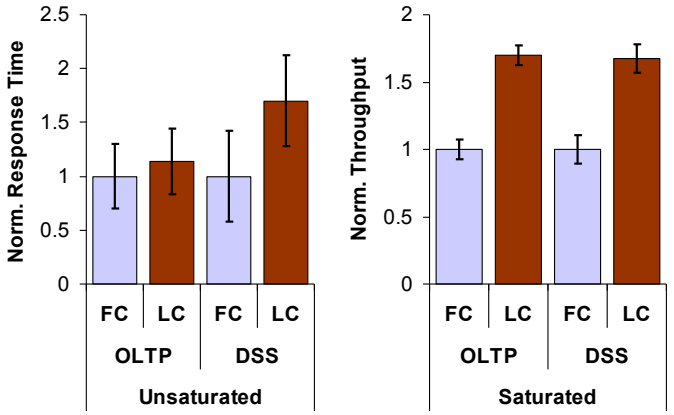
- LC: stalls can dominate under unsaturated
- FC: stalls exposed in all cases

Prevalence of Data Stalls

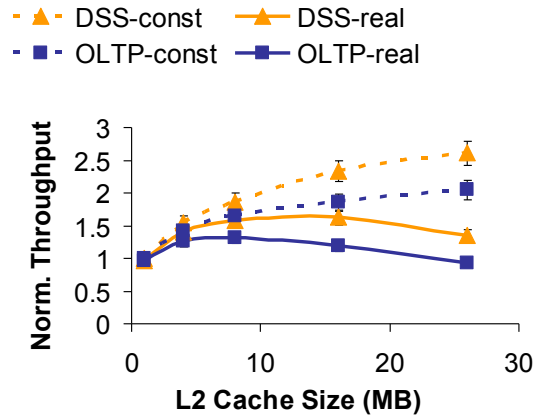


- LC good, but only under heavy load

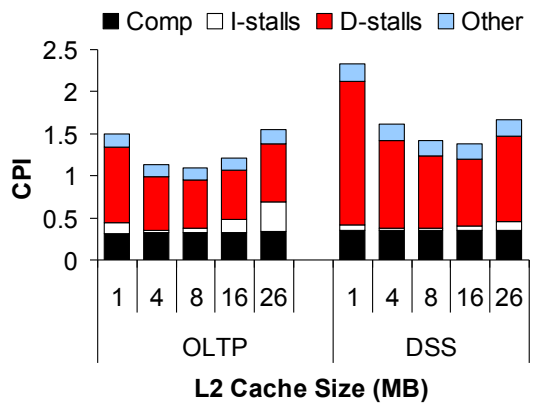
Performance of lean/fat cores



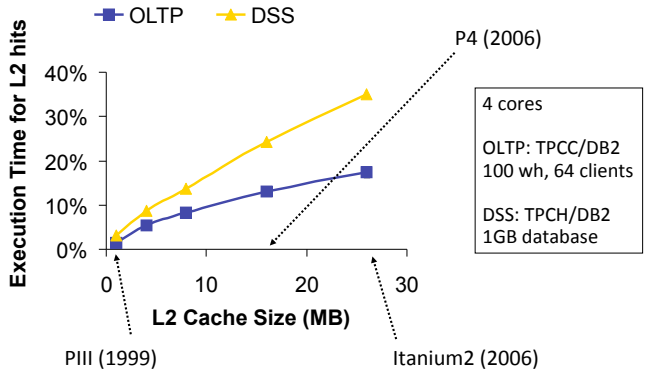
Impact of slower caches on perf.



Impact of slower caches on perf.

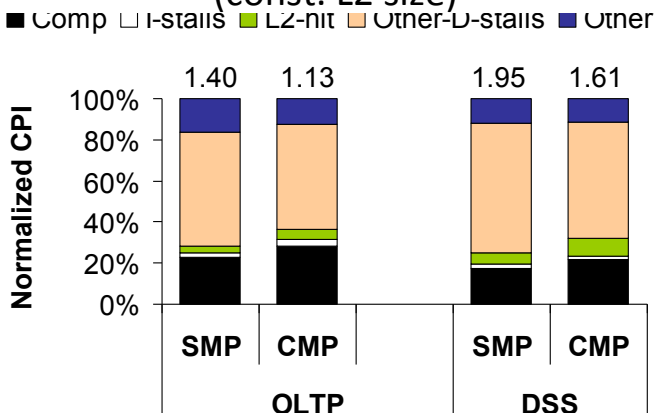


Impact of slower caches on execution



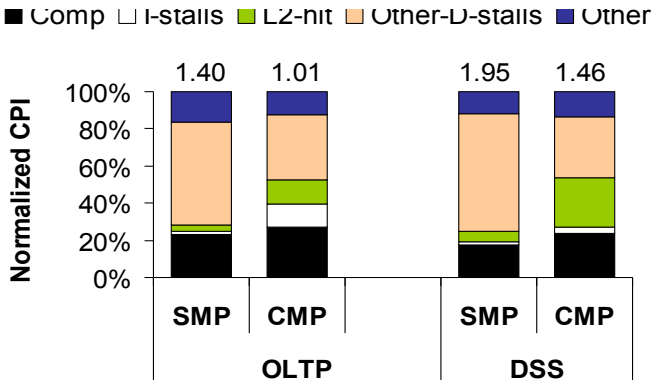
- Even in today's technology, 18-35% of time on cache

Impact of Core Integration (const. L2 size)



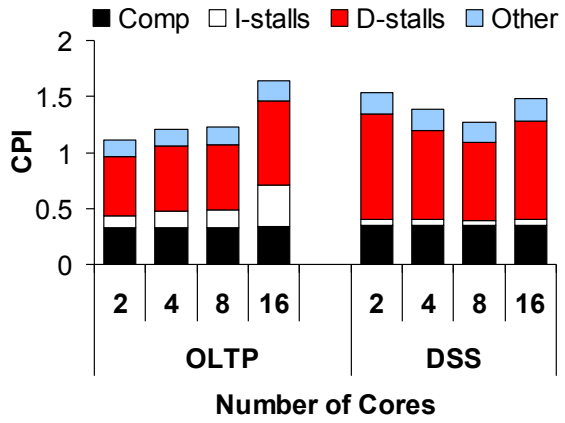
- Core integration alone does not affect L2-hits

Impact of Core Integration (const. aggregate L2 size)

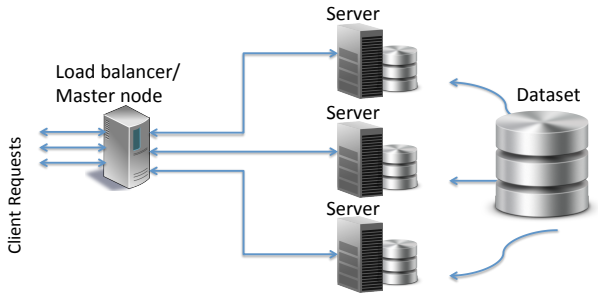


- 7x increase in L2-hit stalls

Impact of core count

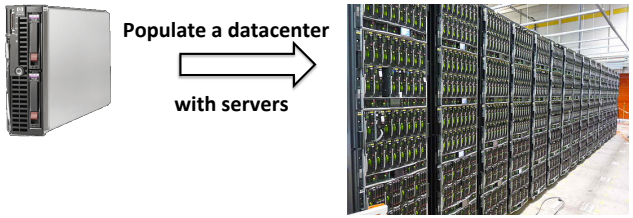


Characteristics of Scale-Out Apps



- Many independent requests/tasks
- Huge dataset split into shards
- Minimal communication among servers

Scale-Out Datacenter Internals



- Tens of thousands of servers
- Large memory per server
- High efficiency = high utilization

Maximize parallelism for efficiency

How Efficient are Today's Servers?

["Clearing the Clouds", ASPLOS '12]

- Created benchmark suite
  - Diverse set of cloud workloads
  - Quantified high-level behavior

- Studied off-the-shelf hardware
  - Used performance counters
  - Identified needs of cloud apps

Modern CPUs don't match needs of cloud apps

Cloud Suite 1.0

(released @ parsa.epfl.ch/cloudsuite)

Linux 2.6.32

Data Serving
Cassandra NoSQL

MapReduce
Machine learning on Hadoop

Media Streaming
Apple Quicktime Server

SAT Solver
Symbolic VM constraint solver

Web Frontend
Nginx, PHP server

Web Search
Apache Nutch

Hardware

Dell PowerEdge M1000e

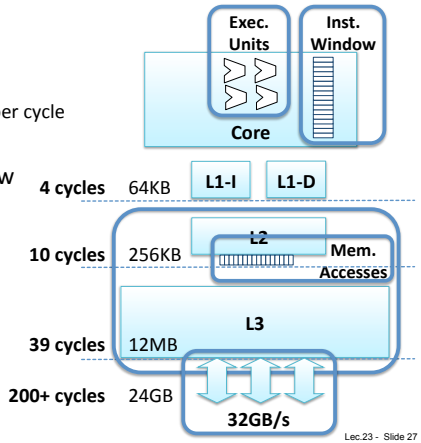
Dell Blades M610

Two Intel x5670
2.9GHz
6-core, 12MB LLC

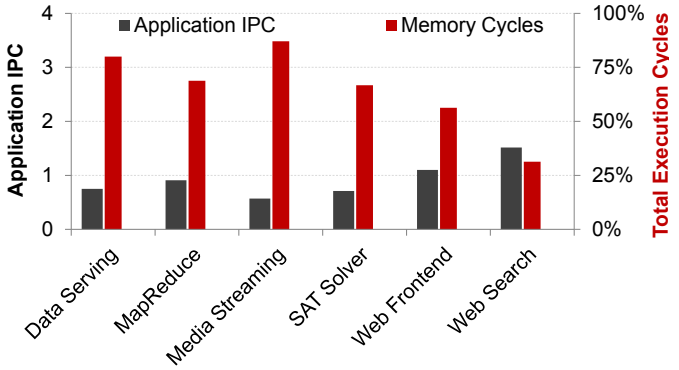
24GB RAM

Methodology: "Server-grade" CPU

- Aggressive OoO cores
  - Run up to 4 instructions per cycle
- Large instruction window
  - 128 instructions in flight
  - 48 loads in flight
- L2 and large L3 caches
- Vast off-chip b/w



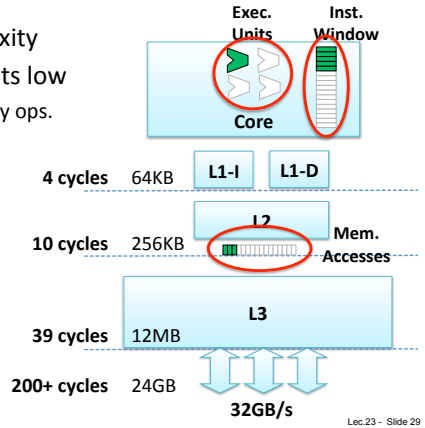
Core Inefficiencies



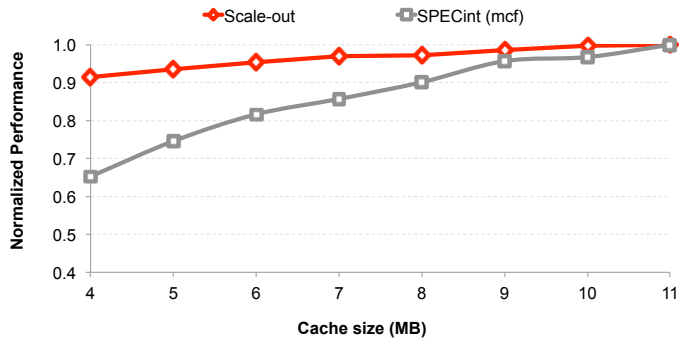
Execute ~1 instruction per cycle

Core Inefficiencies

- Underutilized complexity
- Scale-out requirements low
  - couple parallel memory ops.
  - one execution unit



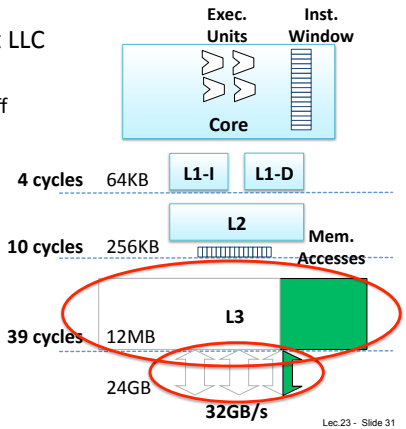
LLC Sensitivity



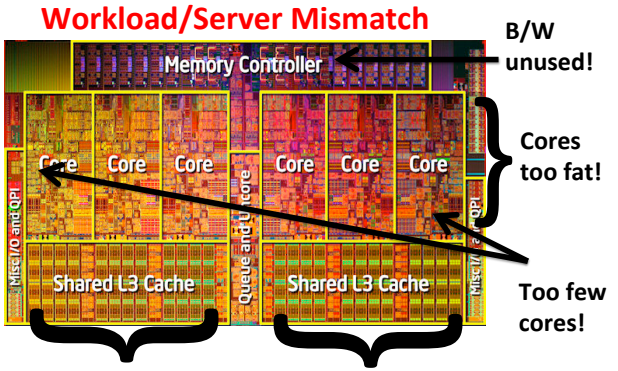
Minimal performance from large LLC

LLC and Bandwidth Inefficiencies

- Scale-out needs modest LLC
  - Beyond 3-4MB useless
  - Area & latency w/o payoff
- Low per-core BW needs
  - <15% utilization
  - Too many channels
  - Too high frequency



Clearing the Clouds in a nutshell [ASPLOS 2012]



8 MB (60%) waste of space (no reuse)!