# Reevaluating Amdahl's Law in the Multicore Era

**Xian-He Sun**

Illinois Institute of Technology
Chicago, Illinois
sun@iit.edu

# TOP 10 Machines (6/2004)

| Rank | Site | Computer | #proc | TF/s | Country |
|---|---|---|---|---|---|
| 1 | Earth simulator center | Earth simulator/NEC | 5120 | 35860 | Japan |
| 2 | LLNL | Thunder/Intel Itanium 2 Tiger 41.4GHz Quadrics | 4096 | 19940 | USA |
| 3 | LANL | ASCI Q/AlphaServer SC45 1.25GHz | 8192 | 13880 | USA |
| 4 | IBM-Rochester | BlueGene/LDD1 Prototype | 8192 | 11680 | USA |
| 5 | NCAS | Tungsten/PowerEdge 1750, P4 Xeon 3.06Ghz | 2500 | 9819 | USA |
| 6 | ECMWF | eServer P Series690 IBM | 2112 | 8955 | UK |
| 7 | RIKEN | RIKEN Super Combined Cluster/Fujitsu | 2048 | 8728 | Japan |
| 8 | IBM-Thomas Watson | BlueGene/LDD2 Prototype | 2096 | 8655 | USA |
| 9 | PNNL | Integrity rx2600 Itanium 21.5 GHz | 1936 | 8633 | USA |
| 10 | Shanghai Supercomputer Center | Dawning 4000A Opteron 2.2GHz | 2560 | 8061 | China |

**http://www.top500.org/list/2004/06/**

# Cloud: Integrated Resource

Provide virtual computing environments on demand

# Scalable Computing: the Way to High-performance

IBM BG/P

Source: ANL ALCF

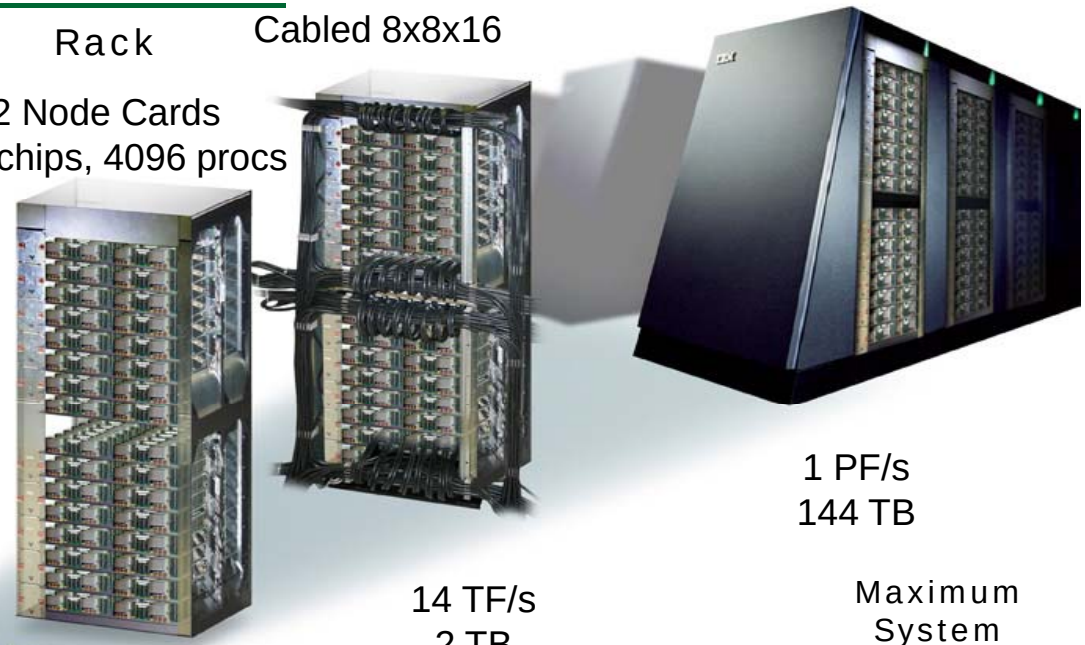
Petaflops System

72 Racks

Cabled 8x8x16

Rack

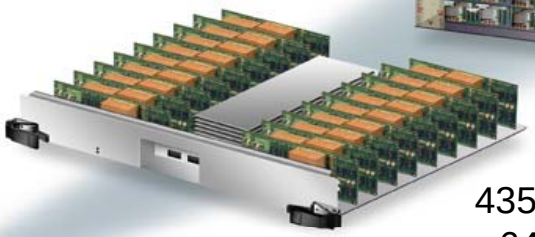
32 Node Cards
1024 chips, 4096 procs



1 PF/s
144 TB

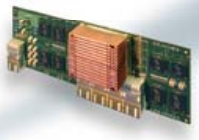
14 TF/s
2 TB

Node Board
(32 chips 4x4x2)
32 compute, 0-2 IO cards



435 GF/s
64 GB

Compute Card
1 chip, 20 DRAMs



13.6 GF/s
2.0 GB DDR
Supports 4-way SMP

Chip
4 cores

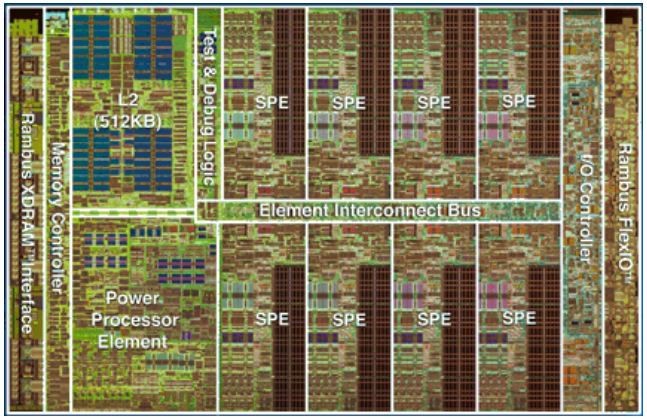
850 MHz
8 MB EDRAM

Front End Node / Service Node
System p Servers
Linux SLES10

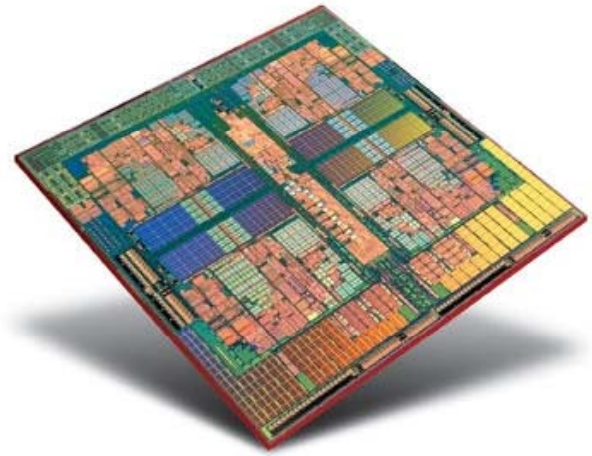
Maximum System
256 racks
3.5 PF/s
512 TB

HPC SW:
Compilers
GPFS
ESSL
Loadleveler

# Multicore Adds in Another Dimension

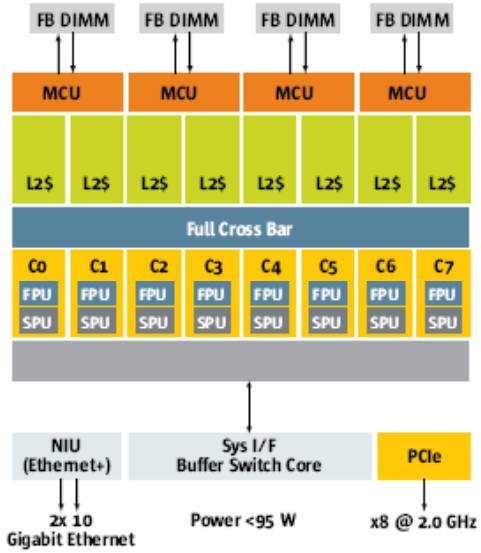


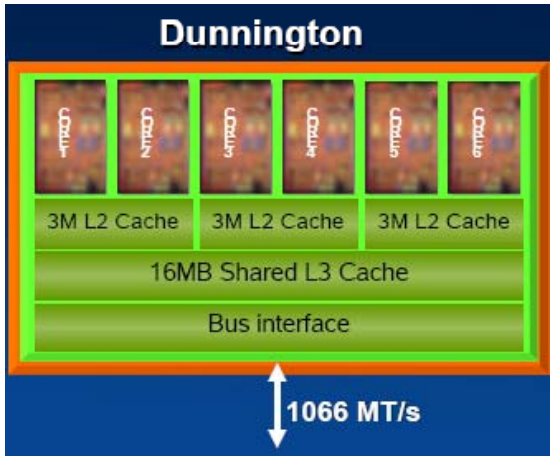

IBM Cell: 8 slave cores
+ 1 master core, 2005



AMD Phenom:
4 cores, 2007

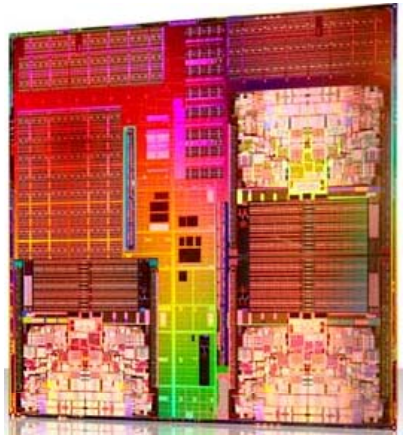
Sun T2: 8
cores, 2007



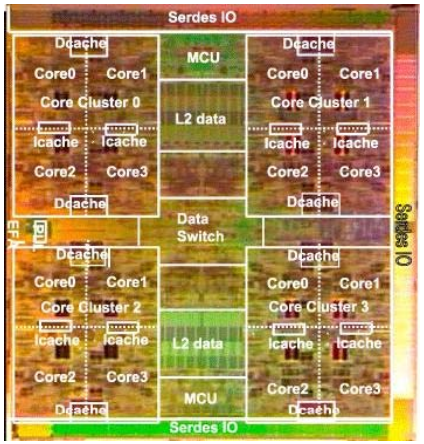




Intel Dunnington: 6 cores, 2008

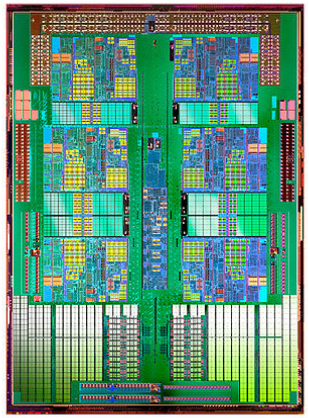
# No in the Mood to Scale Up, yet



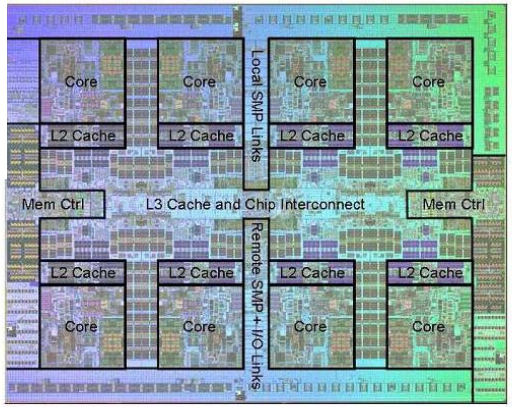
Intel Dunnington:
6 cores, 2009




Sun UltraSPARC Rock:
16 cores, 2009



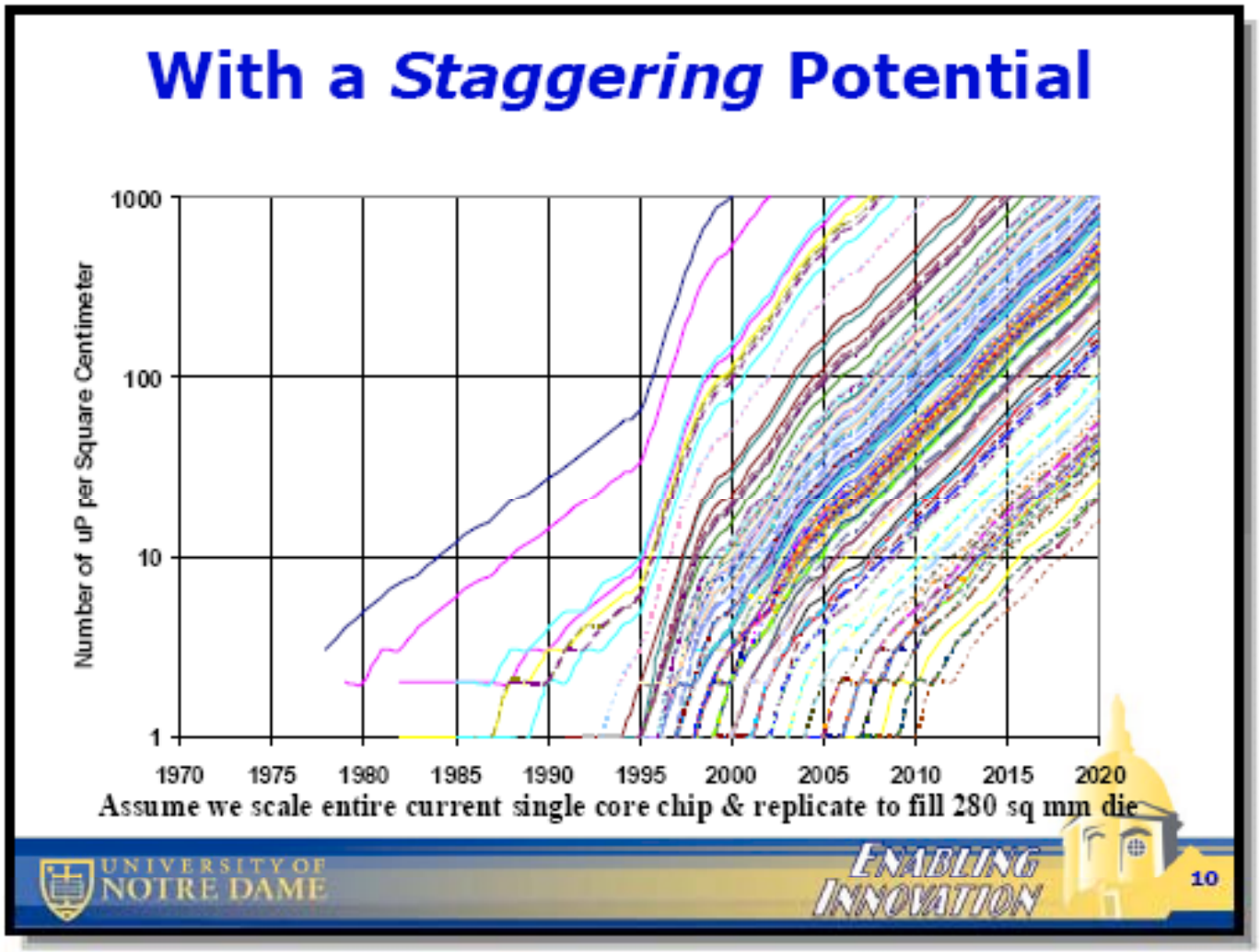
AMD Opteron "*Istanbul*":
6 Cores, 2009




IBM Power-7: 8 cores, 2010

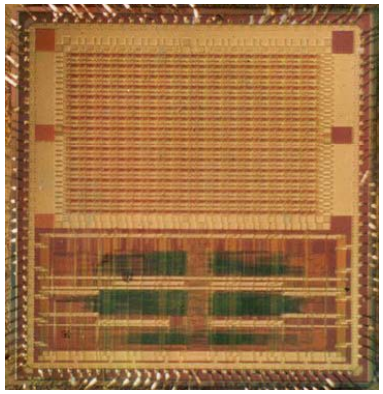
# Why not Scale up the Number of Cores?

**Perception / technology?**



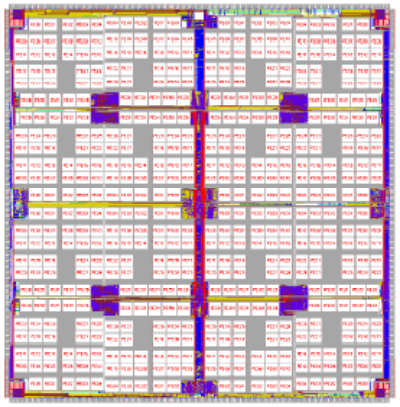

Peter Kogge, 2007

# Whereas Technology is Available

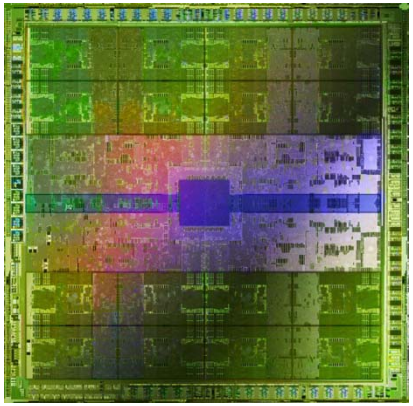


Kilocore: 256-core prototype
By Rapport Inc.

Tesla C1060:
240 cores, by NVDIA



GRAPE-DR chip:
512-core, By Japan



**NVIDIA Fermi:** 512 CUDA cores

Quadro FX 5800: 240 cores,
By NVDIA.

GRAPE-DR testboard

# It All Starts with Amdahl's Law

- Gene M. Amdahl, "*Validity of the Single-Processor Approach to Achieving Large Scale Computing Capabilities*", 1967
- Amdahl's law (Amdahl's speedup model)

$$Speedup_{Amdahl} = \frac{1}{(1-f) + \frac{f}{n}}$$

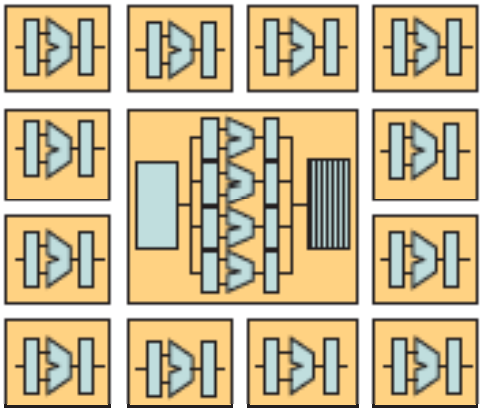
$$\lim_{n \to \infty} Speedup_{Amdahl} = \frac{1}{1-f}$$

- $f$ is the parallel portion
- Implications

# Amdahl's Law for Multicore (Hill&Marty)

- Hill & Marty, "*Amdahl's Law in the Multicore Era*", *IEEE Computer*, July 2008
- Study the limitation of multicore architecture based on Amdahl's law for parallel processing and hardware concern
  - *n* BCEs (Base Core Equivalents)
  - A powerful *perf(r)* core built with *r* BCEs is best choice from design concern

Symmetric



Asymmetric

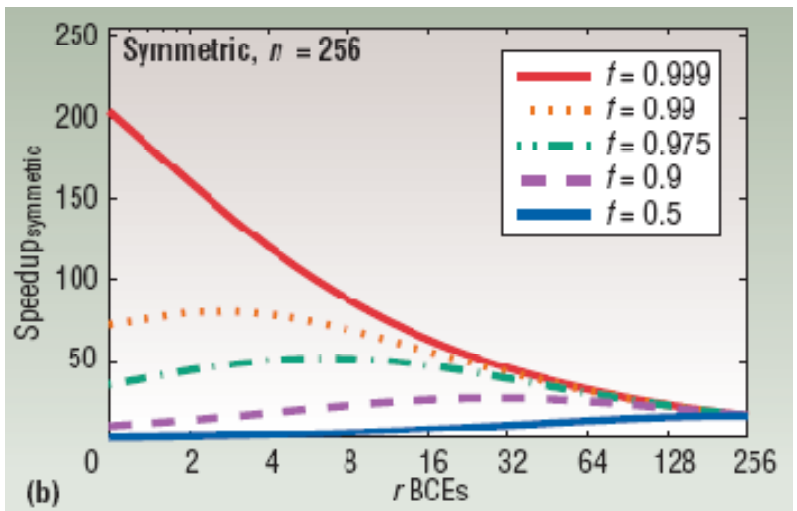
# Amdahl's Law for Multicore (Hill&Marty)

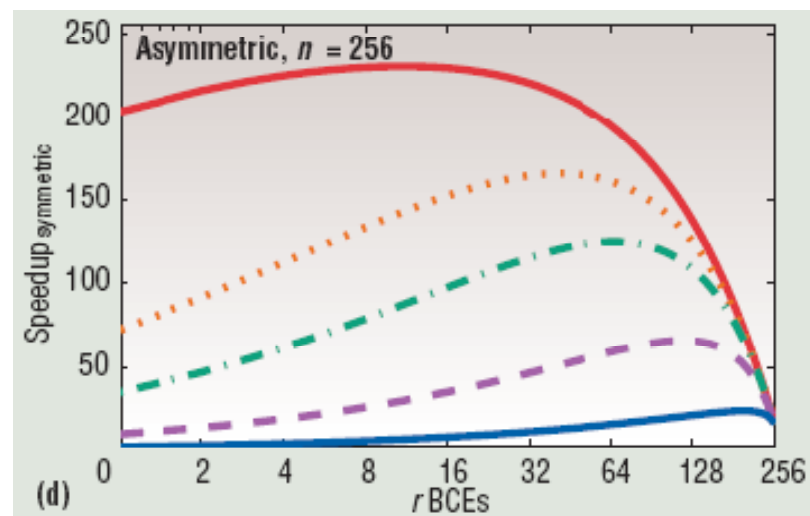
- Speedup of symmetric architecture

$$Speedup_{symmetric}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f \cdot r}{perf(r) \cdot n}}$$

- Speedup of asymmetric architecture

$$Speedup_{asymmetric}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{perf(r) + n - r}}$$



# History Repeats Itself (back to 1988)?

IBM 7030 Stretch

IBM 7950 Harvest

All have up to 8 processors, citing Amdahl's law,

$$\lim_{n\to\infty} Speedup_{Amdahl} = \frac{1}{1-f}$$

Cray X-MP
Fastest computer 1983-1985

Cray Y-MP

# Terms of Scalable Computing (today)

LANL Roadrunner:
18,360 processors, 130,464 cores
2009 World's fastest supercomputer

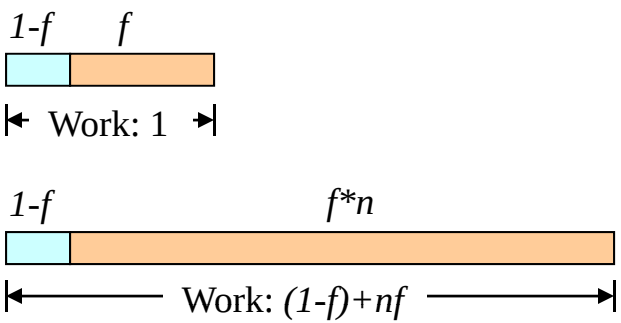
TACC Ranger:
15,744 processors,
2008

The scale size is far beyond implication of Amdahl's law

ANL Intrepid:
20,480 processors, 2008

# Scalable Computing

- Tacit assumption in Amdahl's law
  - The problem size is fixed
  - The speedup emphasizes time reduction
- Gustafson's Law, 1988
  - Fixed-time speedup model




$$Speedup_{fixed-time} = \frac{Sequential\ Time\ of\ Solving\ Scaled\ Workload}{Parallel\ Time\ of\ Solving\ Scaled\ Workload} = (1-f)+nf$$

- Sun and Ni's law, 1990
  - Memory-bounded speedup model

$$Speedup_{memory-bounded} = \frac{Sequential\ Time\ of\ Solving\ Scaled\ Workload}{Parallel\ Time\ of\ Solving\ Scaled\ Workload} = \frac{(1-f)+fG(n)}{(1-f)+fG(n)/n}$$

# Revisit Amdahl's Law for Multicore

$$Speedup_{fixed-size} = \frac{T_{original}}{T_{enhanced}}$$

$$T_{original} = \frac{w}{perf(1)} = w$$

$$T_{enhanced} = \frac{(1-f)w}{perf(r)} + \frac{fw}{\frac{n}{r} \cdot perf(r)}$$

$$Speedup_{fixed-size} = \frac{\frac{w}{perf(1)}}{\frac{(1-f)w}{perf(r)} + \frac{fwr}{n \cdot perf(r)}}$$

$$= \frac{1}{\frac{1-f}{perf(r)} + \frac{f \cdot r}{perf(r) \cdot n}}$$

Hill and Marty's findings

# Fixed-time Model for Multicore

- Emphasis on work finished in a fixed time
- Problem size is scaled from *w* to *w'*
- *w'*: Work finished within the fixed time, when the number of cores scales from *r* to *mr*

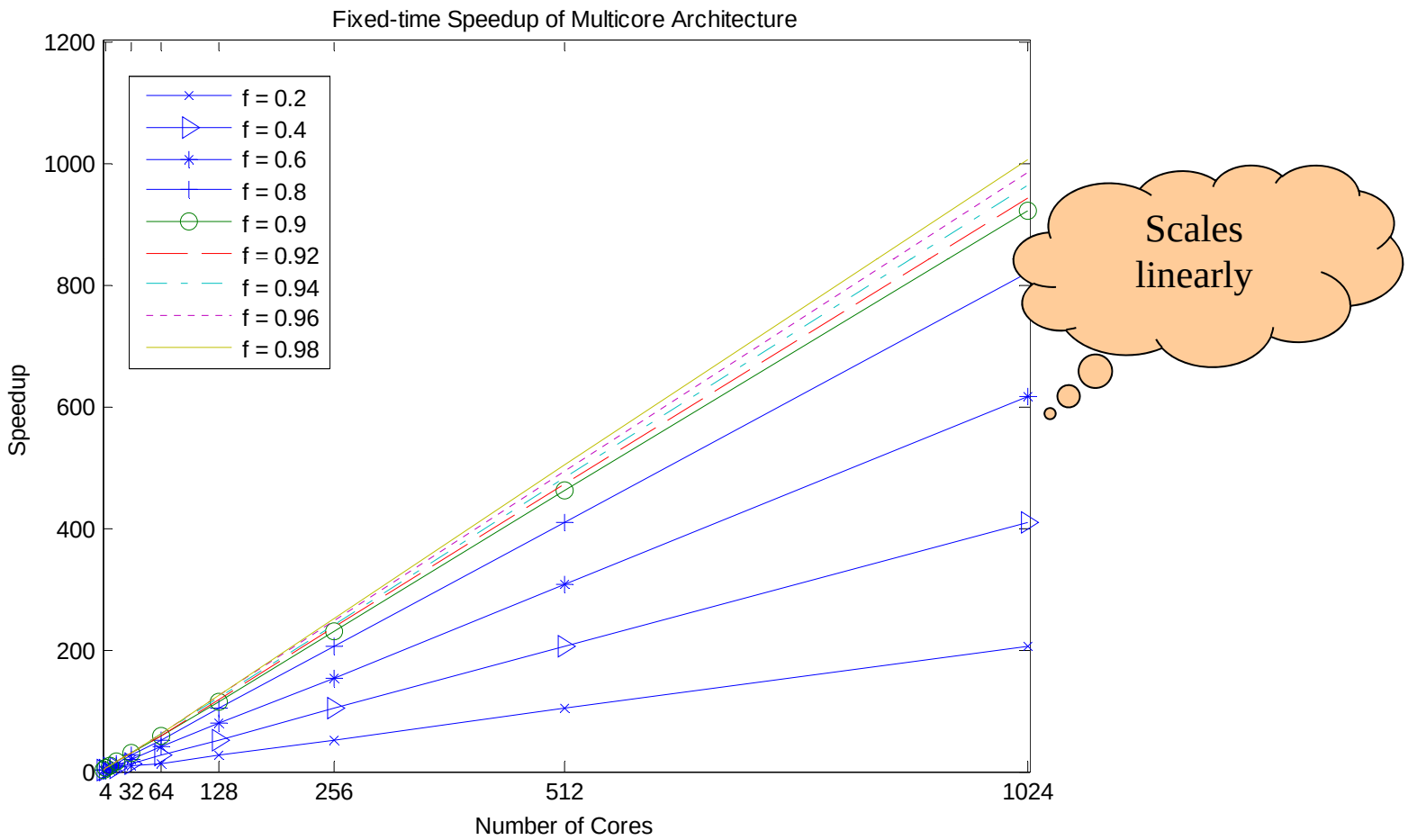
$$\frac{(1-f)w}{perf(r)}+\frac{fw}{perf(r)}=\frac{(1-f)w}{perf(r)}+\frac{fw'}{perf(r)m} \Rightarrow \quad w'=mw$$

- The scaled fixed-time speedup

$$Speedup_{fixed-time}=\frac{Time\ of\ Solving\ w'\ in\ Original\ Mode}{Time\ of\ Solving\ w\ in\ Original\ Mode}$$

$$=\frac{\frac{(1-f)w}{perf(r)}+\frac{fw'}{perf(r)}}{\frac{w}{perf(r)}}=(1-f)+mf$$

# Fixed-time Speedup for Multicore



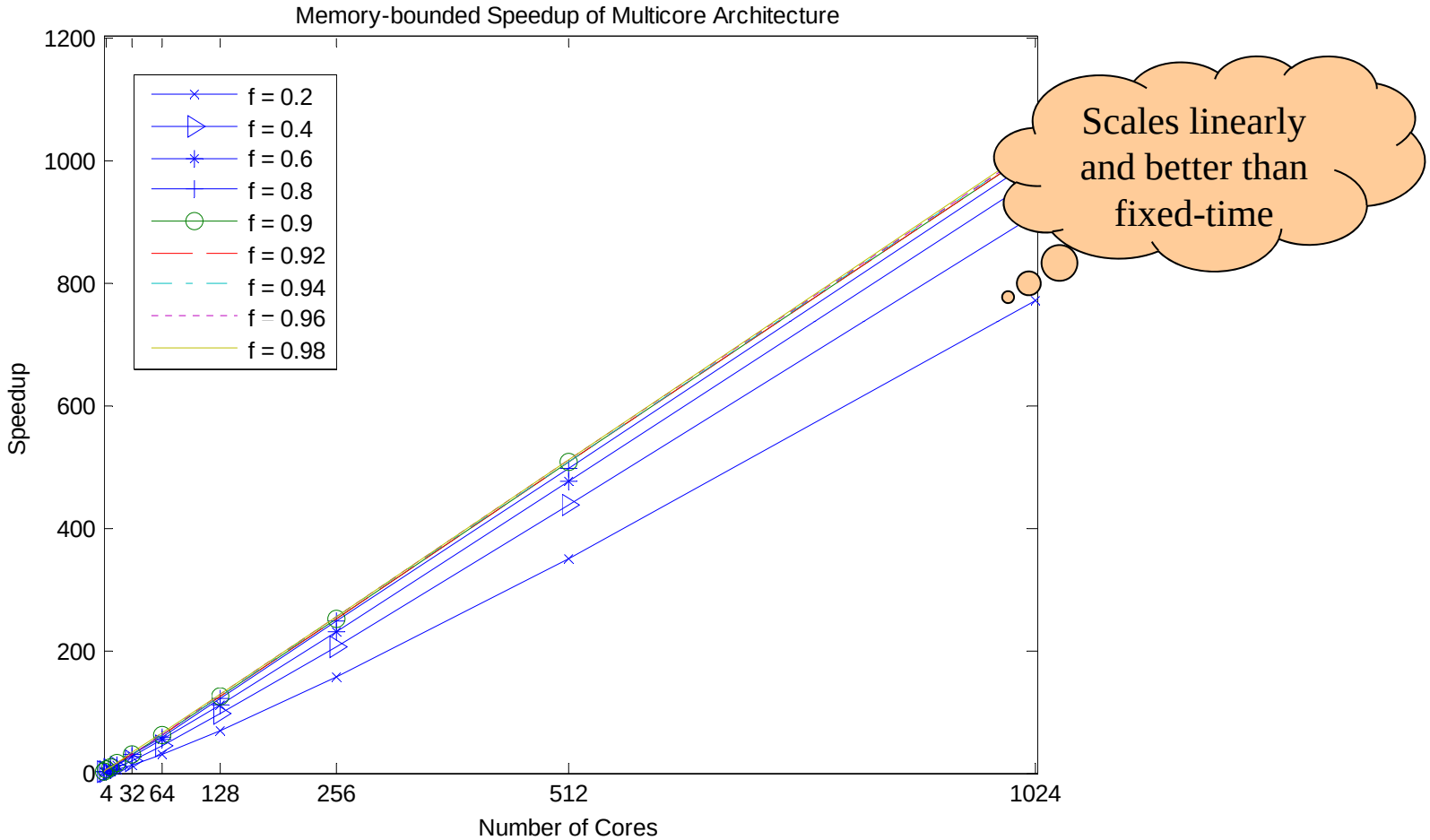
# Memory-bounded Model for Multicore

- Problem size is scaled from *w* to *w**
- *w**: Work executed under memory limitation (each core has its own L1 cache)
- $w^* = g(m)w$, where $g(m)$ is the increased workload as the memory capacity increases *m* times ($g(m) = 0.38m^{3/2}$, for matrix-multiplication $2N^3$ v.s. $3N^2$)
- The scaled memory-bounded speedup

$$Speedup_{memory-bounded} = \frac{Time\ of\ Solving\ w^*\ in\ Original\ Mode}{Time\ of\ Solving\ w\ in\ Original\ Mode}$$

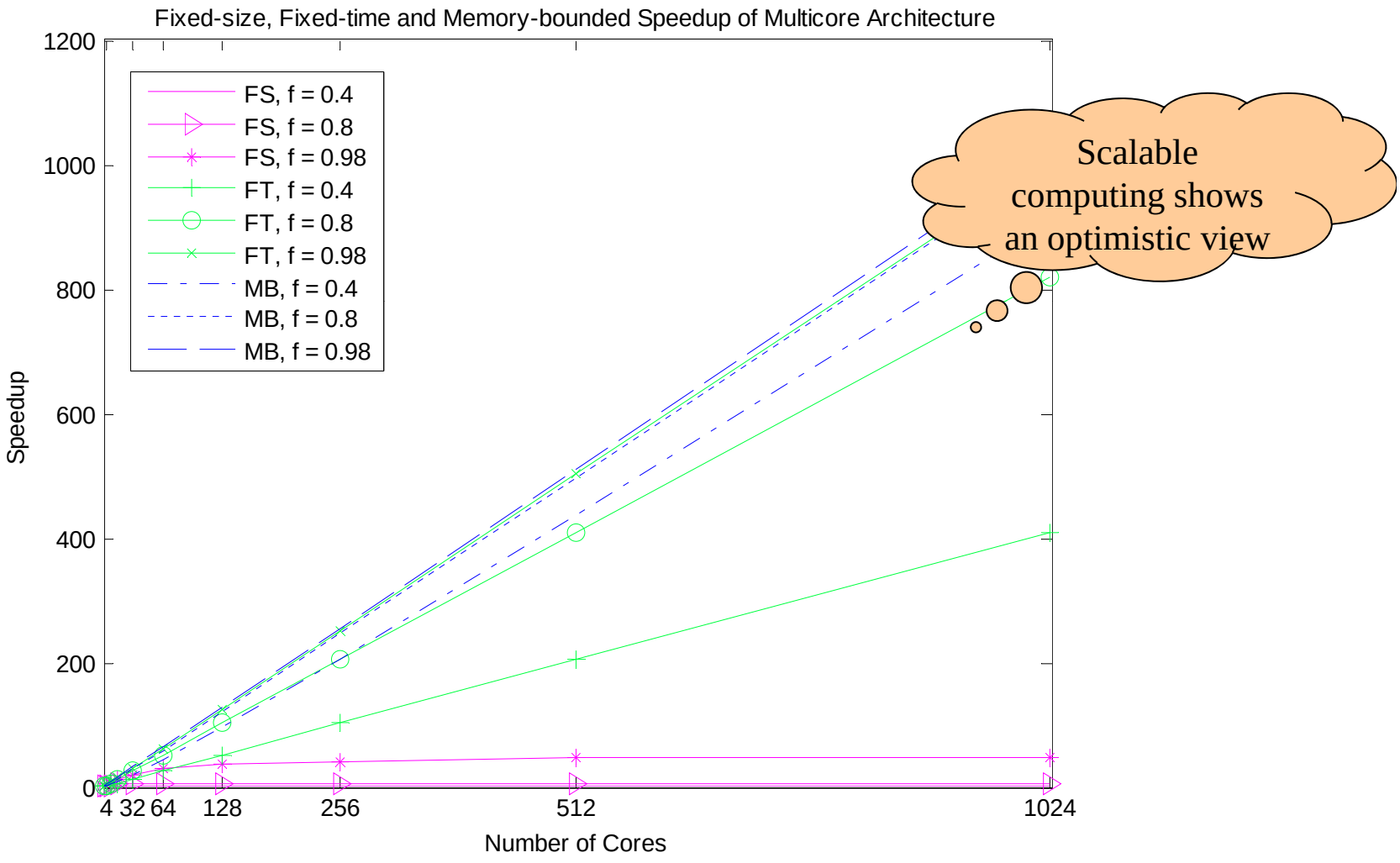
$$= \frac{(1-f)+g(m)f}{(1-f)+\frac{g(m)f}{m}}$$

# Memory-bounded Speedup for Multicore




$g(m) = 0.38m^{3/2}$ for matrix-multiplication *2N3* v.s. *3N2*

# Perspective: a comparison



# Result and Implications

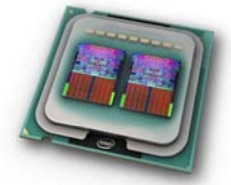
- Result : The scalable computing concept and the two scaled speedup models are applicable to multicore architecture
- Implication 1: Amdahl's law (Hill&Marty) presents a limited and pessimistic view
- Implication 2: Multicore is scalable in term of the number of cores
- Implication 3: The memory-bounded model reveals the relation between scalability and memory capacity and requirement

**Question:**

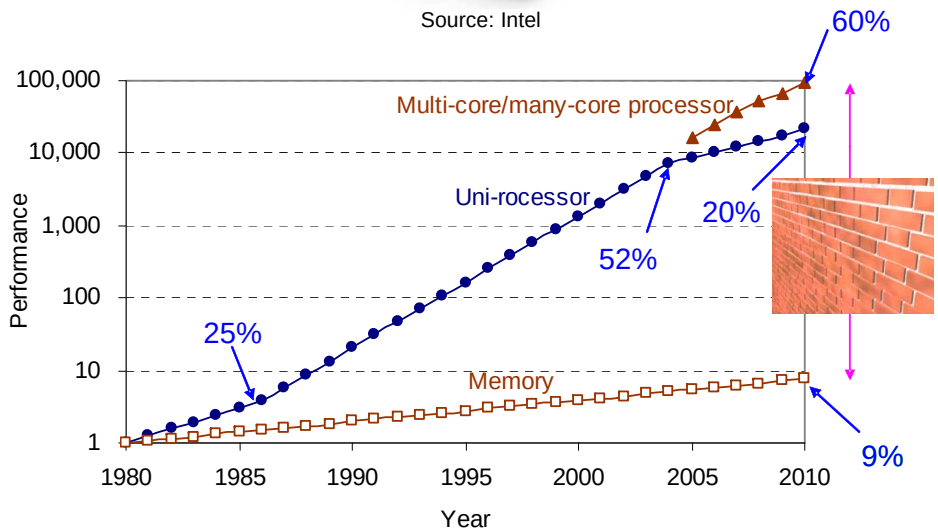
Is data access the actual performance constraint of multicore architecture?

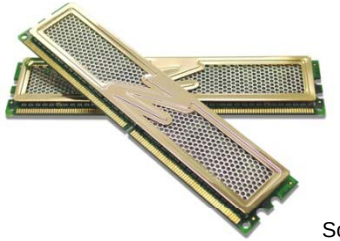
# Processor-memory performance gap

- Processor performance increases rapidly
  - Uni-processor: ~52% until 2004, ~25% since then
  - New trend: multi-core/many-core architecture
    - Intel TeraFlops chip, 2007
  - Aggregate processor performance much higher
- Memory: ~9% per year
- Processor-memory speed gap keeps increasing



Source: Intel

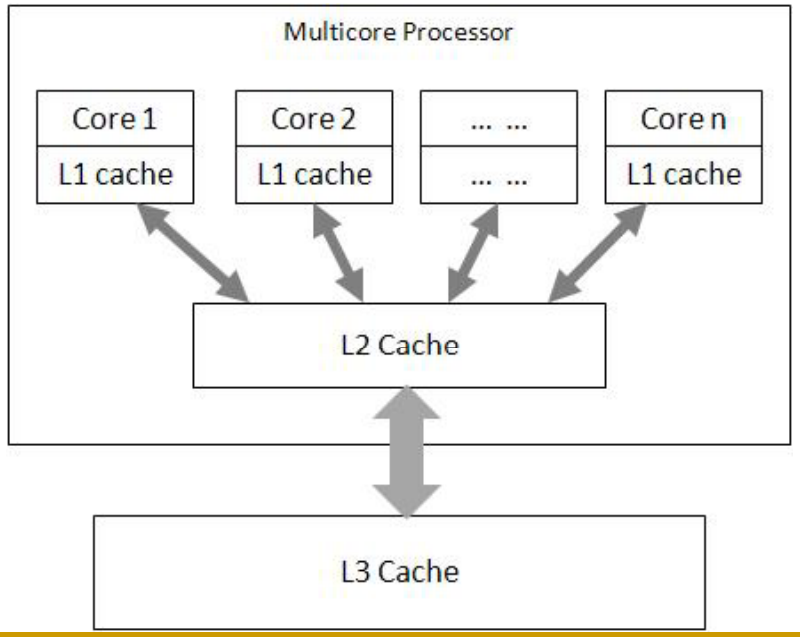


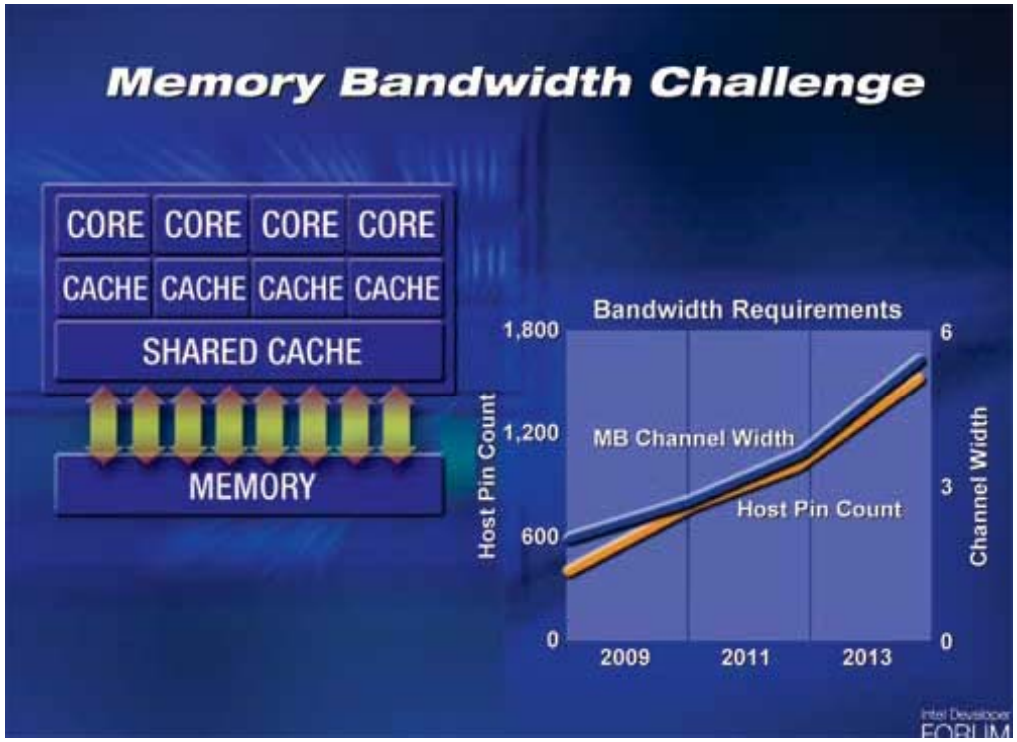



Source: OCZ

# Multicore Scalability Analysis

- Architecture
  - *N* cores
  - Data contention to L2
  - Increase cores does not improve data access speed



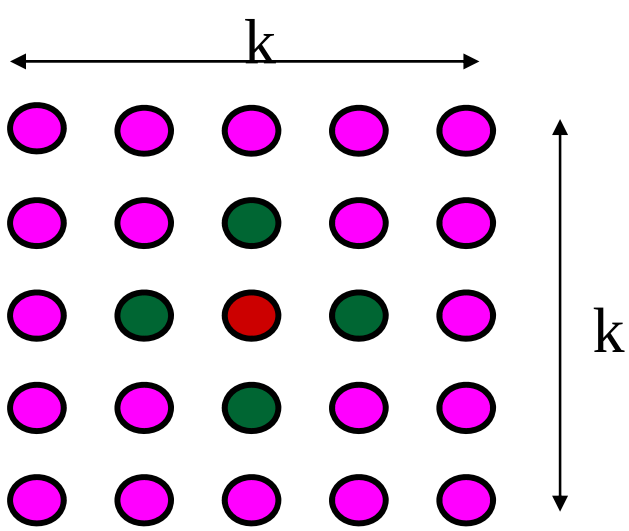



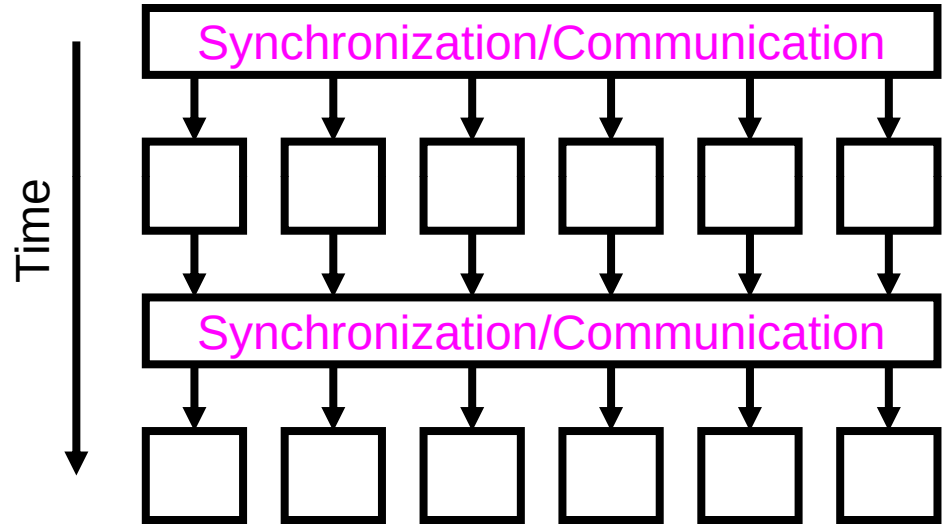
# Application: Iterative Solvers

- Two phases:
  - Computing phase and communication phase

Dense Solver

$k^3$ comp, $k^2$ memory



# Data Access as the Scalability Constraint

- Phased computing model (embarrassing parallel, meta-tasks)
- Assume a task has two parts, $w = w_p + w_c$
  - Data processing work, $w_p$
  - Data communication (access) work, $w_c$
- Fixed-size speedup with data-access processing consideration

$$Speedup = \frac{1}{\frac{w_c}{perf(r)} + \frac{w_p \cdot r}{perf(r) \cdot n}}$$

# Scaled Speedup under Memory-wall

- Assuming data access time is fixed, Fixed-time model constraint

$$\frac{w_c}{perf(r)} + \frac{w_p}{perf(r)} = \frac{w_c}{perf(r)} + \frac{w_p'}{m \cdot perf(r)} \quad => \quad w_p' = m w_p$$

- Fixed-time speedup

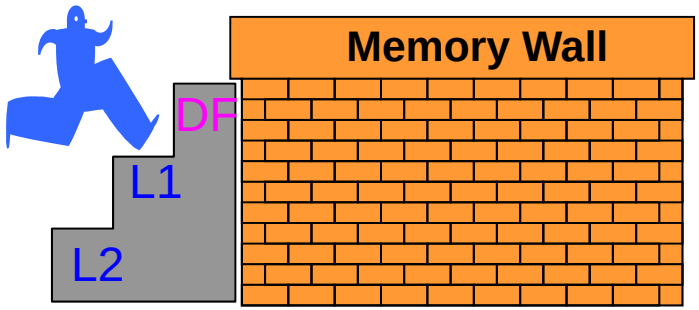
$$\frac{\frac{w_c}{perf(r)} + \frac{w_p'}{m \cdot perf(r)}}{\frac{w_c}{perf(r)} + \frac{w_p}{perf(r)}} = \frac{w_c + m \cdot w_p}{w_c + w_p} = (1 - f') + mf'$$

$$f' = \frac{w_p}{w_c + w_p}$$

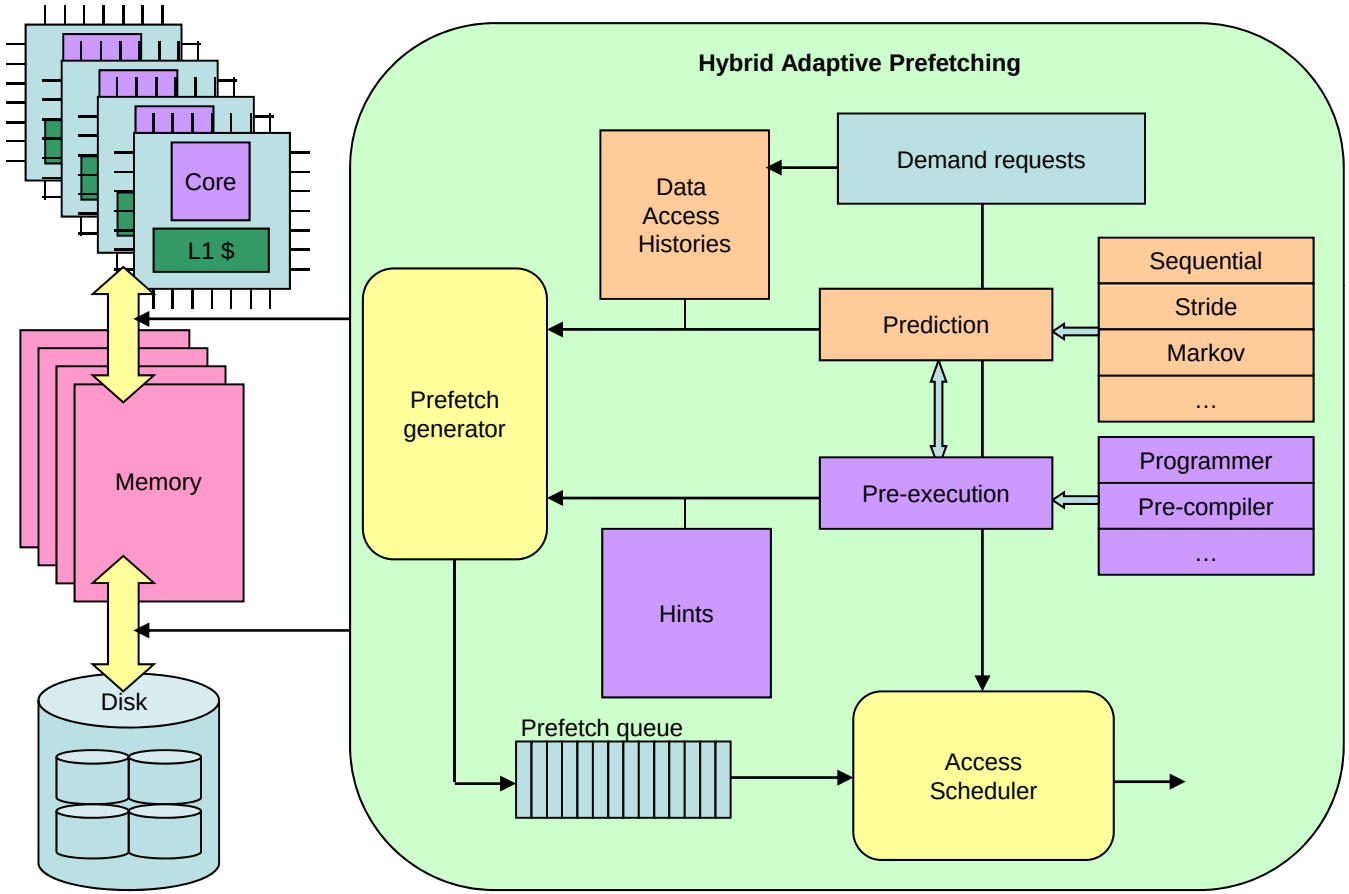
- Memory-bounded speedup
  - With $g(m) = 0.38m^{3/2}$ memory-bounded speedup is bigger than fixed-time speedup
  - *g(m)* equals one, memory-bounded is the as fixed-size, *g(m)* equals *m*, then memory-bound is the same as fixed-time

# Mitigating Memory-wall Effect

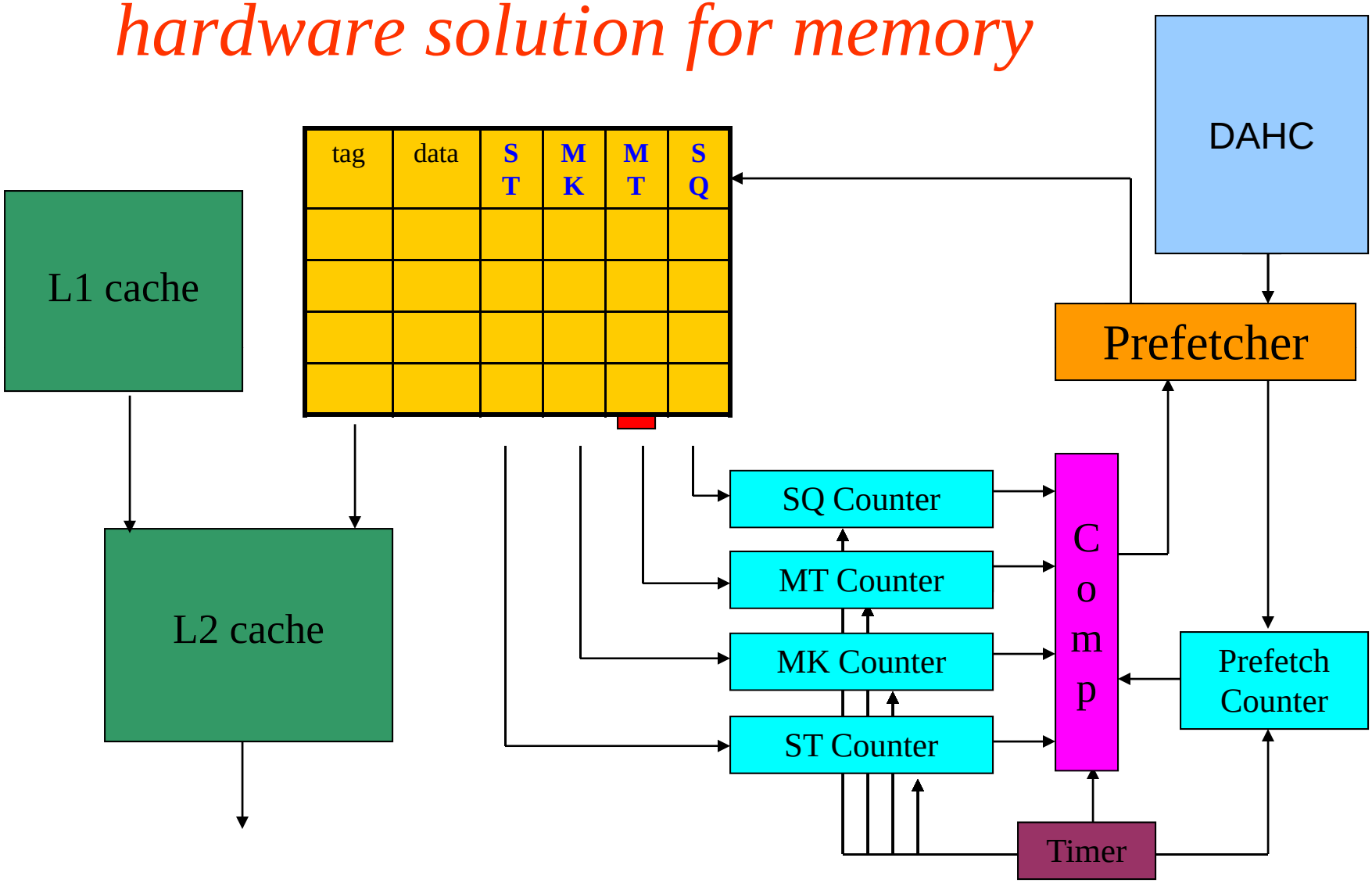
- Result: Multicore is scalable, but under the assumption
  - Data access time is fixed and does not increase with the amount of work and the number of cores
- **Implication:** Data access is the bottleneck needs attention
- **Data Prefetching**
  - Software prefetching technique
    - Adaptive, compete for computing power, and costly
  - Hardware prefetching technique
    - Fixed, simple, and less powerful
- **Our Solutions**
  - Data Access History Cache (DAHC)
  - Server-based Push Prefetching



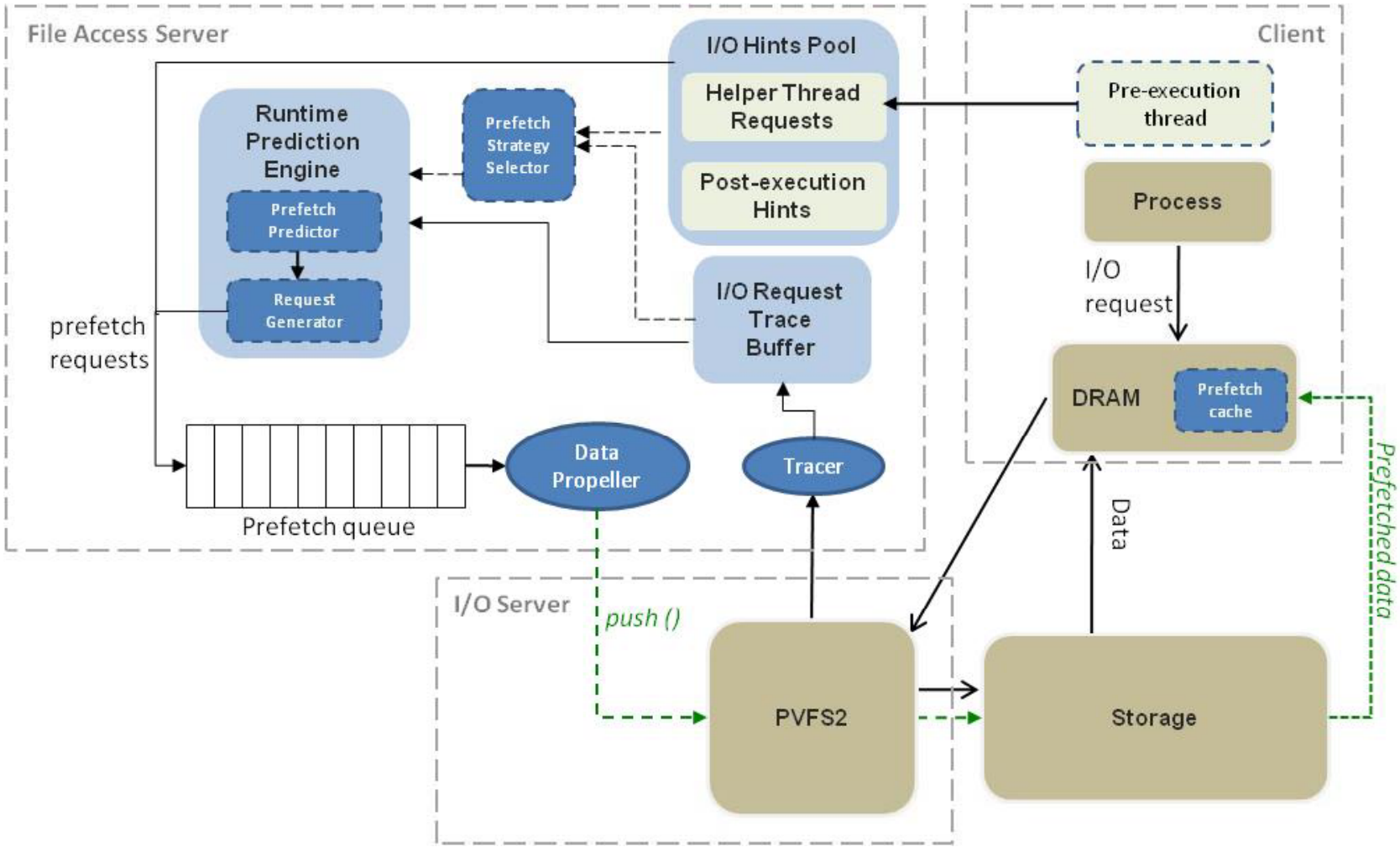
# Hybrid Adaptive Prefetching Architecture



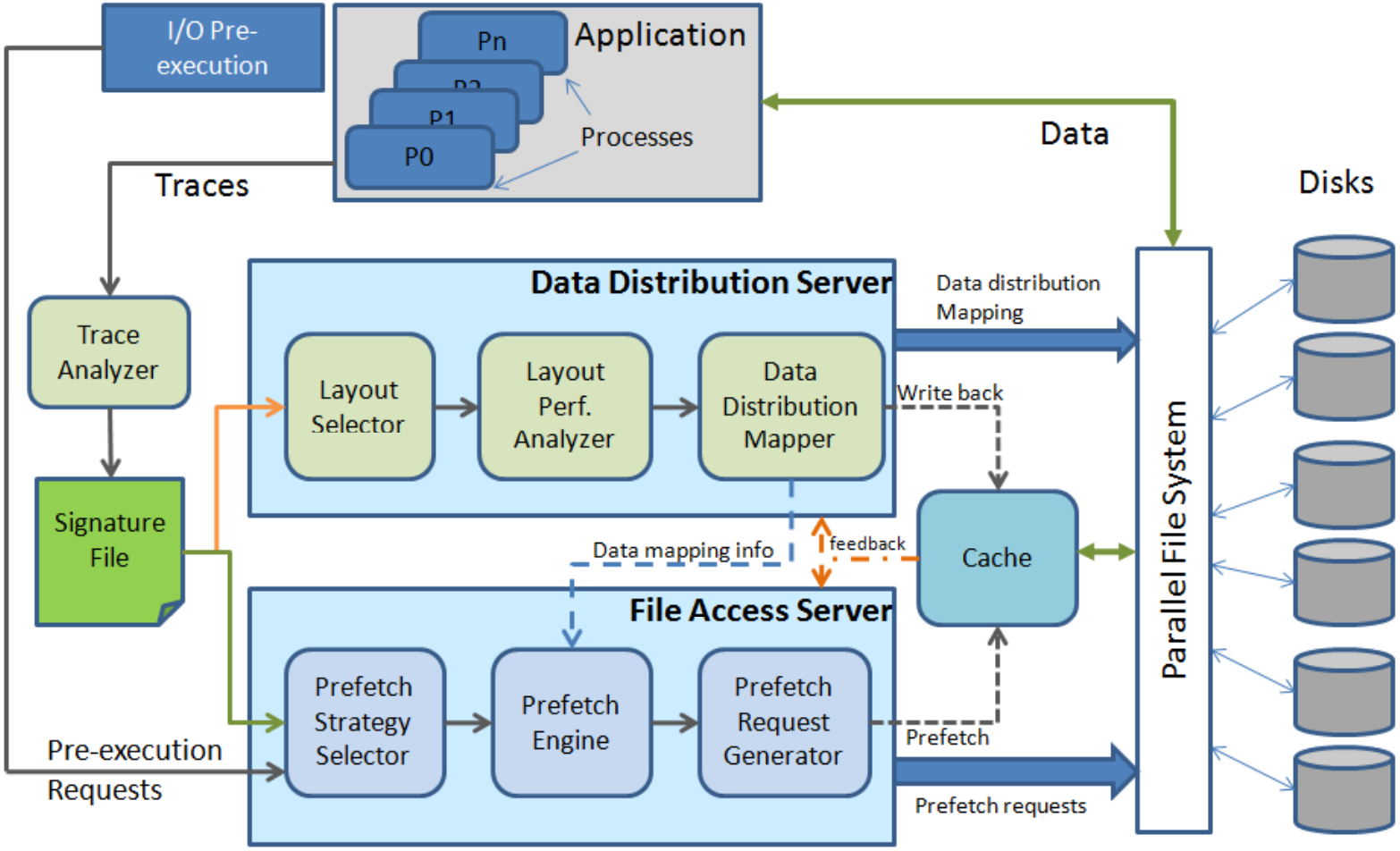
# Data Access History Cache: a *hardware solution for memory*



# Push-IO: A Software Solution for I/O



# Dynamic Application-specific I/O Optimization Architecture



# Conclusion

- Cloud computing and multicore/manycore architecture lead to the future of computing
- Multicore architecture is scalable
- Scaling up the number of cores can continually improve performance, if the data access delay is fixed
- Data access is the killing factor of performance
- Mitigating memory-wall: Data prefetching
  - Data Access History Cache (DAHC)
  - Server-based Push Prefetching
- Mitigating memory-wall: **Application-specific data access system**

# Thank you!

To visit http://www.cs.iit.edu/~scs