# Meet the Walkers

## Accelerating Index Traversals for In-Memory Databases

Babak Falsafi
Director, EcoCloud
ecocloud.ch

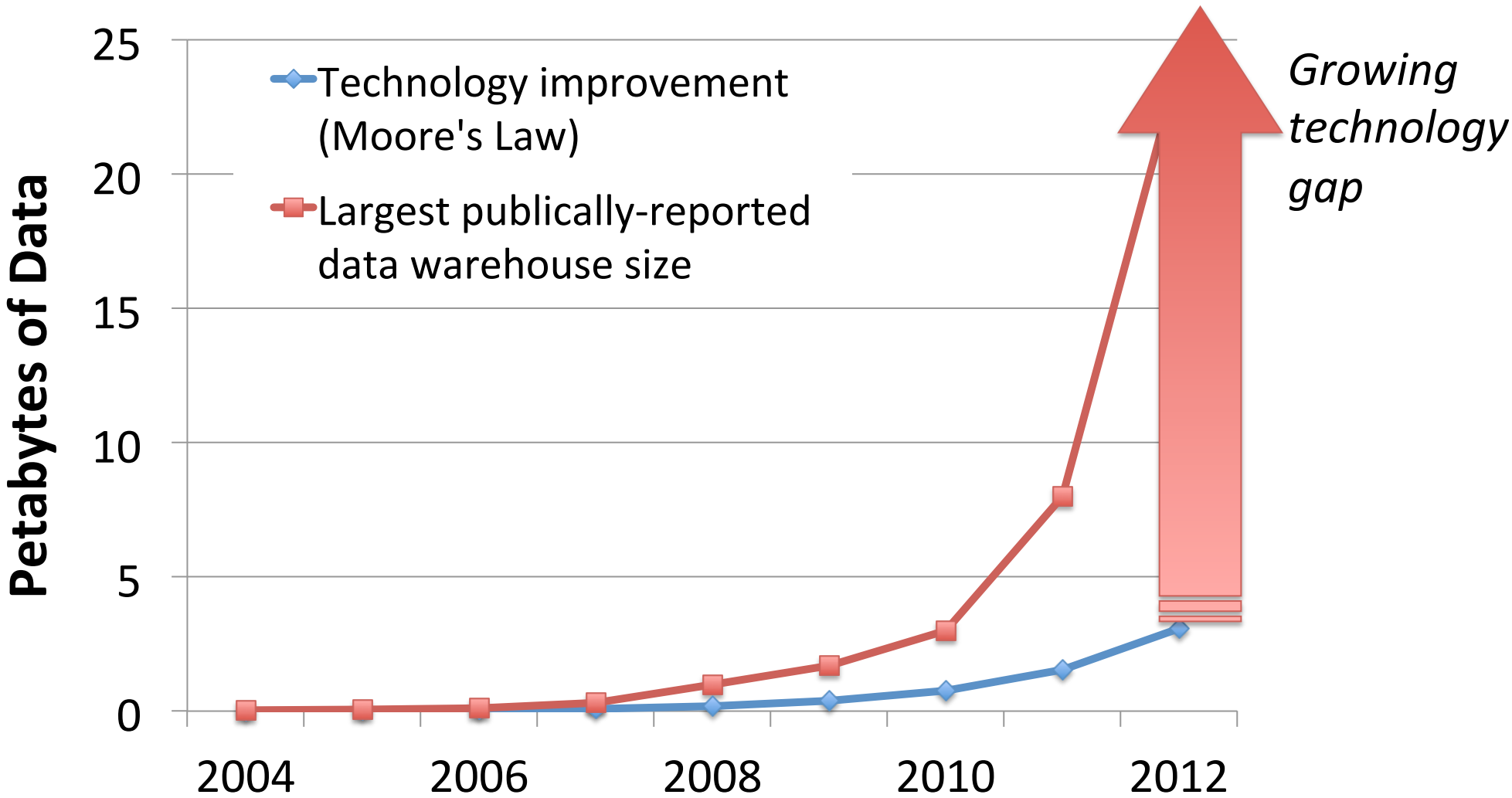
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

# Big Data is Here


[source: Economist]

- Data growth (by 2015) = 100x in ten years [IDC 2012]
  - Population growth = 10% in ten years
- Monetizing data for commerce, health, science, services, ….
- Big Data is shaping IT & pretty much whatever we do!

# Data Growing Faster than Technology




WinterCorp Survey, www.wintercorp.com

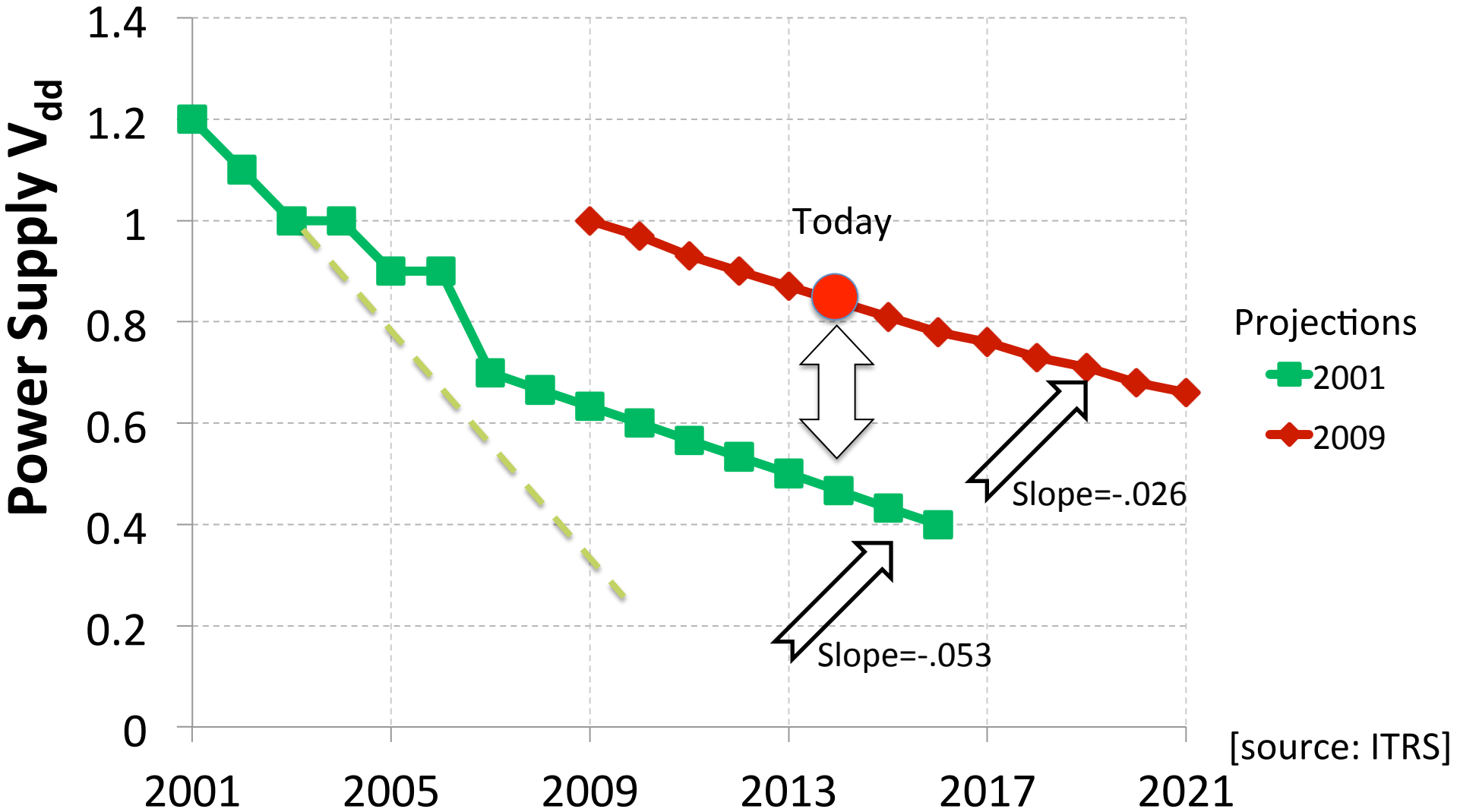
# Data-Centric IT Growing Fast

Source: James Hamilton, 2012


**Daily** IT growth in 2012 = IT first five years of business!

amazon web services

# But, Efficiency is No Longer Free




The fundamental energy silver bullet is gone!

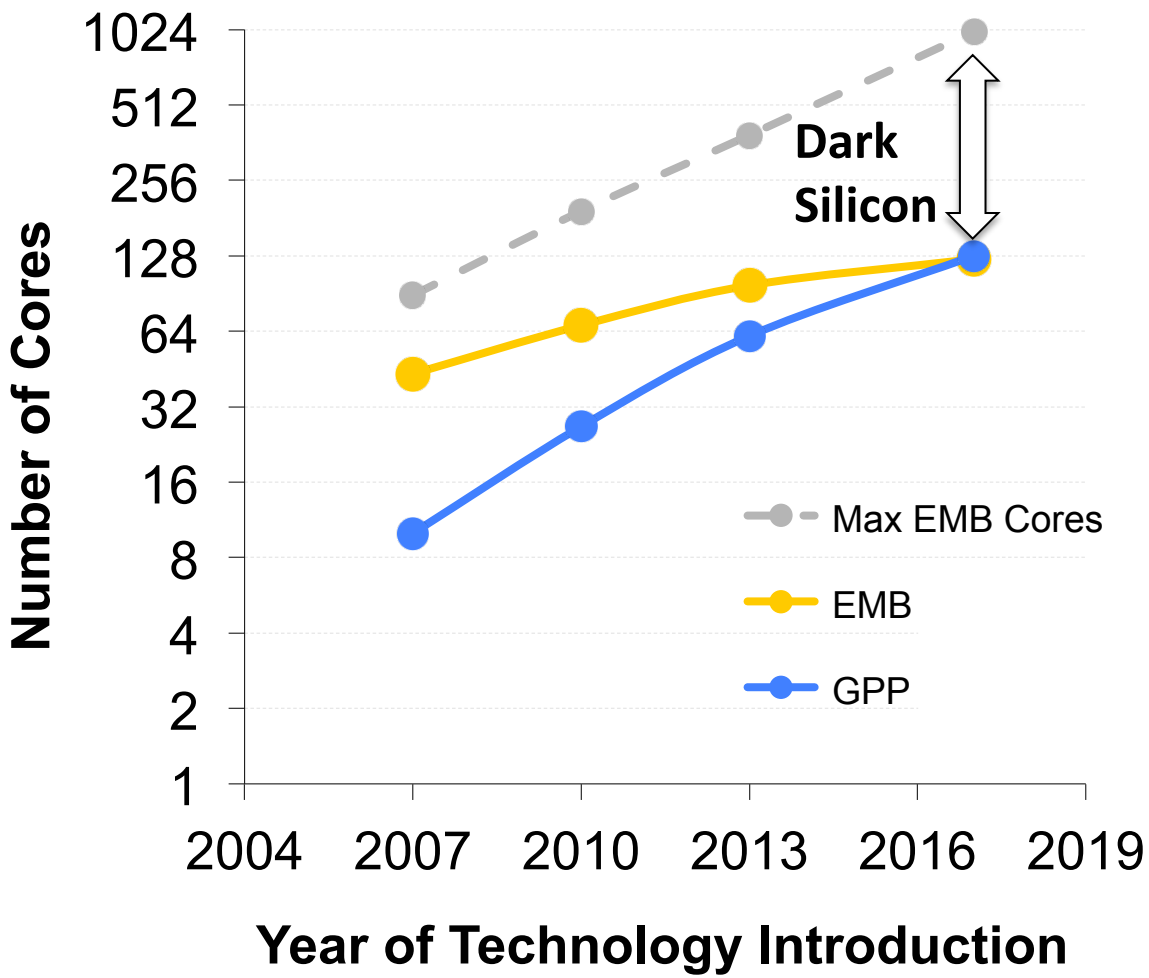
# Dark Silicon: End of Multicore/Manycore Scaling

Parallel computing is a popular solution for efficiency

But parallelism alone can not offset leveling voltages!

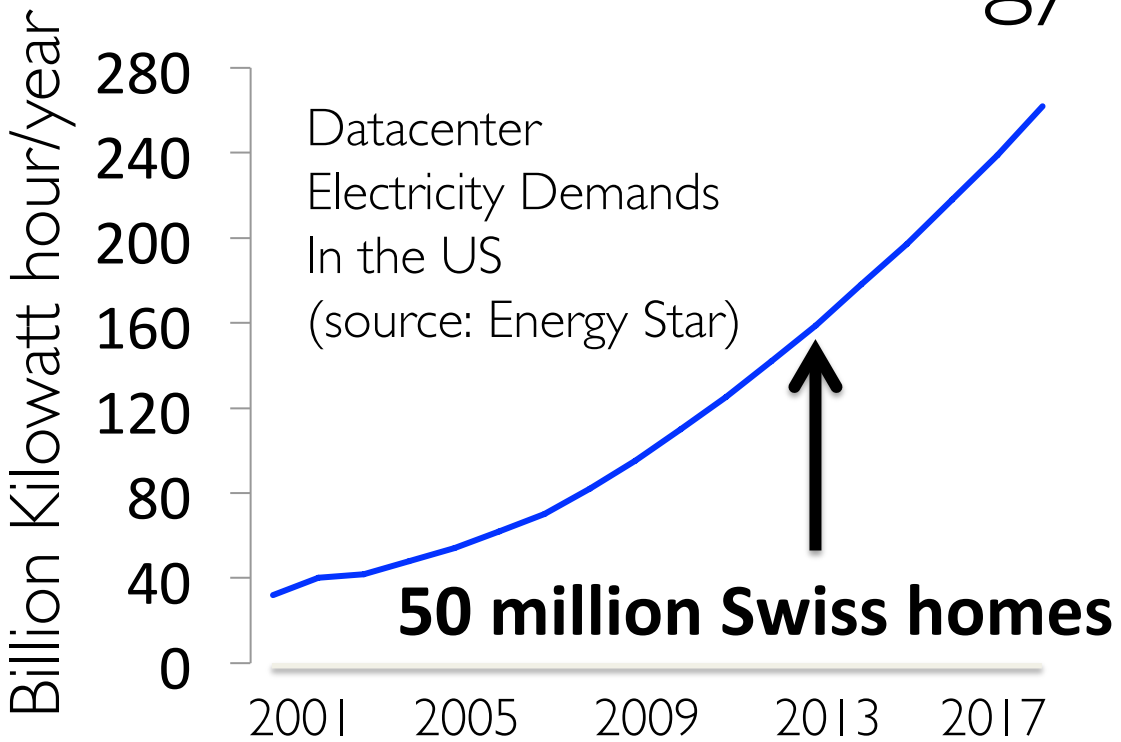
Even in servers:

- With abundant parallelism
- Programmable cores are at efficiency limits
- Soon, cannot power all chip




Hardavellas et. al., "Toward Dark Silicon in Servers", IEEE Micro, 2011

# Higher Demand + Lower Efficiency: Datacenter Energy Not Sustainable!





- Modern datacenters ➔ 20 MW!
- In modern world, 6% of all electricity, growing at >20%!

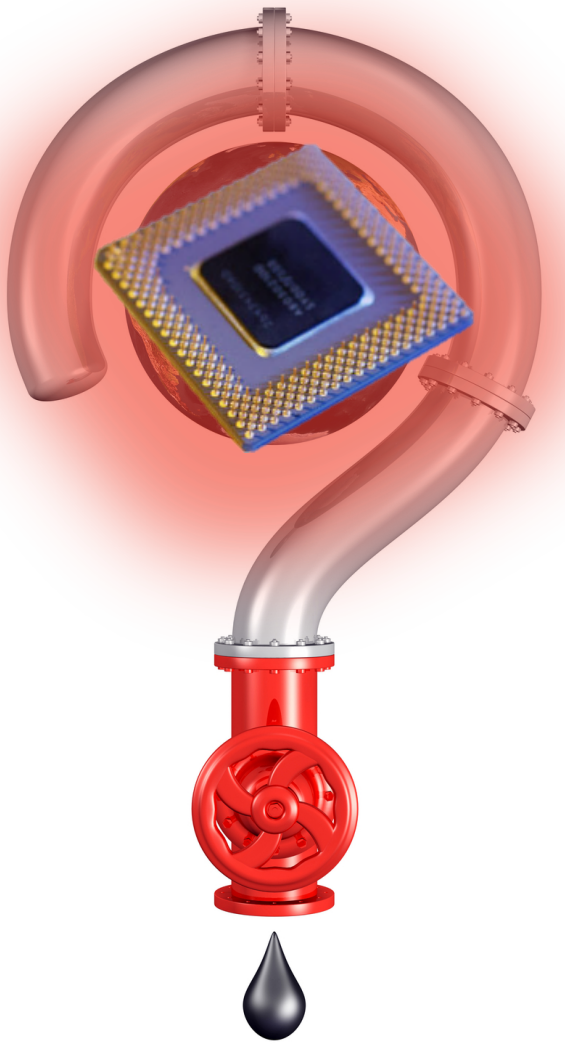
# Big Data


# IT's Future

Bridging

Technologies

# Big Energy



ecocloud

# Center to bring efficiency to data

- 16 faculty, 50 researchers
- Around $3M/year budget

# Mission:

- Energy-efficient data-centric IT
- From algorithms to machine infrastructure
- Maximizing Performance/TCO for Big Data



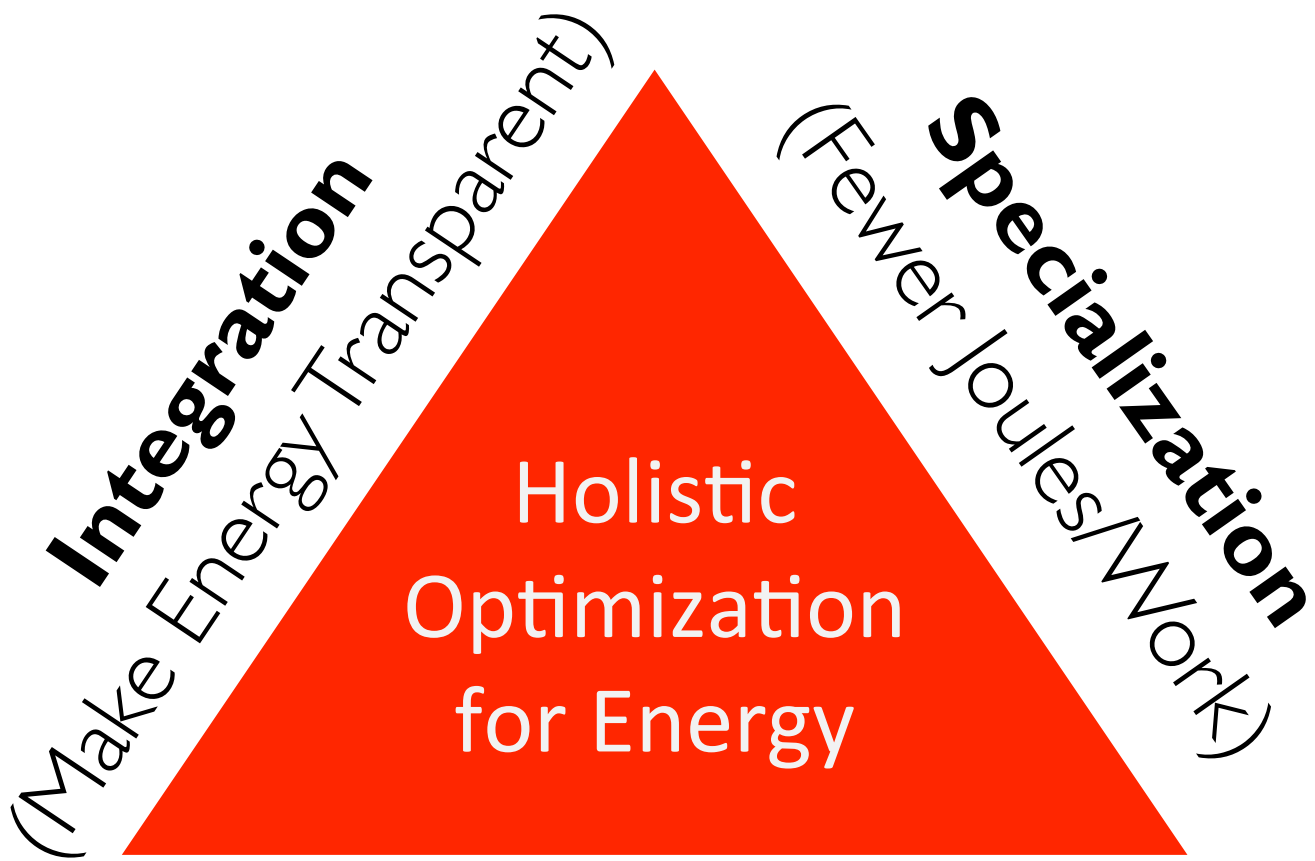







ecocloud.ch

# Our Vision:
# The ISA Triangle of Efficiency



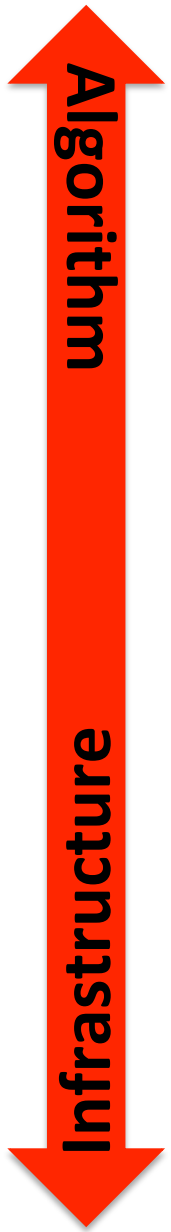
# Holistic Optimization for Energy

Conventional stacks:

- Functionality interfaces

ISA stacks interfaces for:

- Resources monitoring/ allocation
- Specialization bypass
- QoS/accuracy spec.

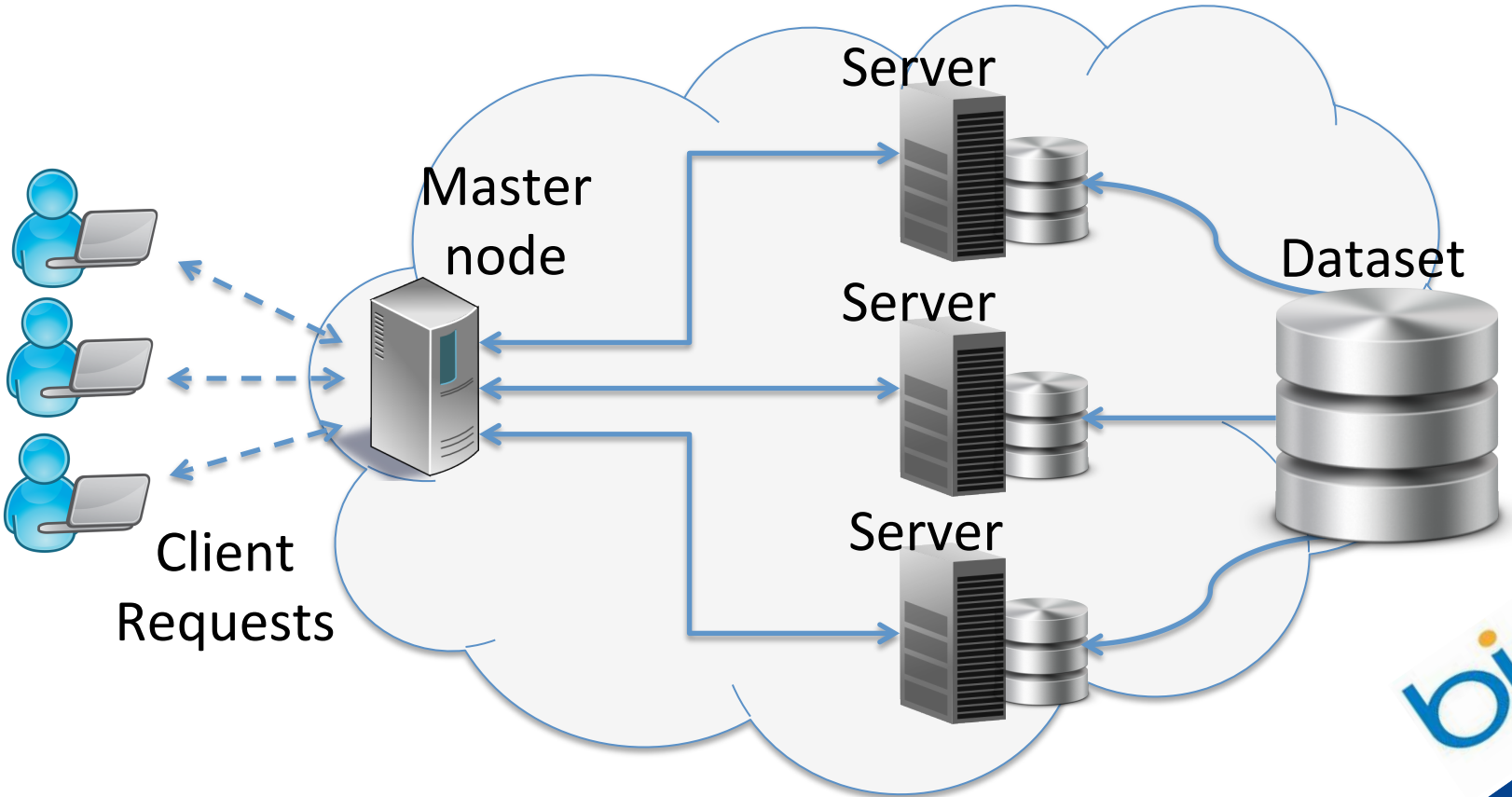




# Outline

1. Two inflection points colliding
2. How bad are today's servers?
   - Scale-Out Workloads
   - CloudSuite 2.0
3. How do we build efficient servers?

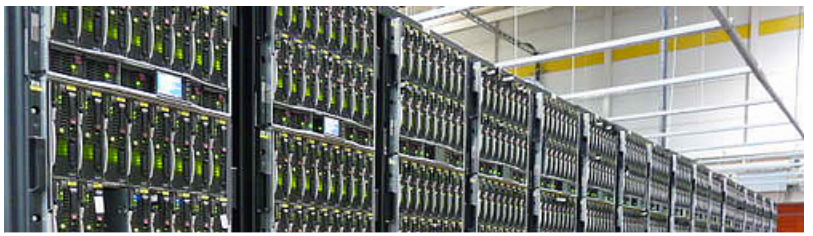
# Data-Intensive Online Services




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

# Scale-Out Datacenters

Vast data sharded across servers

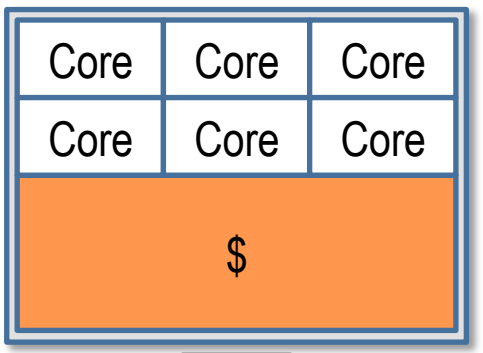


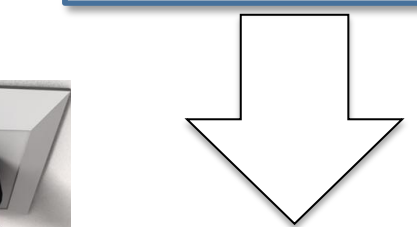
Memory-resident workloads

- Necessary for performance
- Major TCO burden

Processors access data in memory

- Abundant request-level parallelism
- Performance scales with core count



*Data*

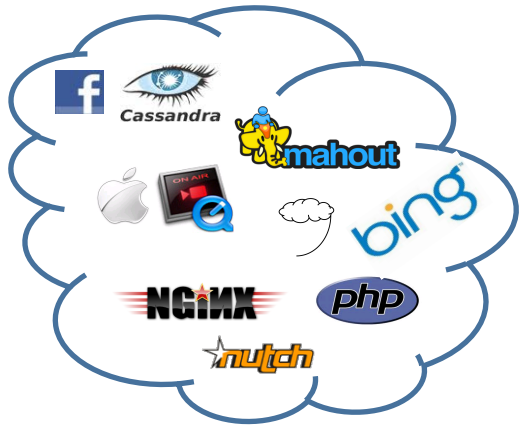
Memory

**Maximize performance for better TCO**

# 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

# CloudSuite 2.0

(released @ parsa.epfl.ch/cloudsuite)

**Data Analytics**
Machine learning


**Data Caching**
Memcached


**Data Serving**
Cassandra NoSQL



**Graph Analytics**
TunkRank



**Media Streaming**
Apple Quicktime Server



**SW Testing as a Service**
Symbolic constraint solver

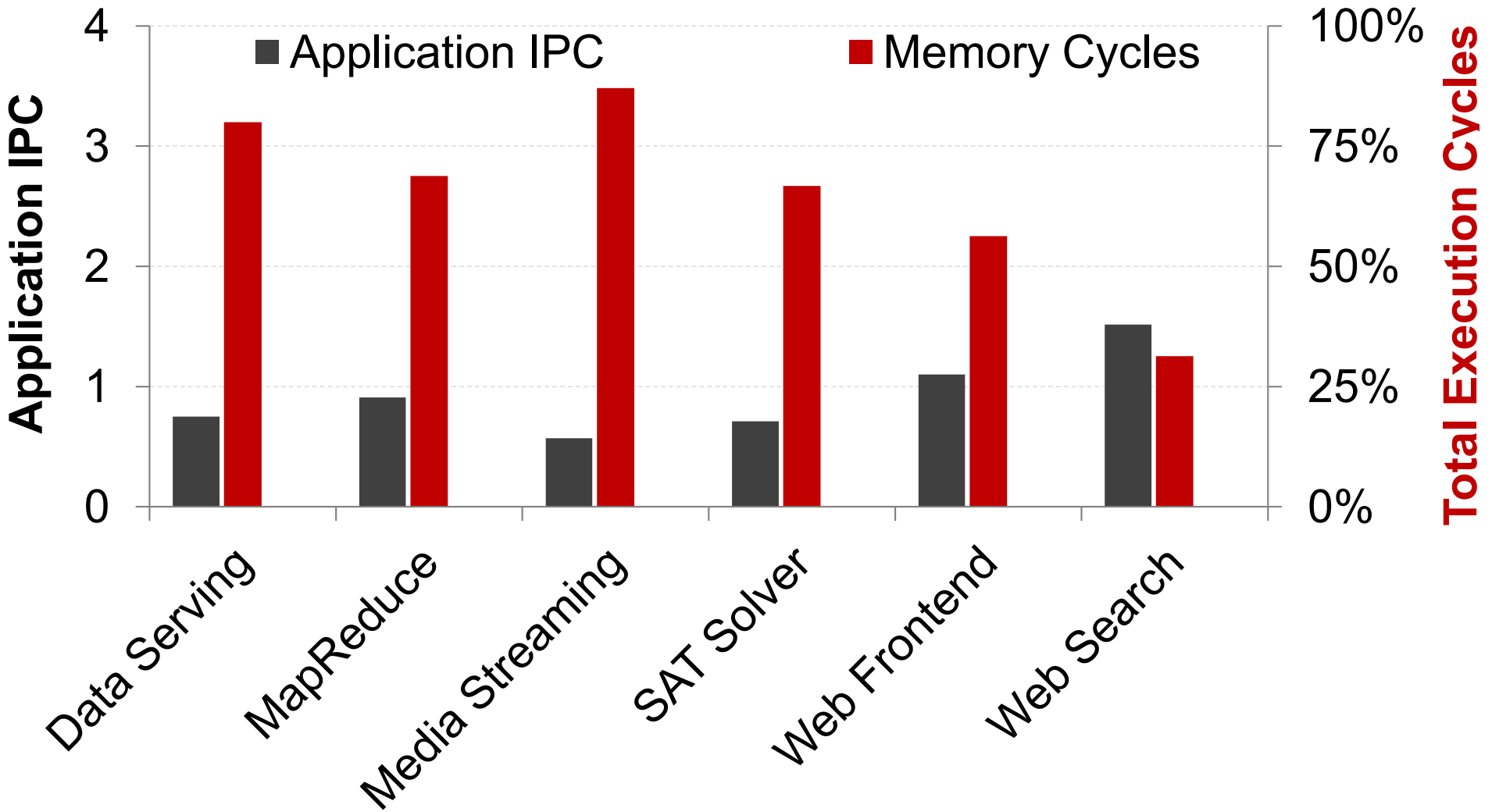
**Web Search**
Apache Nutch



**Web Serving**
Nginx, PHP server


***Covers popular scale-out services***

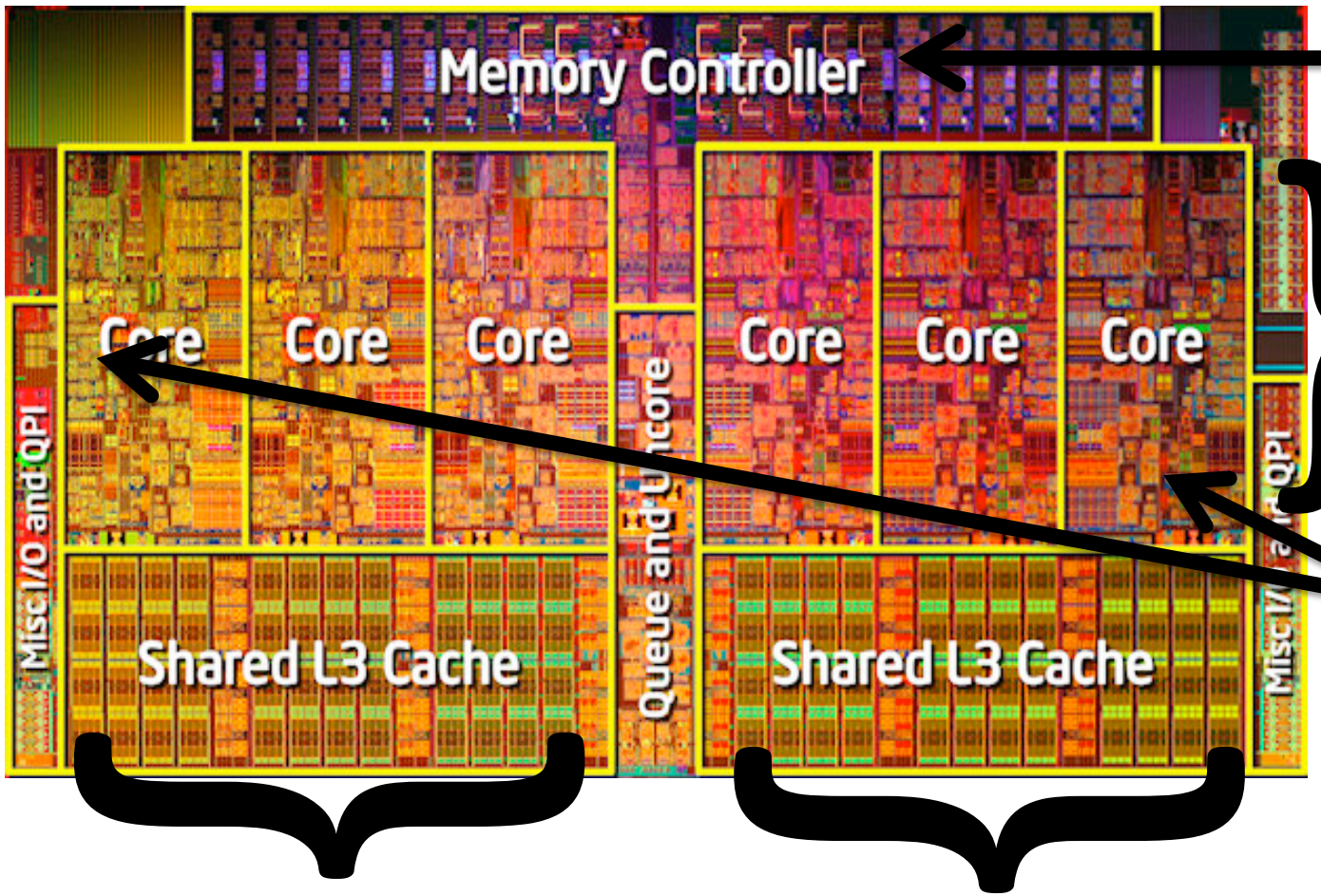
# How efficient are today's cores?




Execute ~1 instruction per cycle

# Clearing the Clouds in a nutshell [ASPLOS 2012]

## Workload/Server Mismatch




B/W unused!

Cores too fat!

Too few cores!

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

# Outline

1. Two inflection points colliding
2. How bad are today's servers?
3. How do we build efficient servers?
   - Scale-Out Processors [ISCA'12, IEEE Micro'12]
   - Scale-Out NoCs [MICRO'12]
   - Die-Stacked Caches [ISCA'13]
   - DB Accelerators [MICRO'13]

# Scale-Out Processors: [ISCA'12]
# Optimal TCO for Scale-Out Services

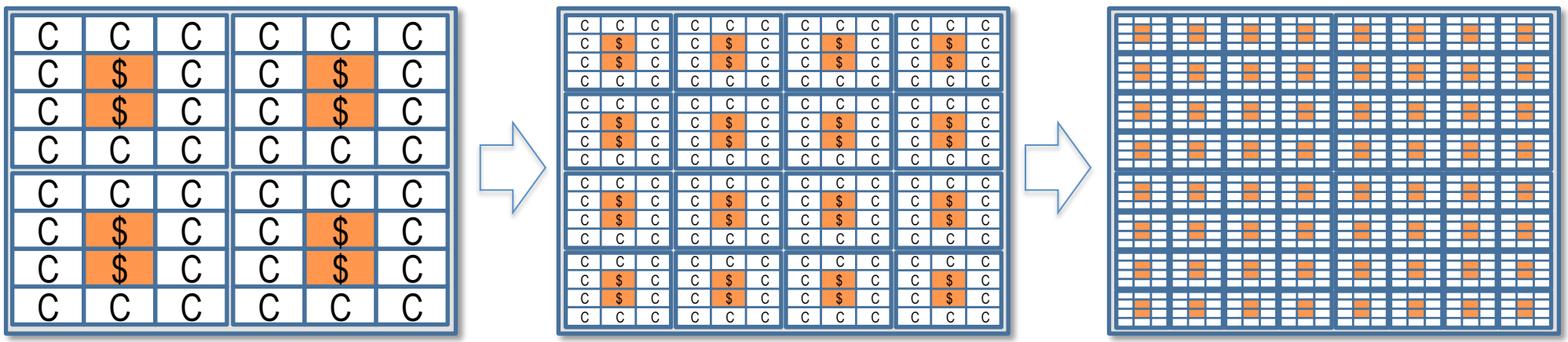
One or more pods

Each pod is a standalone server

- Runs a full OS/software stack

No inter-pod connectivity or coherence

- Scalability and optimality across generations



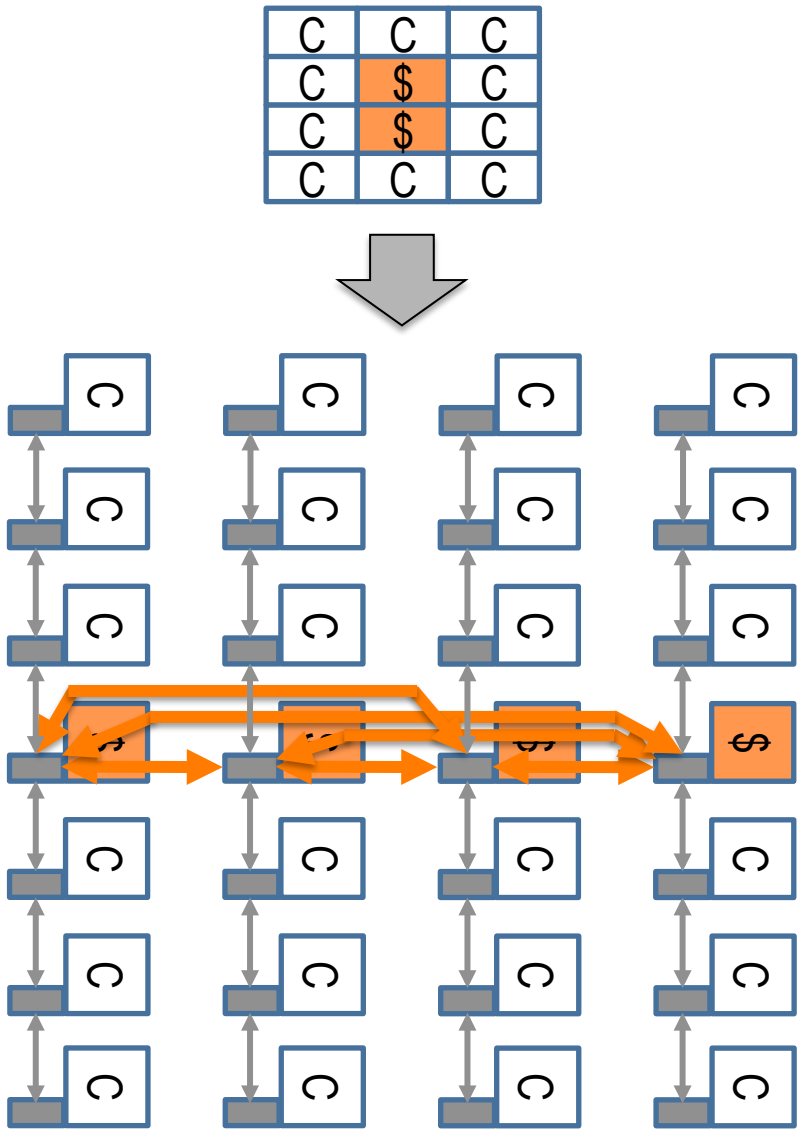

Inherently optimal & scalable

# NOC-Out: [Micro'12]
# Specialized Network-on-Chip for Pods

LLC network:
- Flattened Butterfly (FB) topology
- Expensive but limited to LLC

Request & Reply networks:
- Tree topology
- Limited connectivity for efficiency
- SW stack unaffected

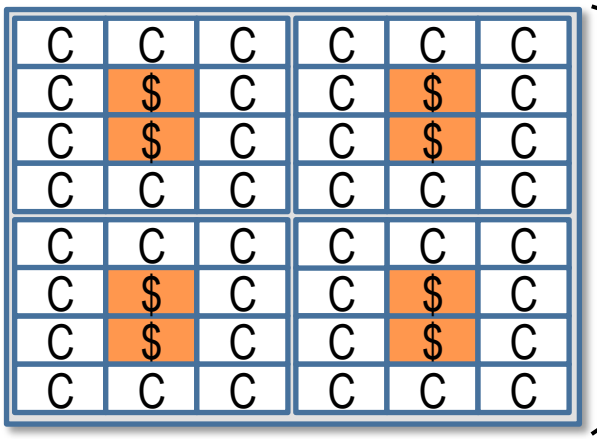
FB's performance at 1/10$^{th}$ cost

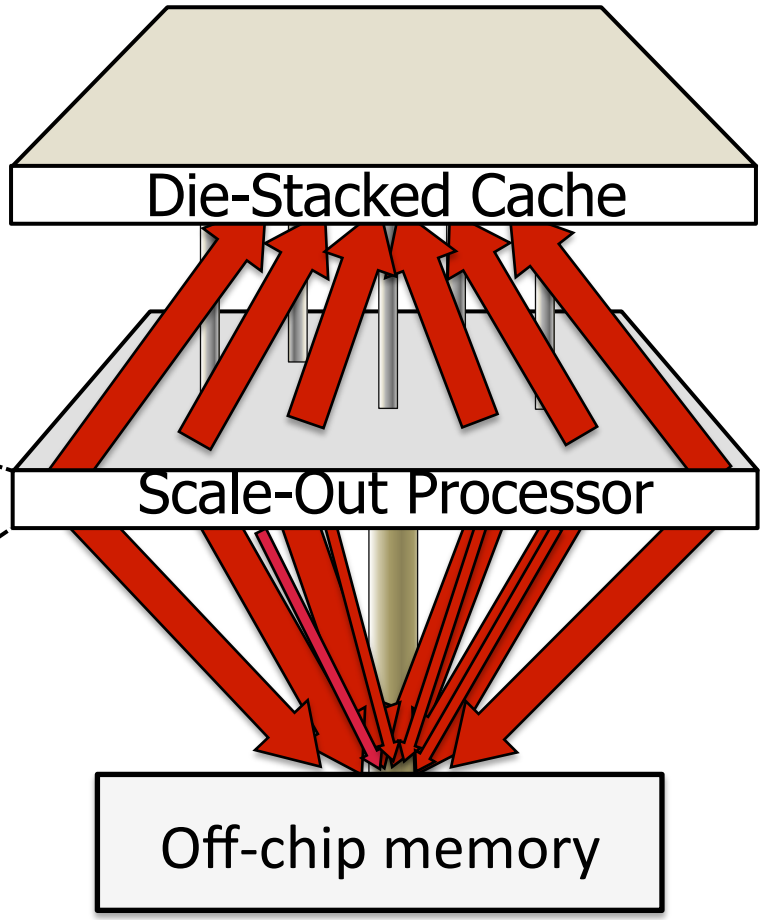


# Footprint Cache: [ISCA'13]
# Effective Die-Stacked Caching for Servers

Die-Stacked Caching:

- Rich connectivity → High on-chip BW
- High capacity → Low off-chip BW




Footprint Cache:

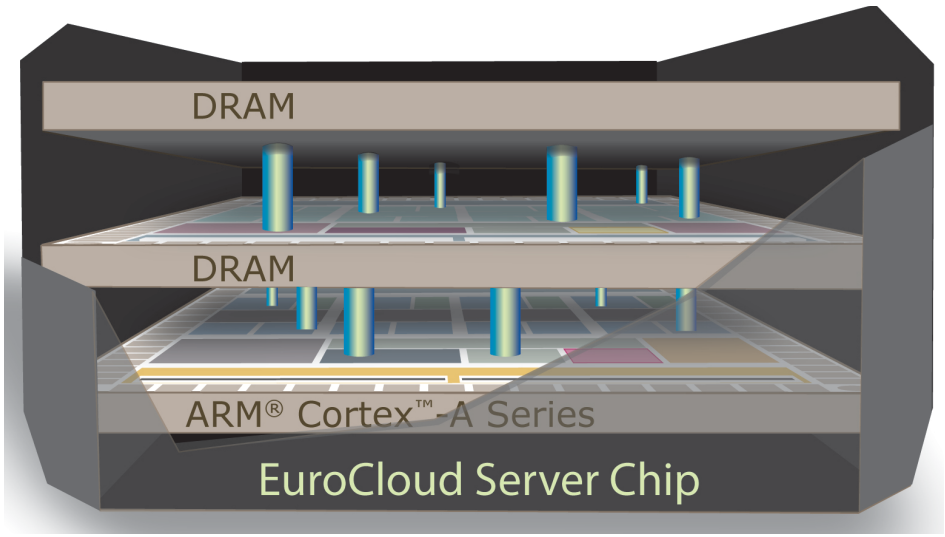
- Allocate tags for pages
- Predict & fetch page's footprint

# 3D Scale-Out Chip: An Architecture for Big Data

(www.eurocloudserver.com)

## Specialized chip for servers:

- Scale-Out Processors
- Die-stacked cache
- 10x more efficiency
- Runs Linux LAMP stack




## Foundation for EuroServer!

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ARM

SEVENTH FRAMEWORK PROGRAMME

NOKIA Connecting People

imec

UCyprus

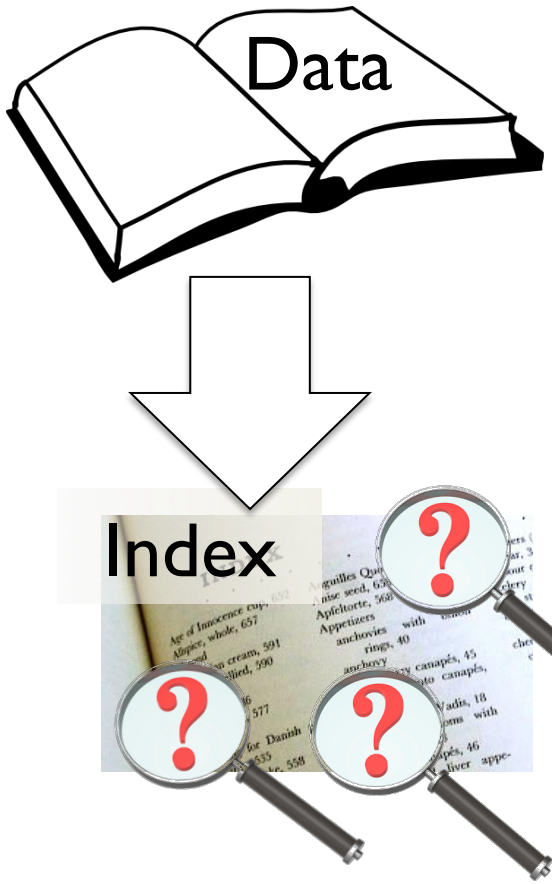
# Outline

1. Two inflection points colliding
2. How bad are today's servers?
3. How do we build efficient servers?
   - Scale-Out Processors [ISCA'12, IEEE Micro'12]
   - Scale-Out NoCs [MICRO'12]
   - Die-Stacked Caches [ISCA'13]
   - DB Accelerators [MICRO'13]

# Databases underlie data-intensive apps

Most frequent task: find data

- E.g., build a user's Facebook page




Indexes used for fast data lookup

- Rely on pointer-intensive data structures

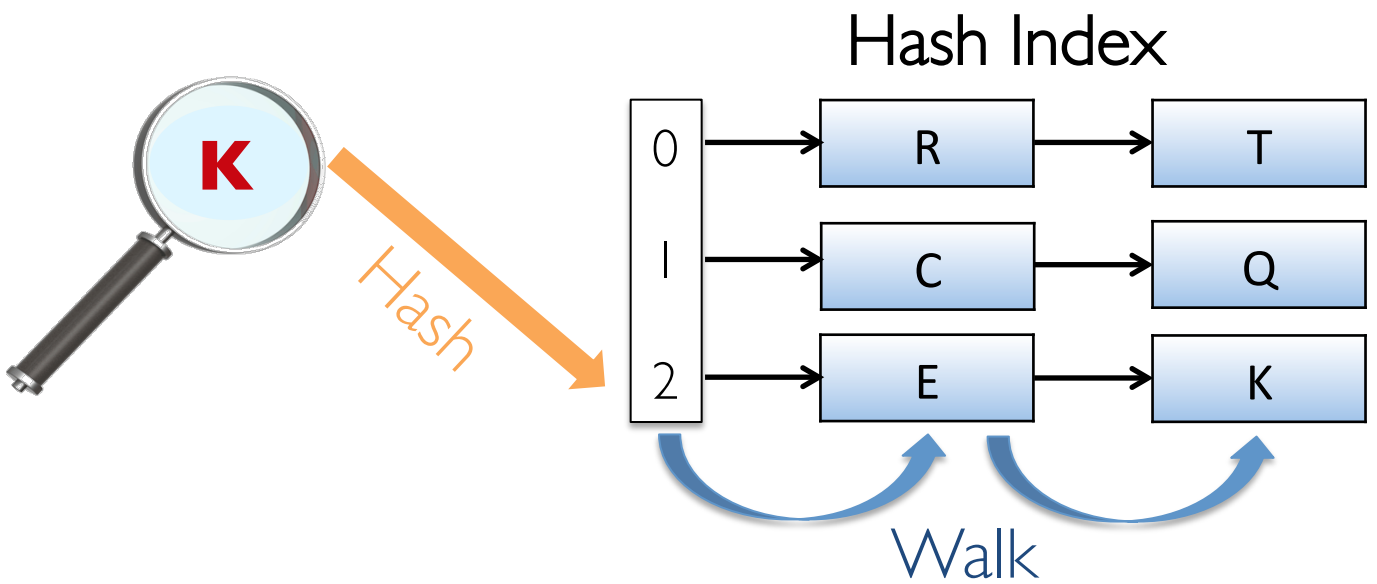
Indexing efficiency is critical

- Many requests, abundant parallelism
- Power-limited hardware

**Need high-throughput and energy-efficient index lookups**

# Indexing Basics

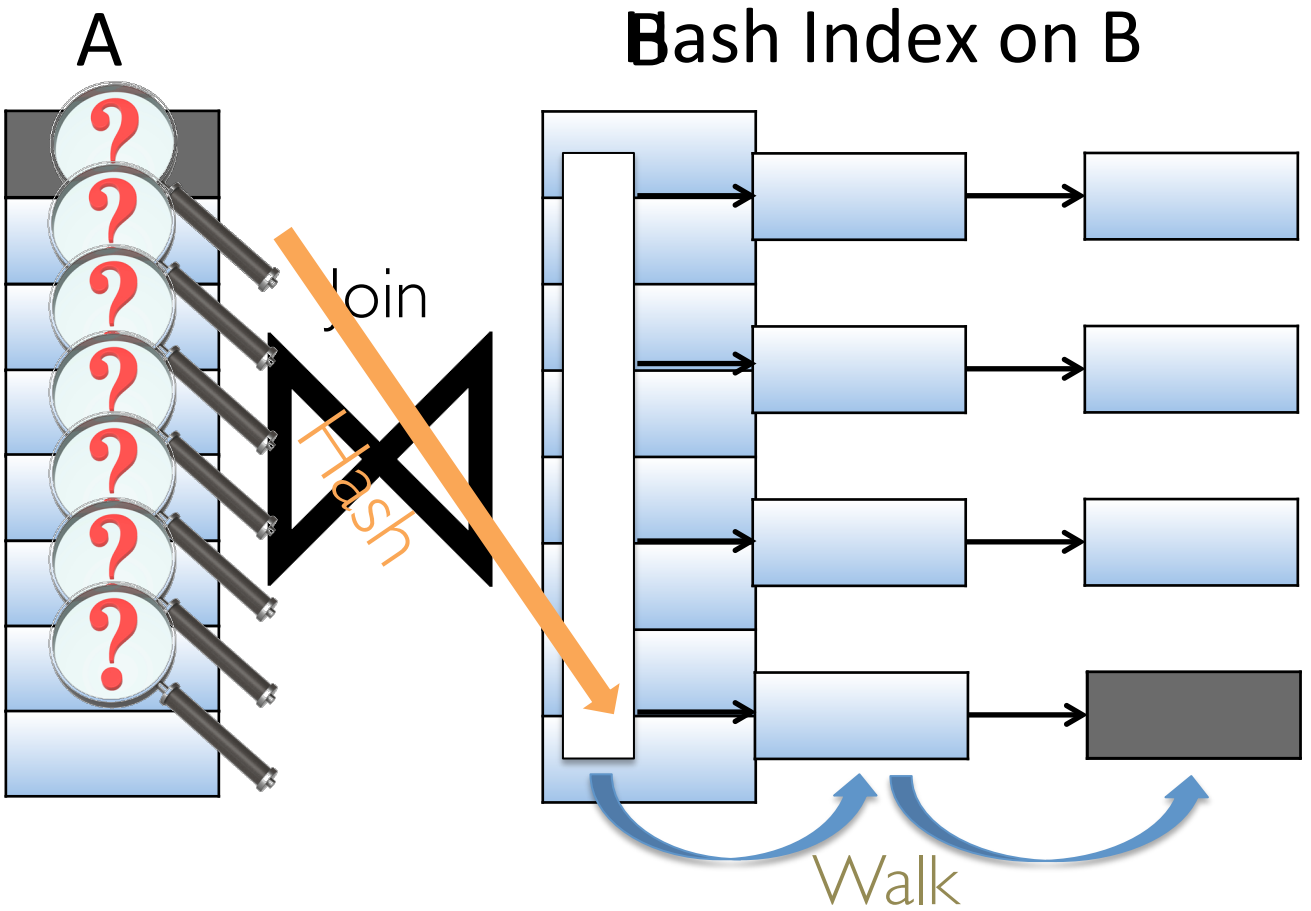
Hash index: fundamental index structure




Dominant operation: join via hash index

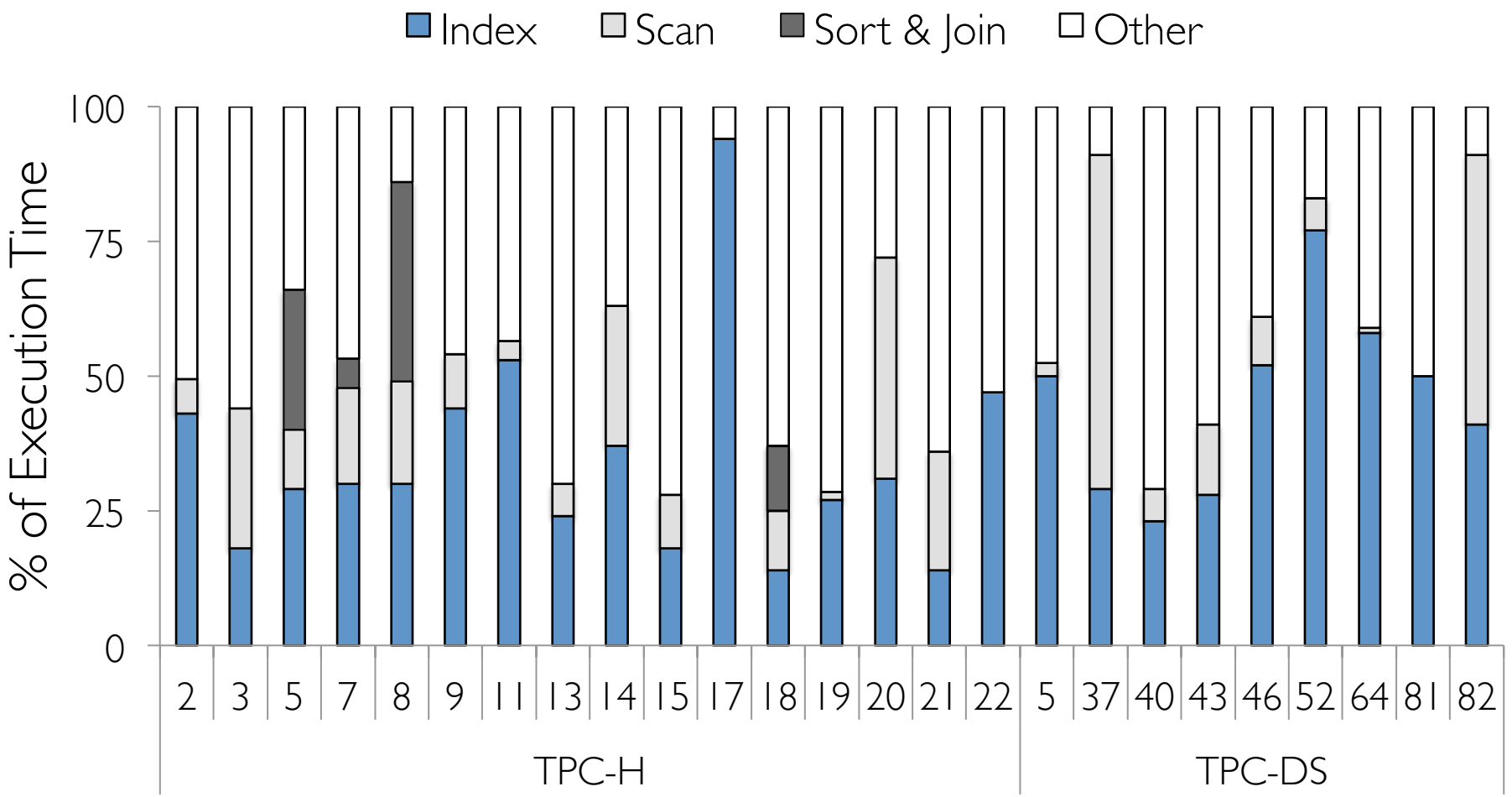
# Join via Hash Index

Look up the index for every entry in A



# How Much Time is Spent Indexing?

**Measurement on Xeon 5670 CPU with 100GB Dataset**




Indexing is the biggest contributor to execution time

# Dissecting Index Lookups

Hash: Avg. 30% time of each lookup

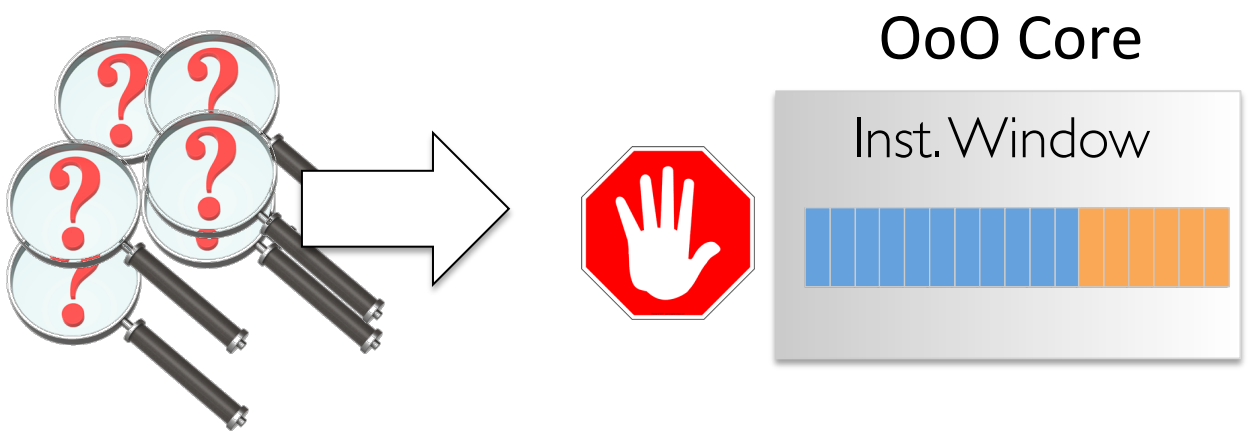
– Computationally intensive, high cache locality

Walk: Avg. 70% time of each lookup

– Trivial computation, low cache locality

Next lookup: Inherently parallel

– Beyond the instruction window capacity



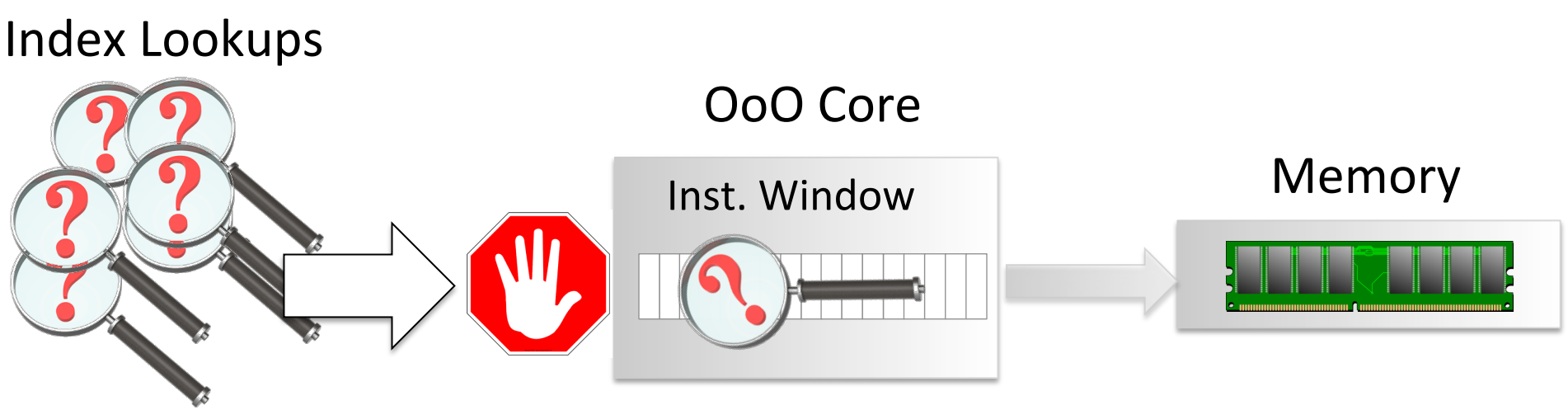
# Index Lookups on General-Purpose Cores

## Index Lookups

- Data in memory
- Inherent parallelism

## OoO Cores

- Pointer-chasing → Low MLP
- Limited OoO inst. window
  - One lookup at a time




OoO cores ill-matched to indexing

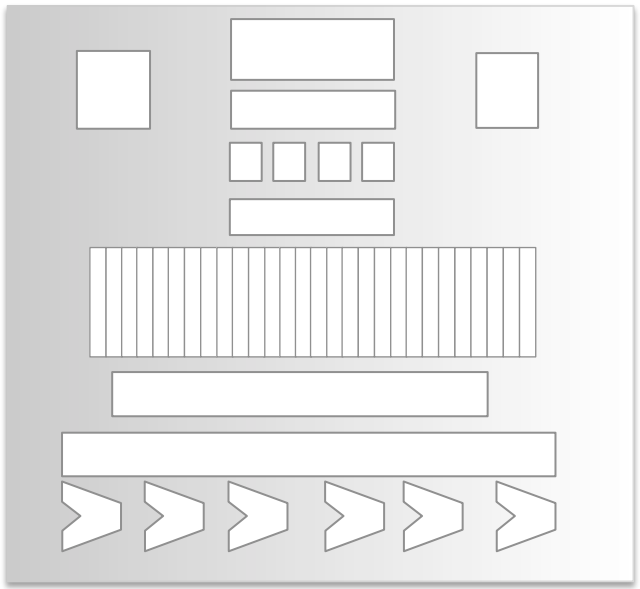
# Roadmap for Efficient and High-Throughput Indexing

1. Specialize
   - Customize hardware for hashing and walking
2. Parallelize
   - Perform multiple index lookups at a time
3. Generalize
   - Use a programmable building block

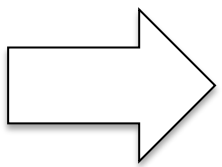
# Step 1: Specialize

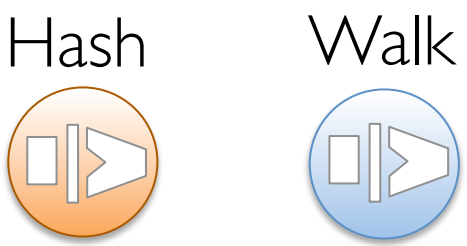
Design a dedicated unit for hash and walk

- Hash: compute hash values from a key list
- Walk: access the hash index and follow pointers



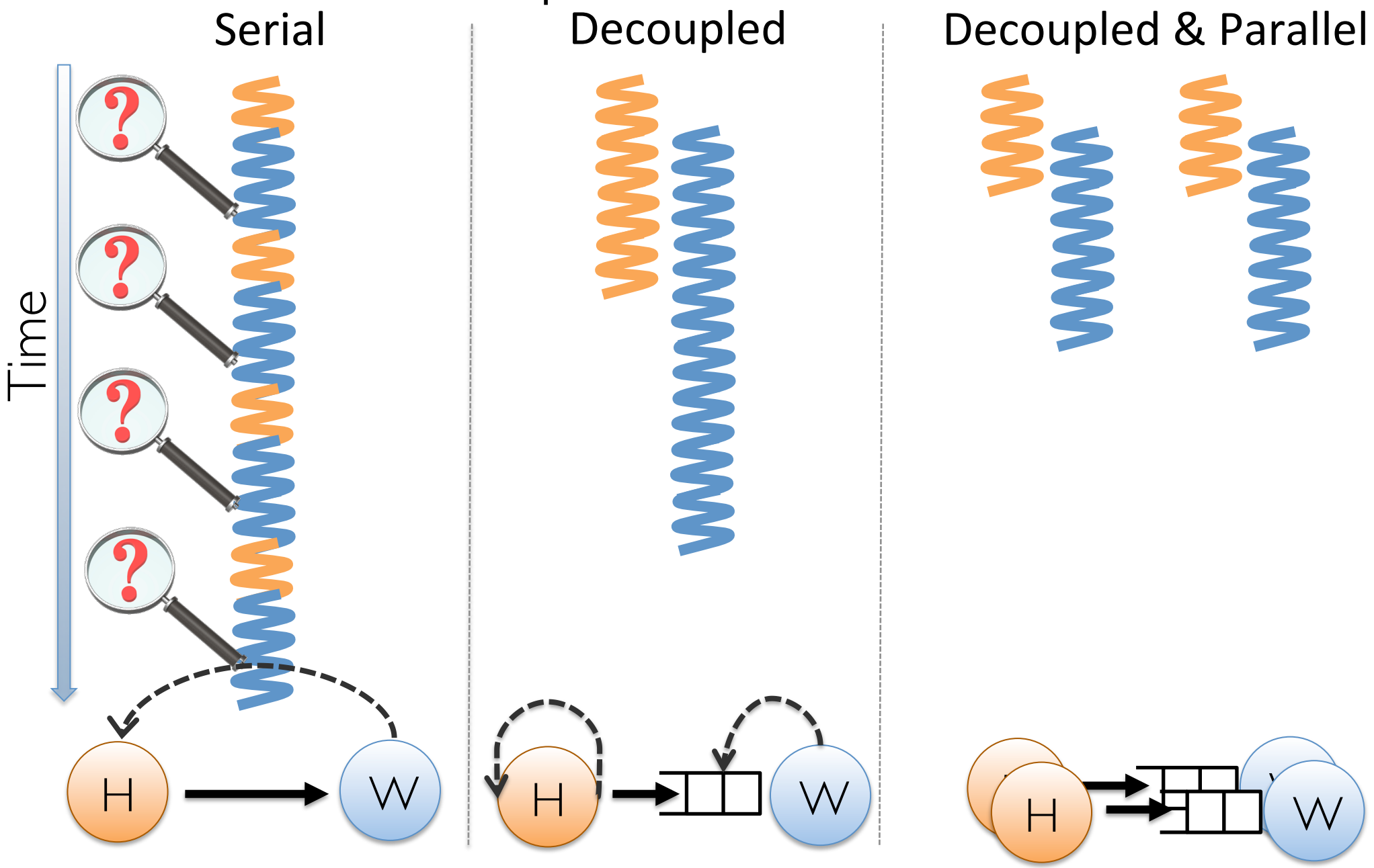
General-purpose
OoO





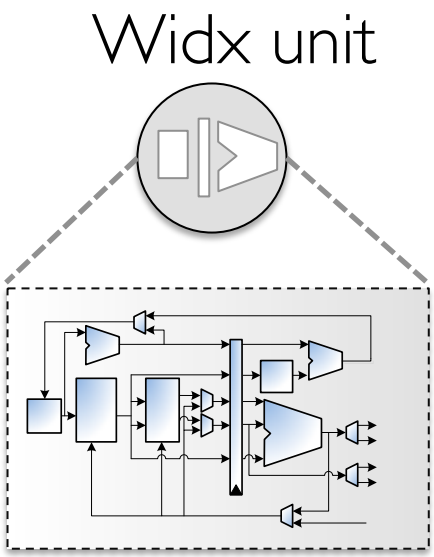

Specialized
hash and walk hardware

# Step 2: Parallelize

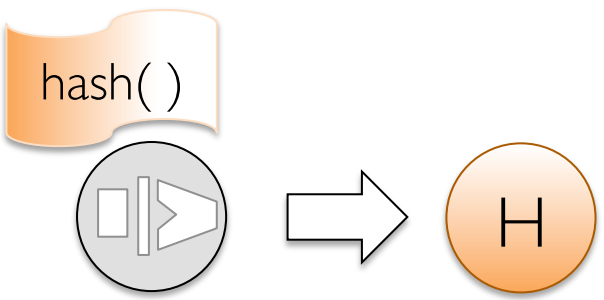


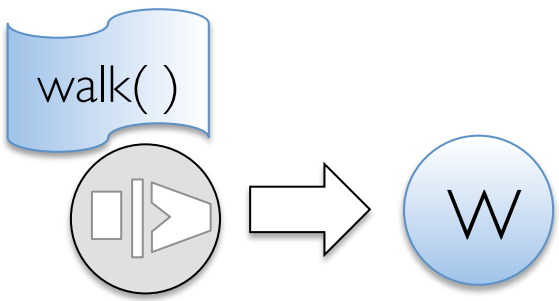
# Step 3: Generalize Widx Units

Common building block for hash and walk
- Two-stage RISC core
- Custom ISA



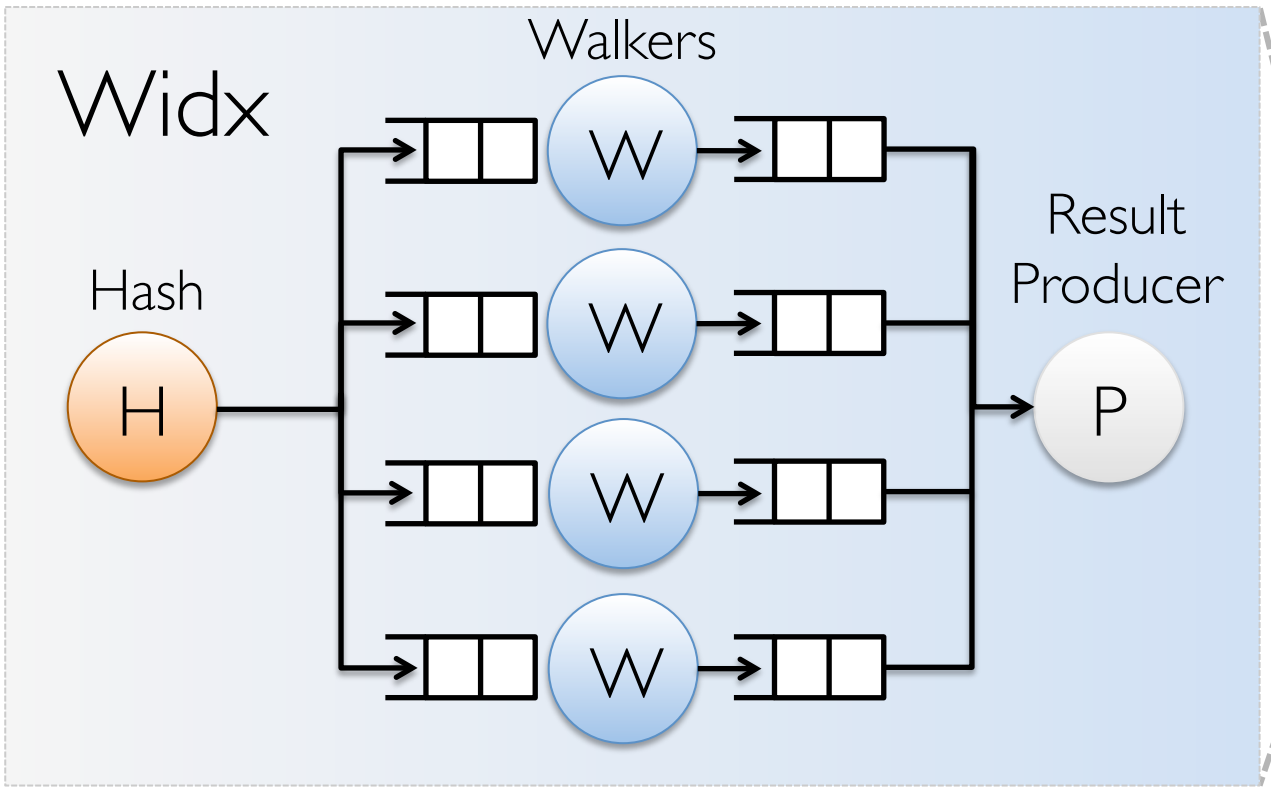

Programmable
- Execute functions written in Widx ISA
- Support limitless number of data structure layouts



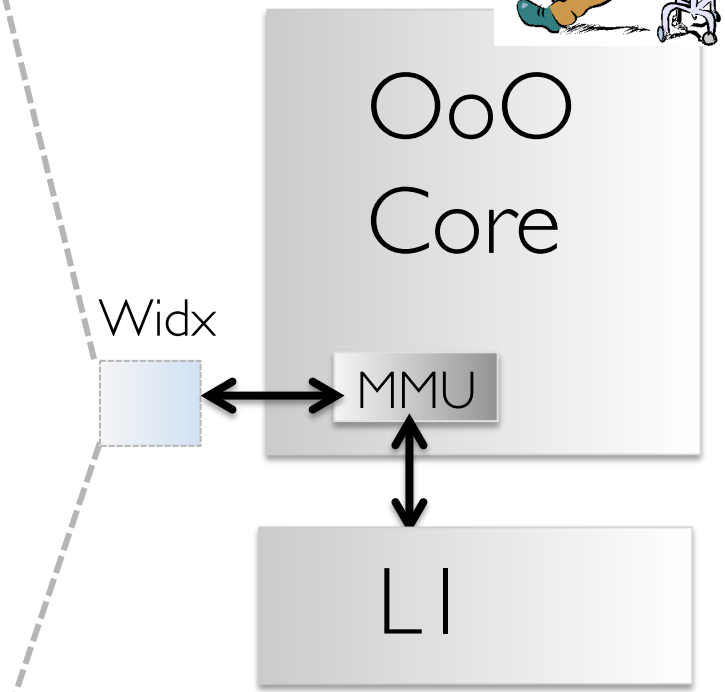



# Putting it all together: Widx

When Widx runs, core goes idle





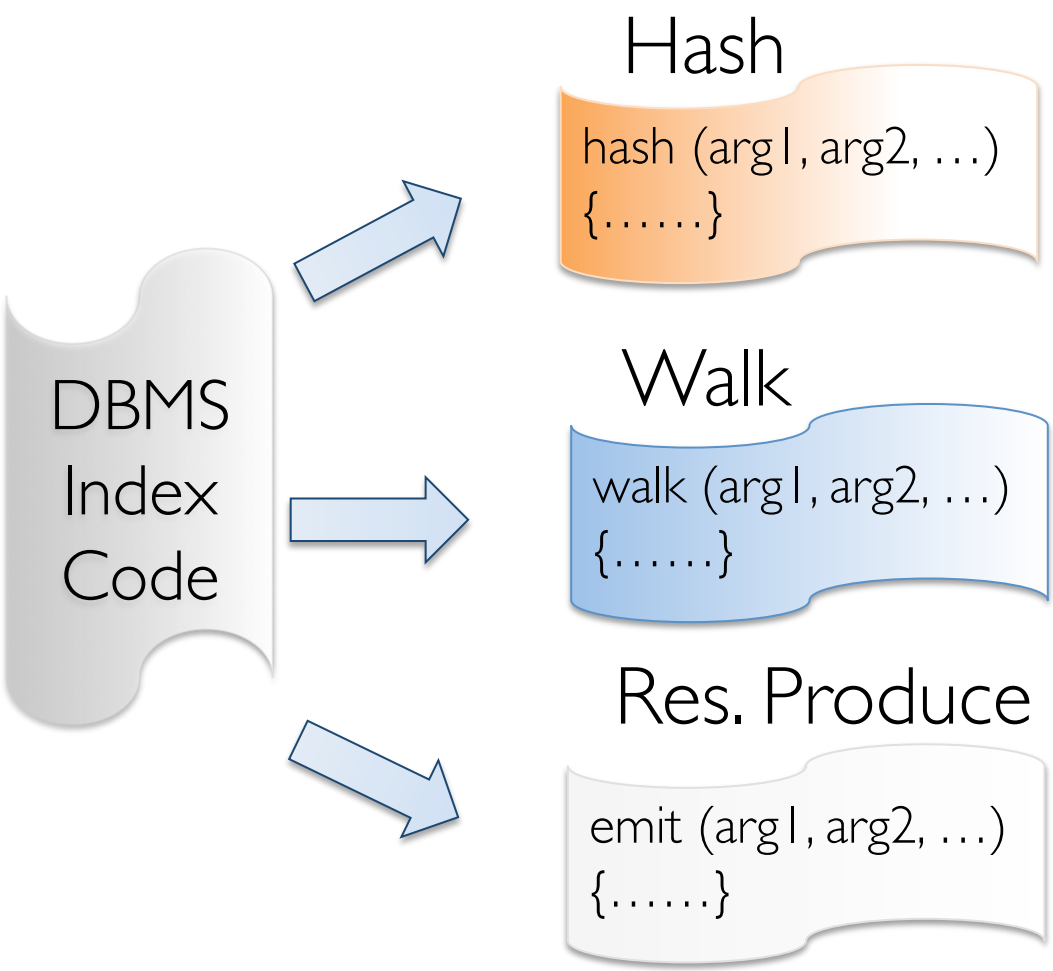



Simple, parallel hardware

# Programming Model

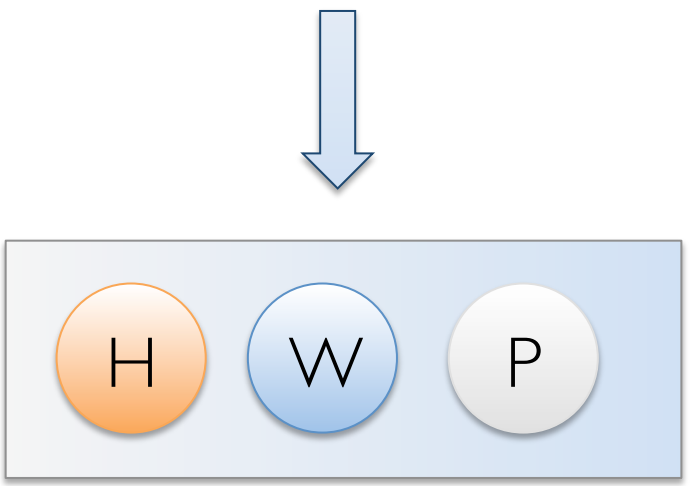
## Development

Write code for each unit
and compile for Widx ISA




## Execution

Communicate
Load the code
query-specific inputs



# Methodology

Flexus simulation infrastructure [Wenisch '06]

## Benchmarks

- TPC-H on MonetDB
- TPC-DS on MonetDB
- Dataset: 100GB

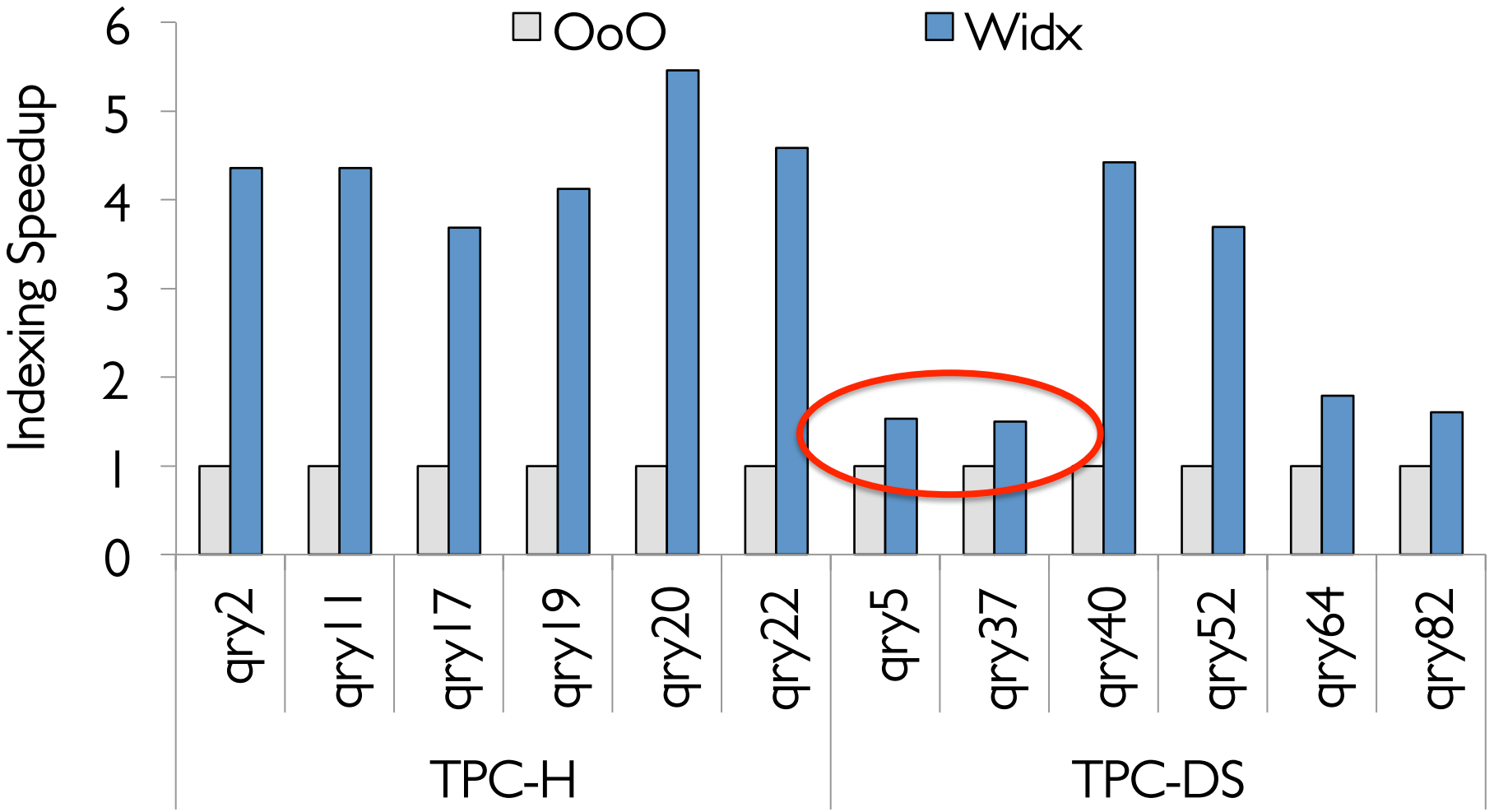
## uArch Parameters

- Core Types
  - OoO: 4-wide, 128-entry ROB
  - In-order: 2-wide
- Frequency: 2GHz
- L1 (I & D): 32KB
- LLC: 4MB

## Area and Power

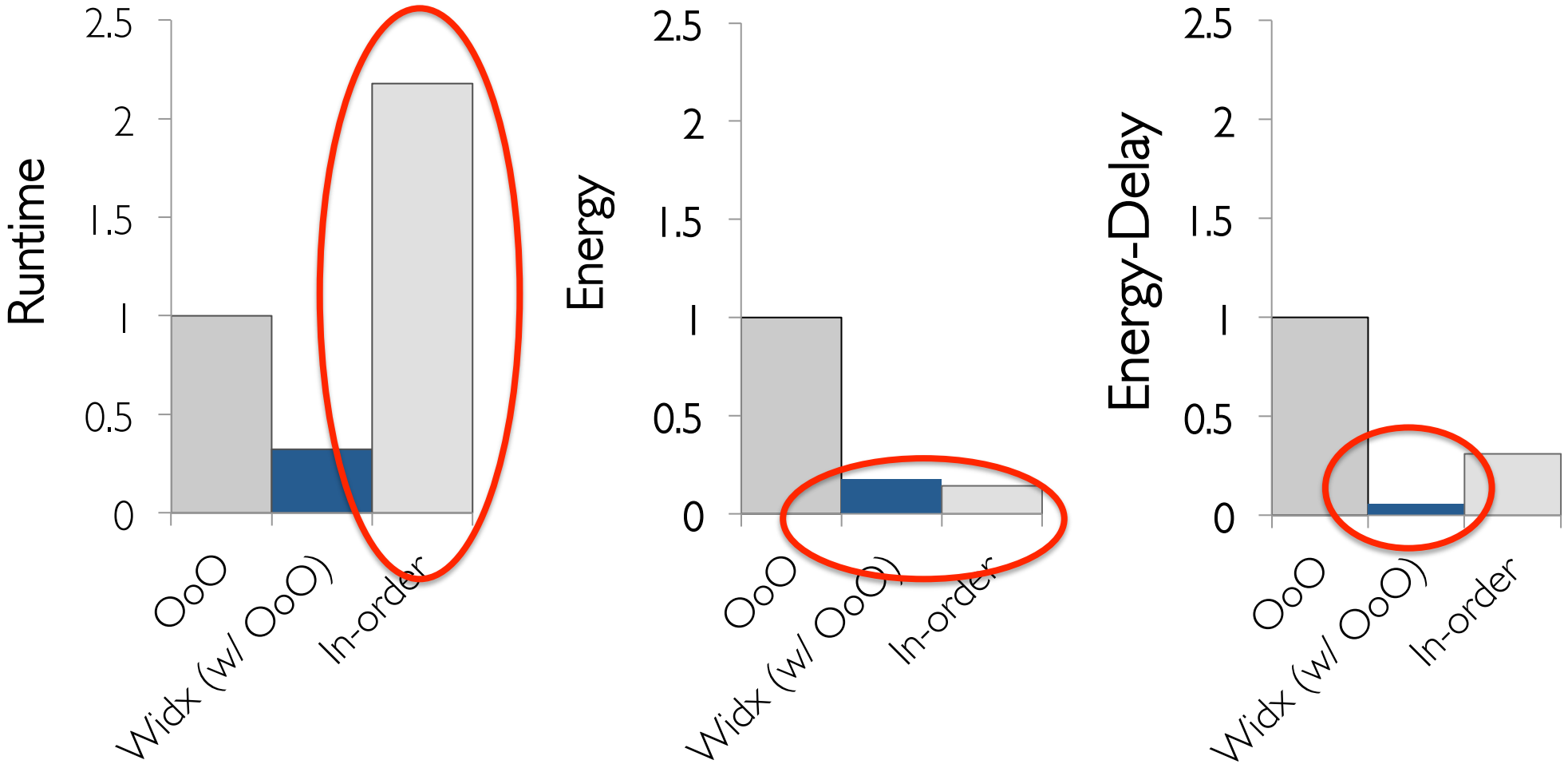
- Synopsys Design Compiler
- Technology node: TSMC 40 nm, std. cell
- Frequency: 2GHz
- Widx Area: 0.24mm$^2$
- Widx Power: 0.3W

# Widx Performance




3x higher indexing throughput

# Widx Efficiency




5.5x reduction in indexing energy vs. OoO core

# Summary

Two IT trends on a collision course:

- Data growing at ~10x/year
- Nearing end of Dennard & Multicore Scaling
- Need technologies to bring efficiency to data

Our contributions:

- Scale-Out Processors
- High-throughput index accelerators

Long term:

Integrate + Specialize + Approximate (ISA for Big Data)

# Thank You!

For more information please visit us at

**ecocloud.ch**



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE