We’re still having FAWN

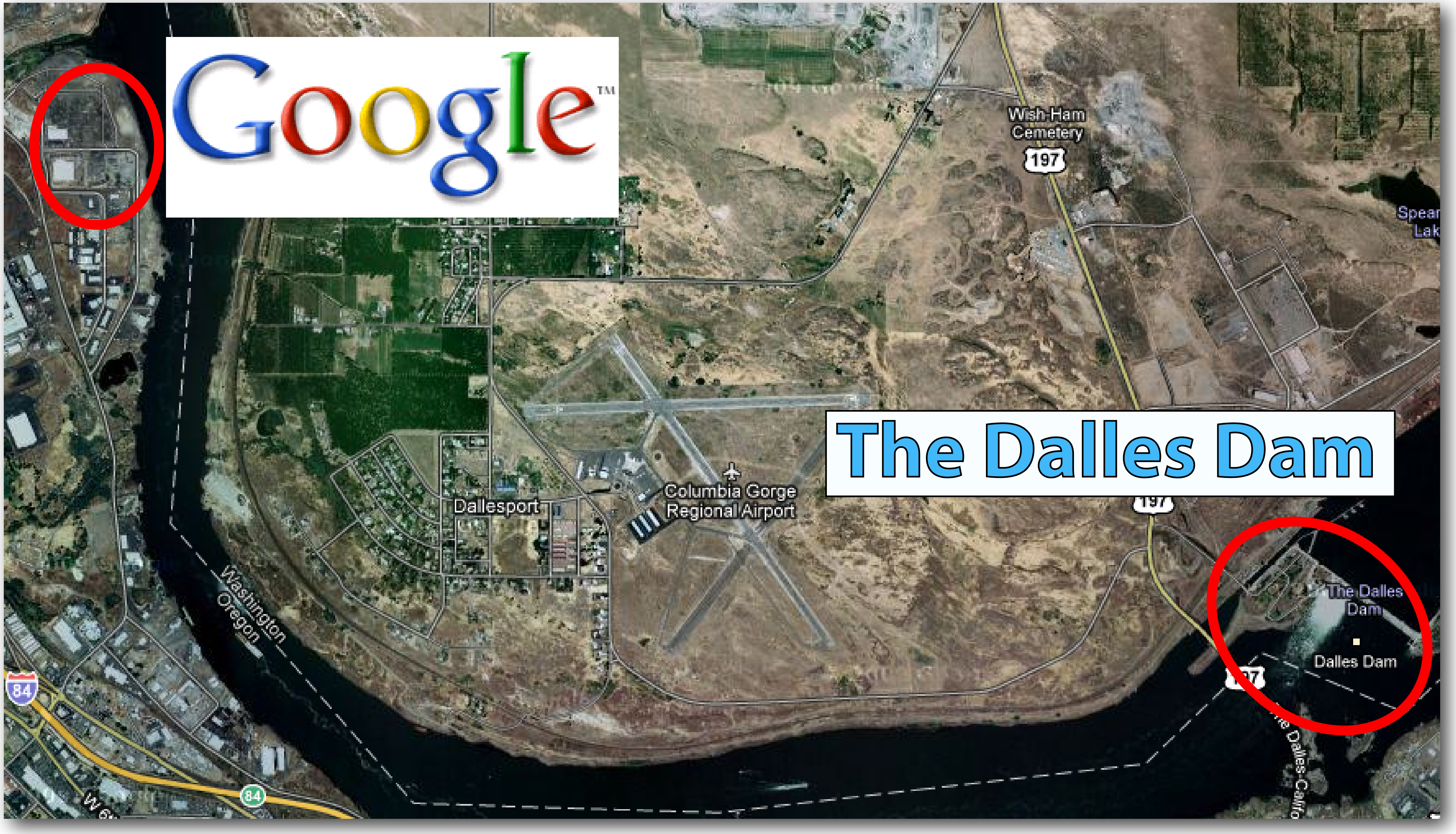
David Andersen, Michael Kaminsky*, Michael A. Kozuch*, Padmanabhan Pillali* ,Vijay Vasudevan, Amar Phanishayee, Lawrence Tan, Jason Franklin, Iulian Moraru, Sang Kil Cha, Hyeontaek Lim, Bin Fan, Reinhard Munz, Nathan Wan, Jack Ferris, Hrishikesh Amur**, Wolfgang Richter, Michael Freedman***, Wyatt Lloyd***, Dong Zhou

Carnegie Mellon University *Intel Labs Pittsburgh
** Princeton University *** Georgia Tech

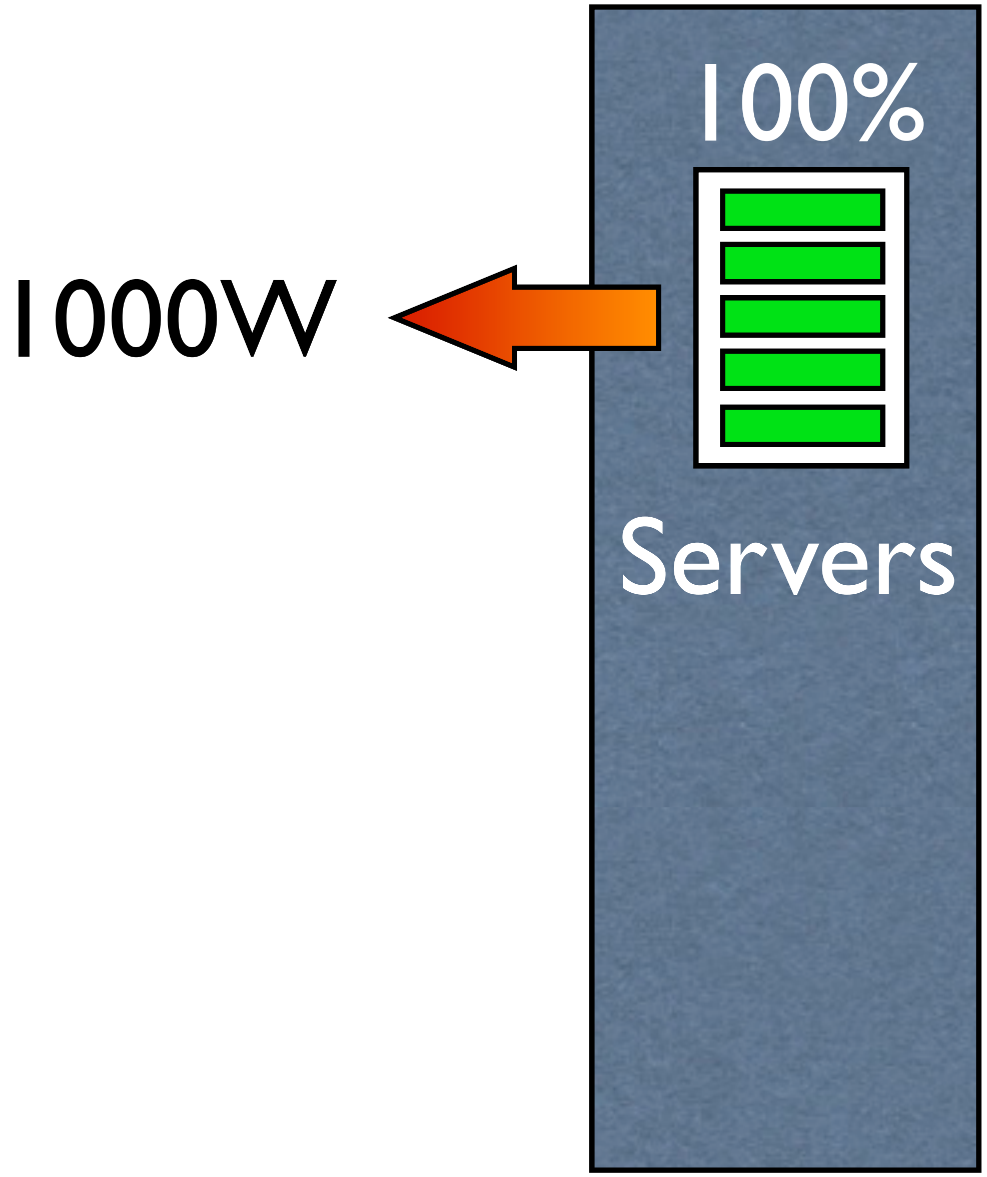
Power limits computing

Power limits computing

Power limits computing



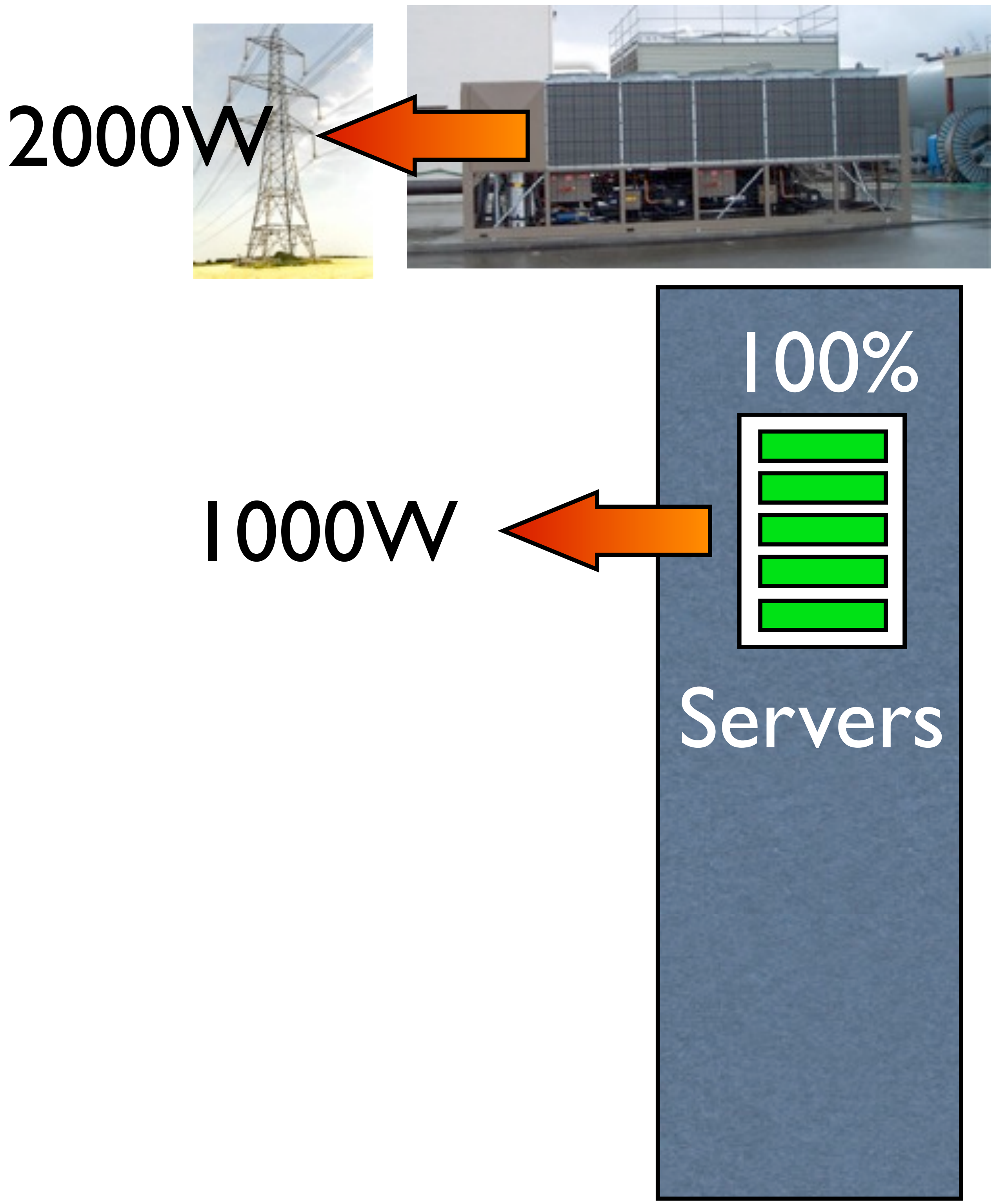
Power limits computing



1000W

100%

Servers



2000W

100%

1000W

Servers

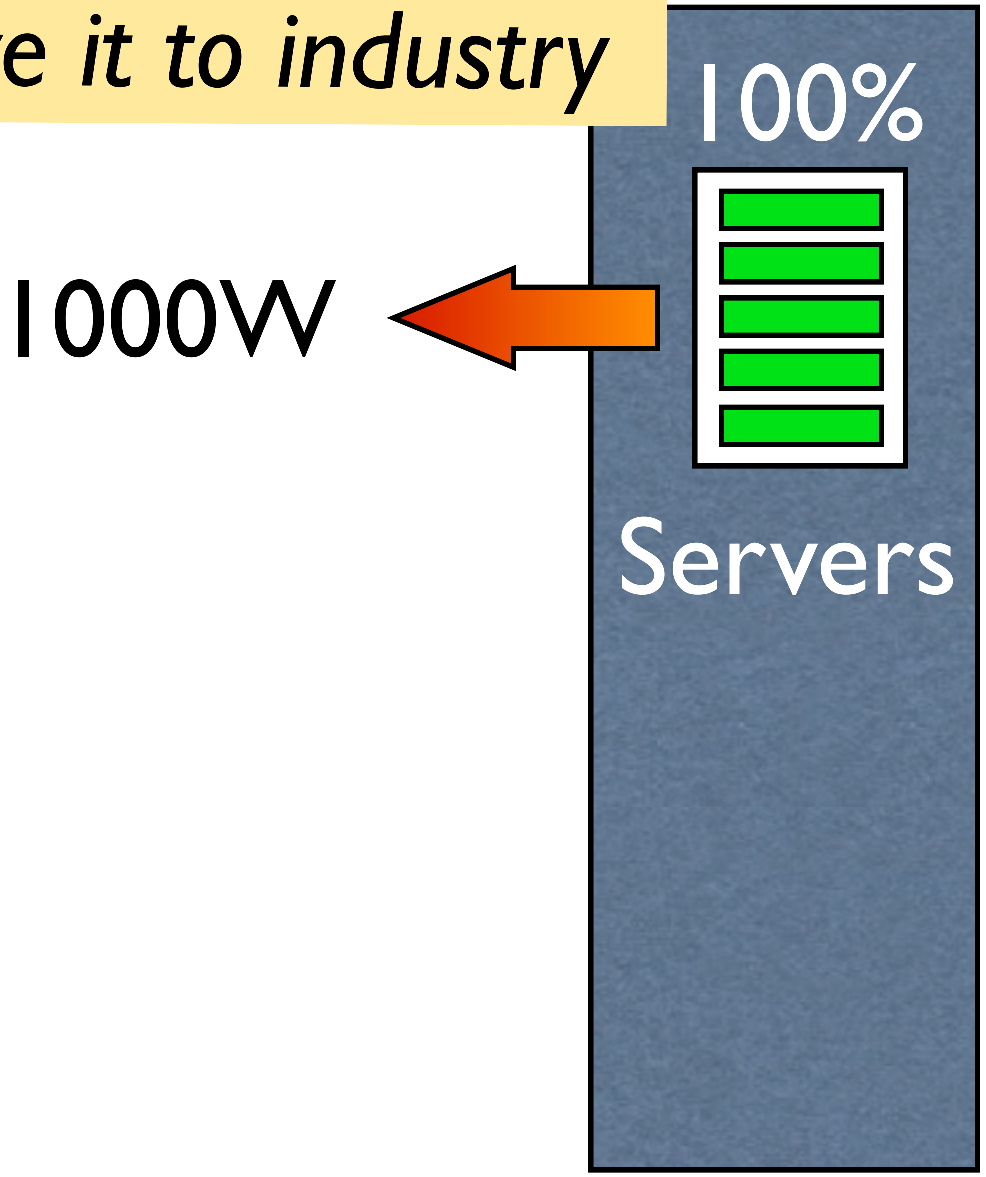
Infrastructure:

PUE

2005: 2–3

2012: ~1.1

Leave it to industry



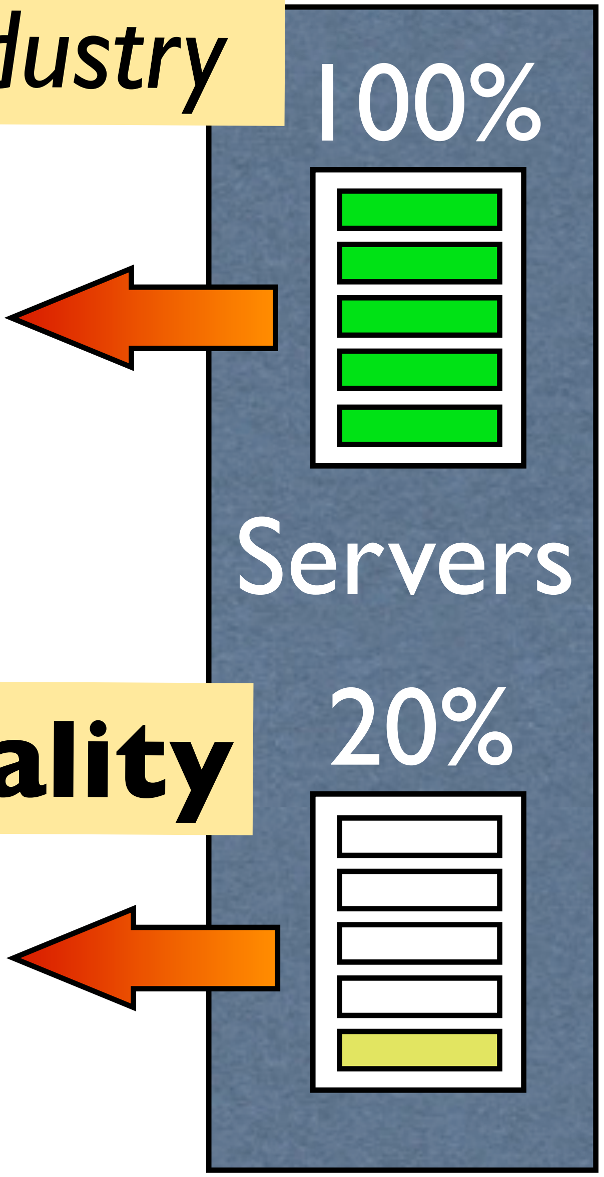
Infrastructure:

PUE

2005: 2–3

2012: ~1.1

Leave it to industry



1000W

Proportionality

750W

200W

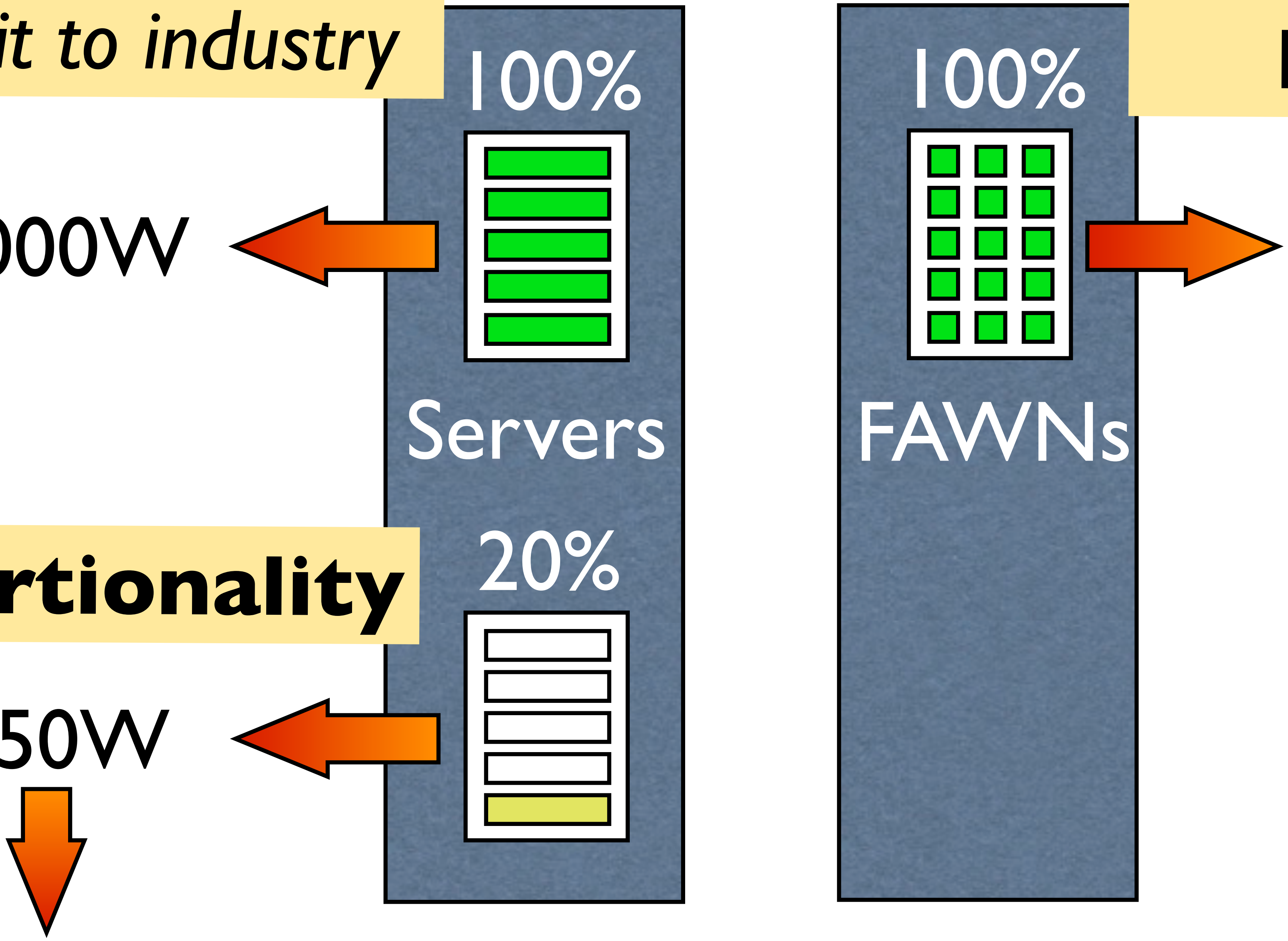
Infrastructure:

PUE

2005: 2–3

2012: ~1.1

Leave it to industry



1000W

Proportionality

750W

200W

Efficiency

300W

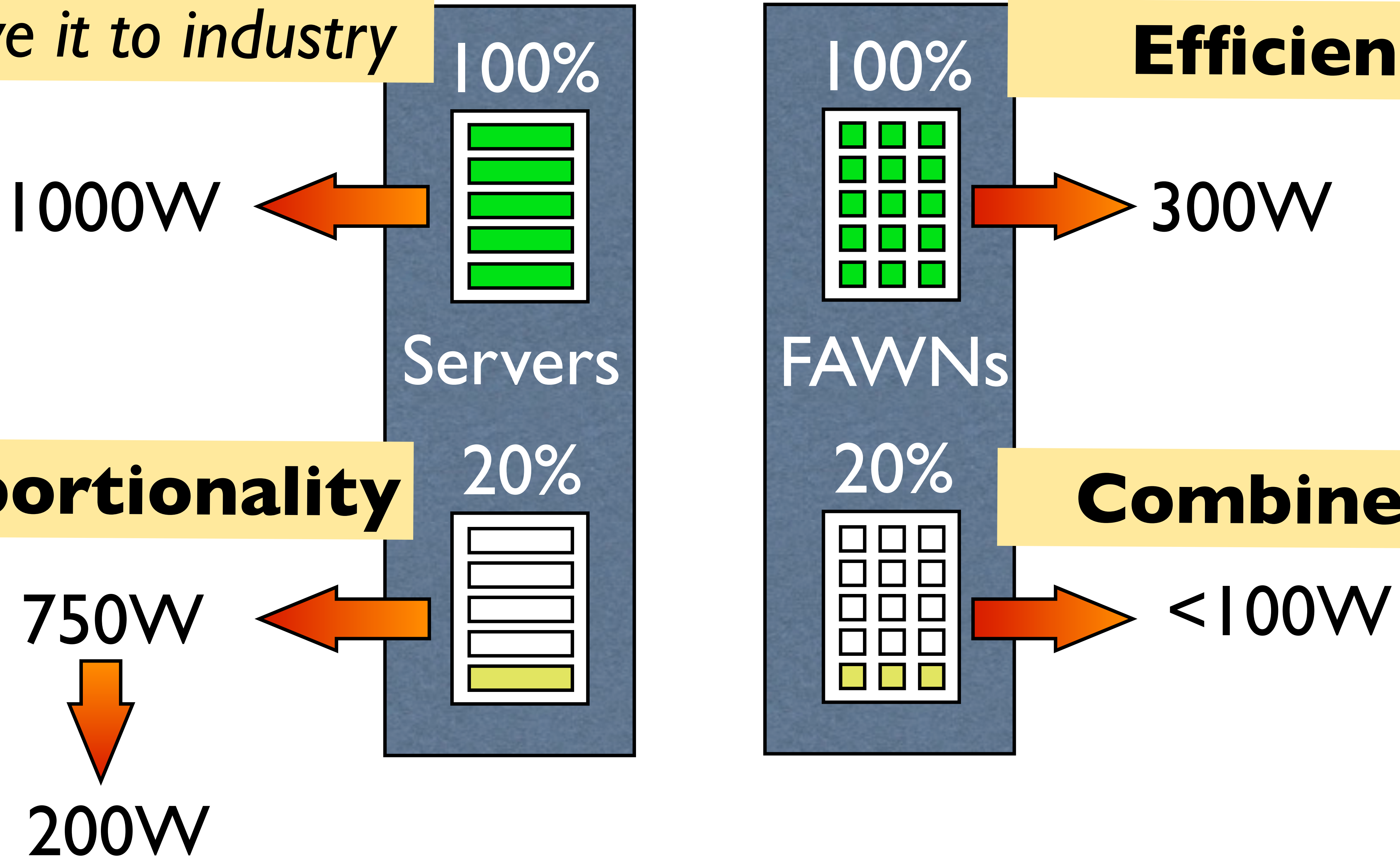
Infrastructure:

PUE

2005: 2–3

2012: ~1.1

Leave it to industry



Efficiency

Proportionality

Combined...

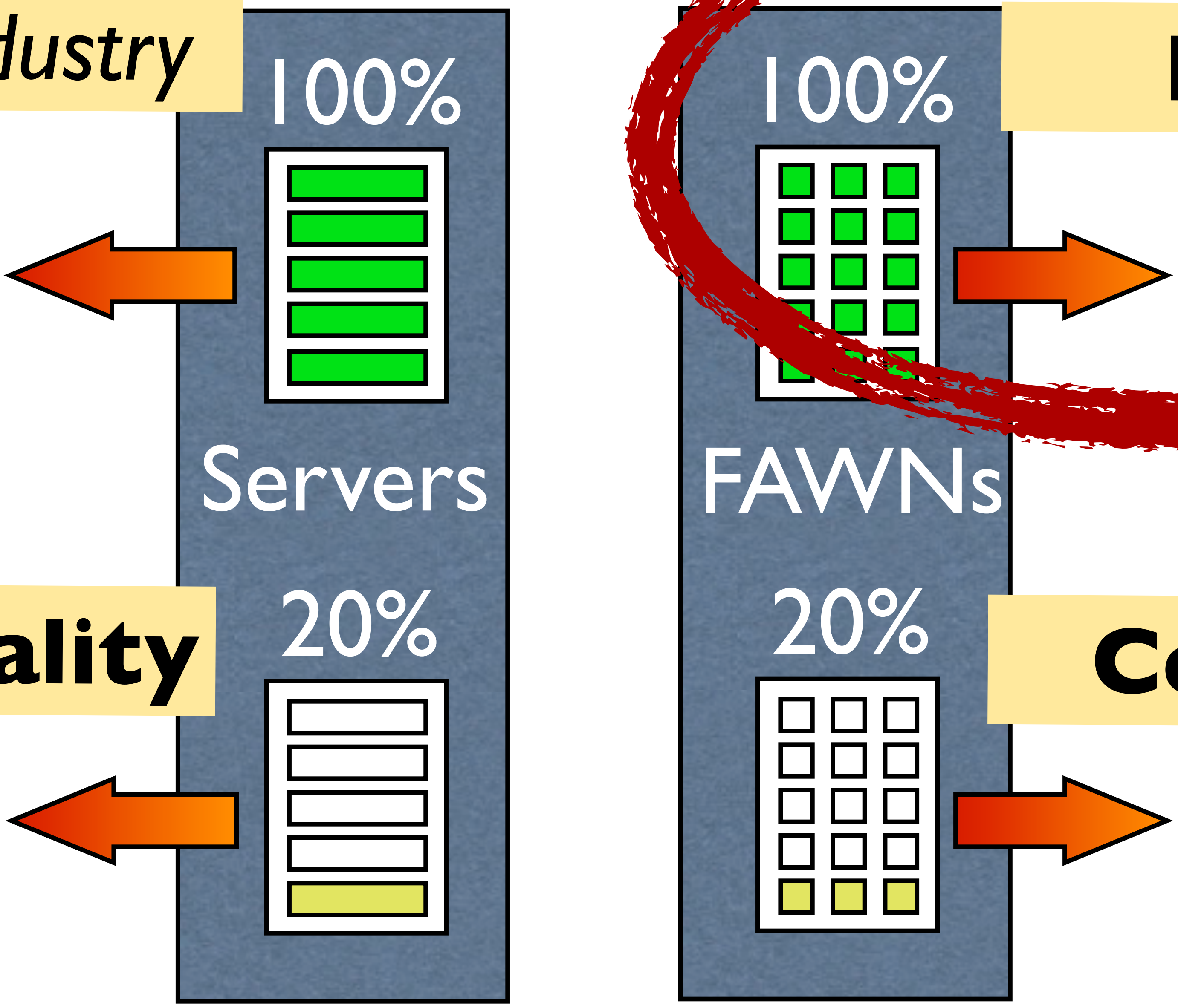
Infrastructure:

PUE

2005: 2–3

2012: ~1.1

Leave it to industry



100%

Servers

20%

100%

FAWNs

20%

1000W

Proportionality

750W

200W

Efficiency

300W

Combined...

<100W

Example:

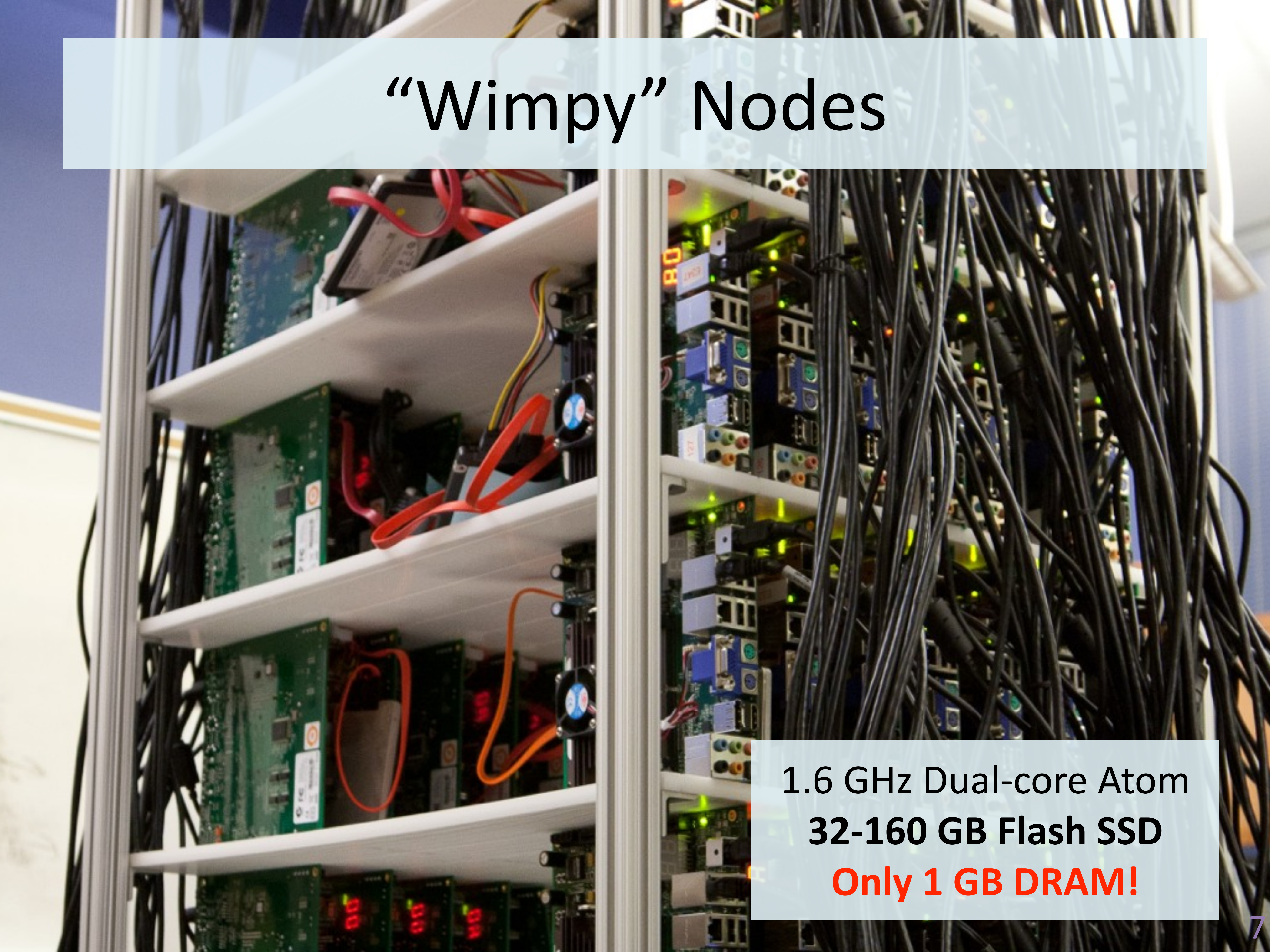
1.6 GHz Atom Z2460

1 GB LPDDR2

Two pillars

- Gigahertz costs twice:
    - Once for the switching speed
    - Once for the memory wall
- Memory capacity costs (at least) once:
    - Longer buses < efficient

“Wimpy” Nodes

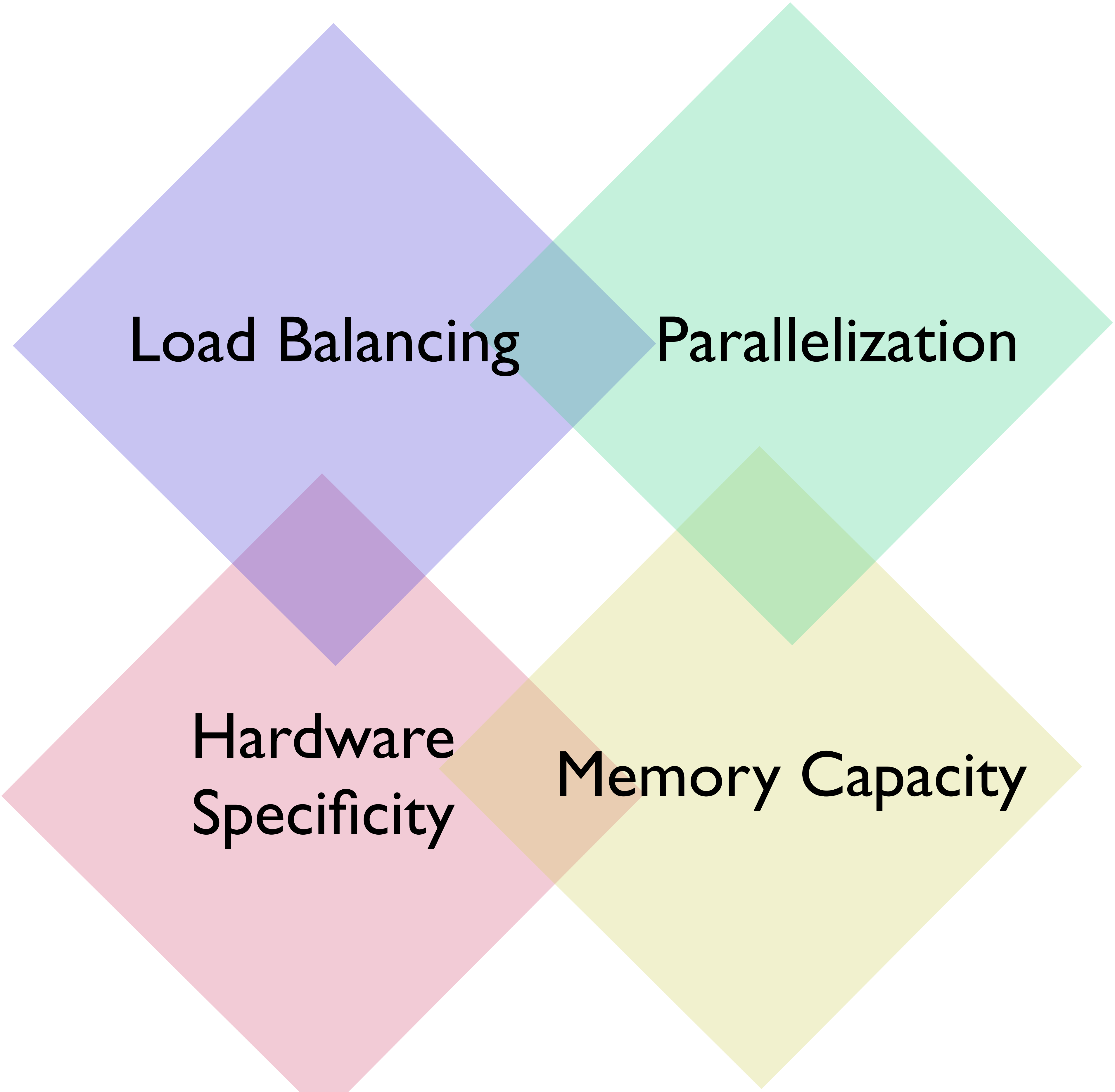


1.6 GHz Dual-core Atom

32-160 GB Flash SSD

Only 1 GB DRAM!

“Each decimal order of magnitude increase in parallelism requires a major redesign and rewrite of parallel code” - Kathy Yelick



Load Balancing

Parallelization

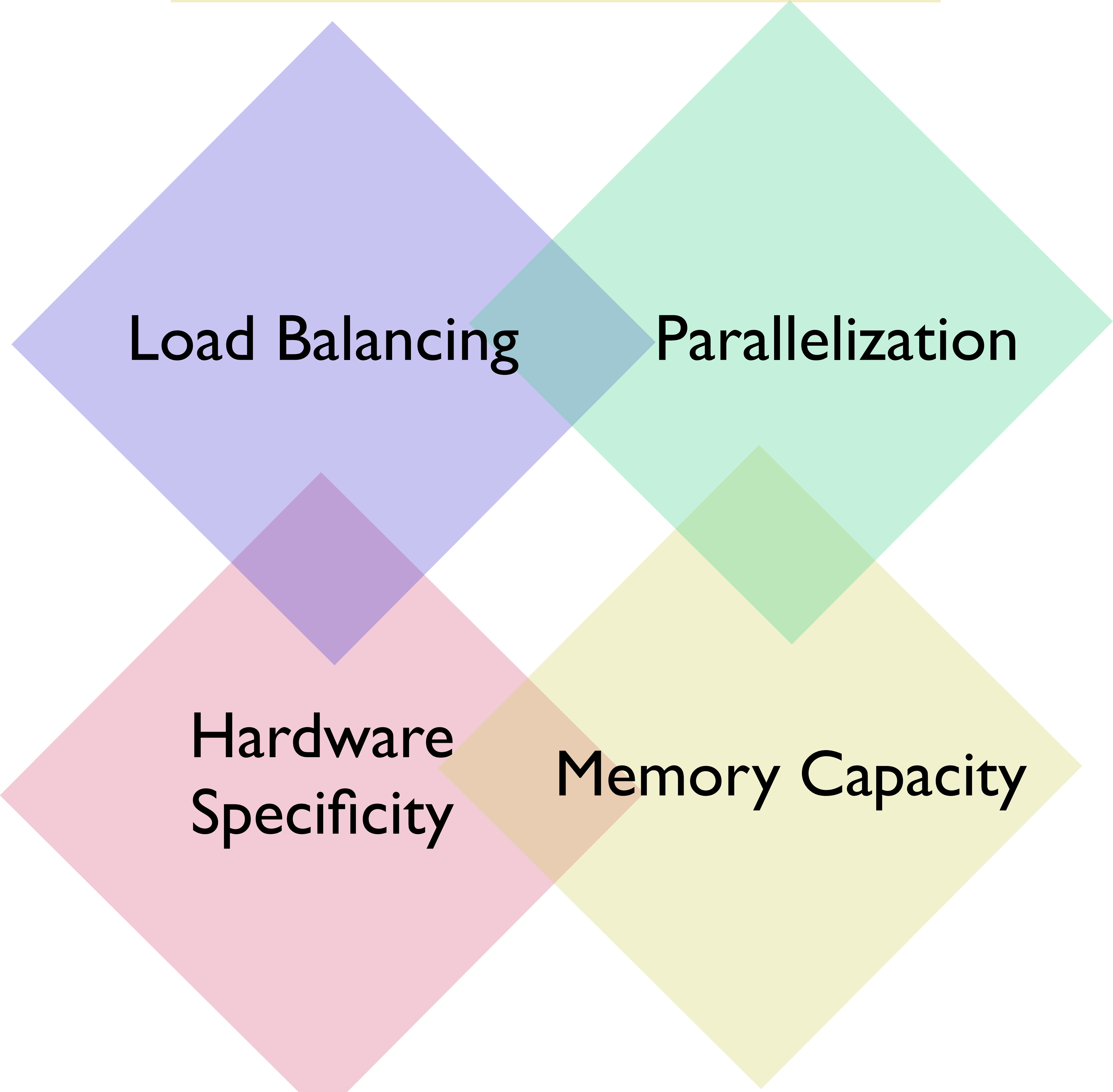
Hardware Specificity

Memory Capacity

Bigger Clusters

Wimpy Nodes

The FAWN Quad of Pain



Load Balancing

Parallelization

Hardware Specificity

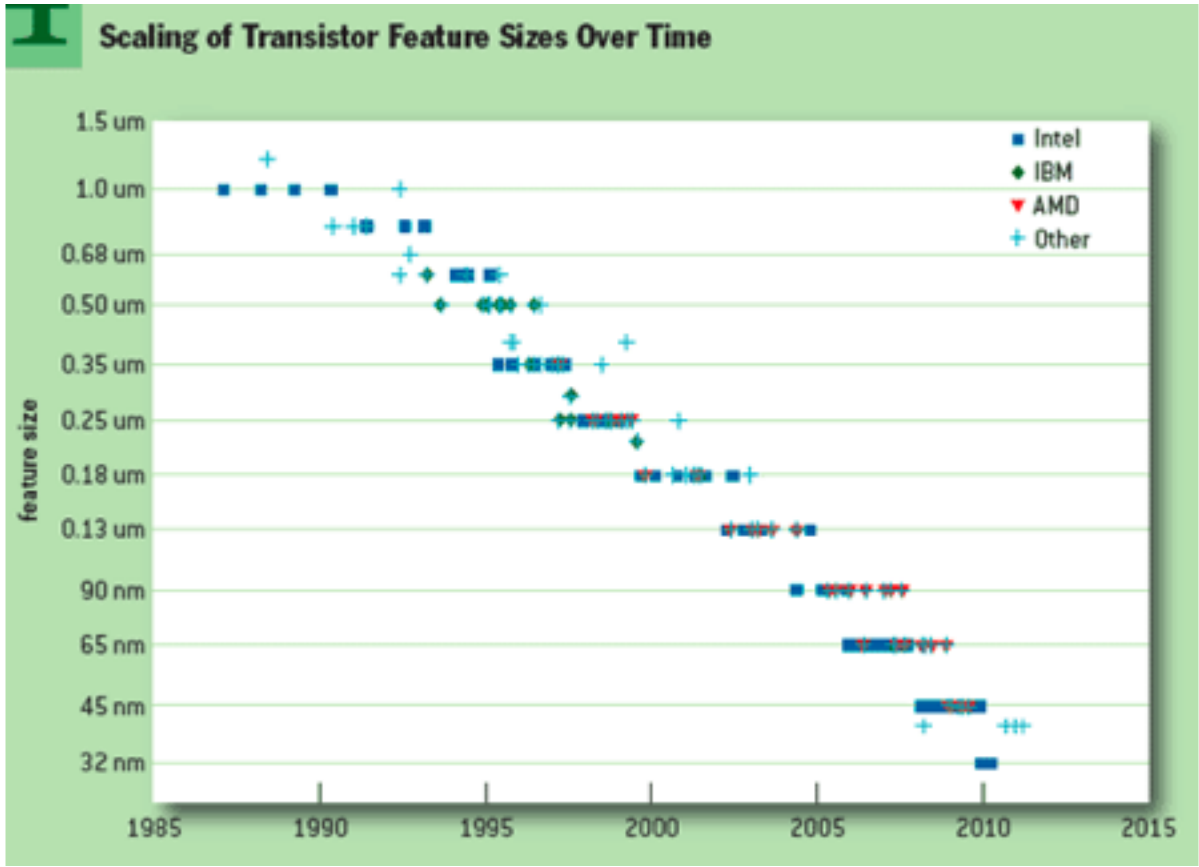
Memory Capacity

Bigger Clusters

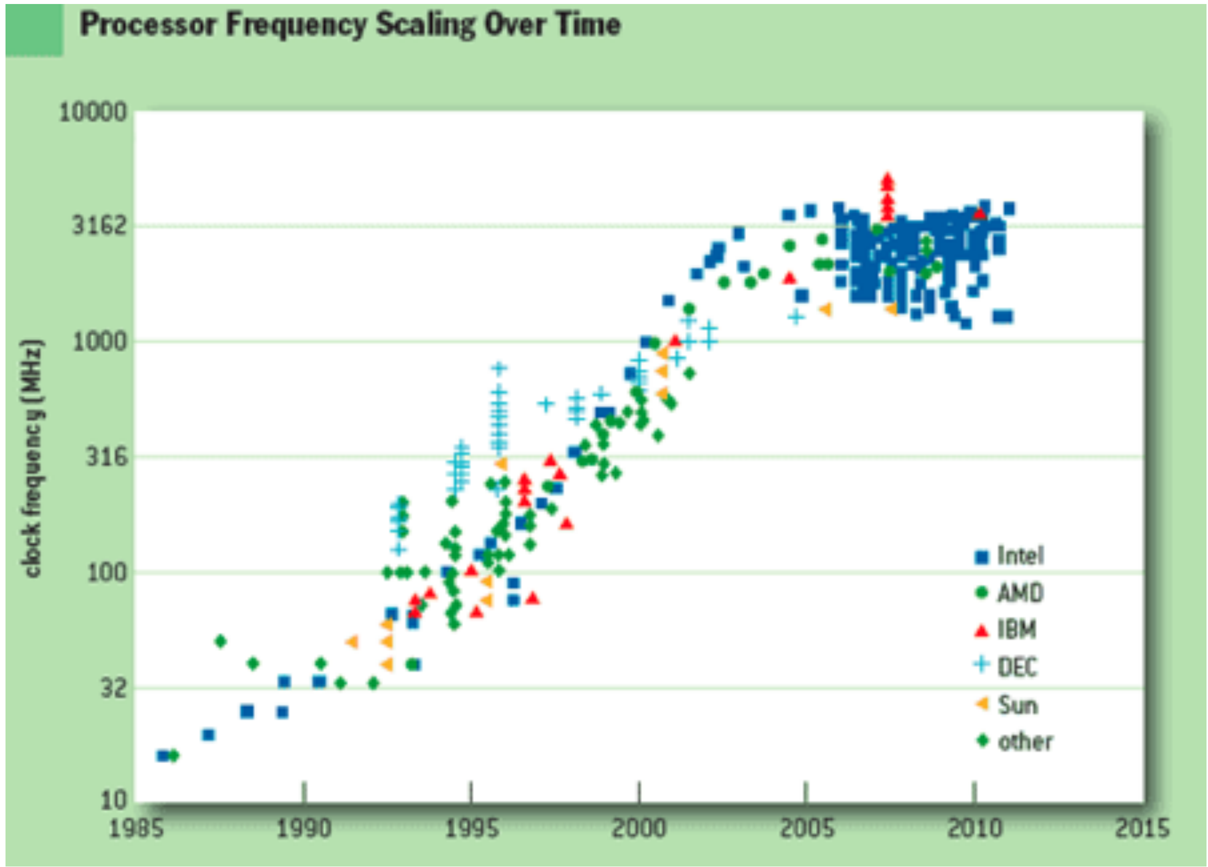
Wimpy Nodes

It's not just masochism

Moore



Dennard

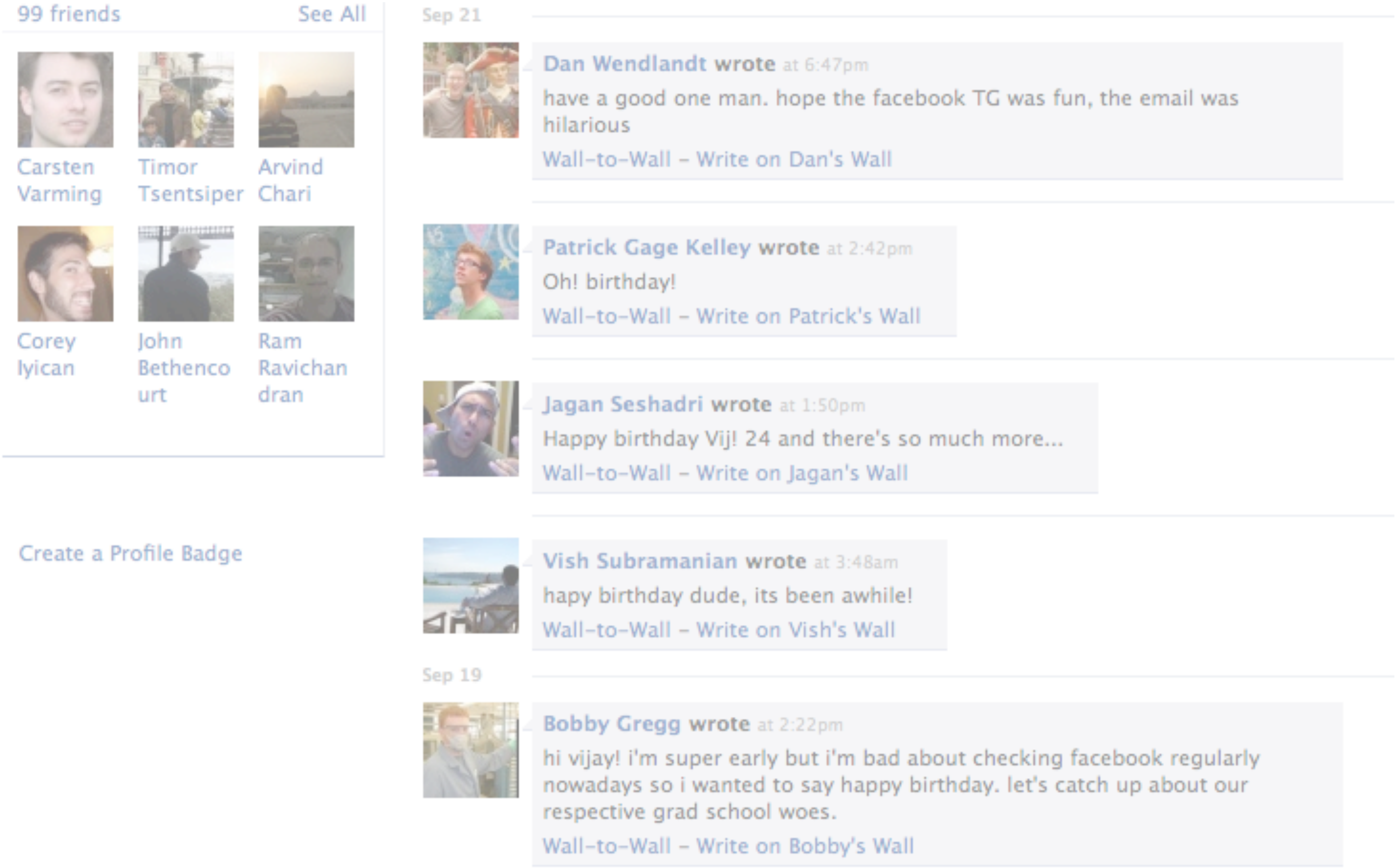


(Figures from Danowitz, Kelley, Mao, Stevenson, and Horowitz: CPU DB)

All systems will face this challenge over time

FAWN:
It started
with a key-value store

Small record, random access



99 friends See All

Carsten Varming
Timor Tsentsiper
Arvind Chari
Corey Iyican
John Bethencourt
Ram Ravichandran

Create a Profile Badge

Sep 21

Dan Wendlandt wrote at 6:47pm
have a good one man. hope the facebook TG was fun, the email was hilarious
Wall-to-Wall - Write on Dan's Wall

Patrick Gage Kelley wrote at 2:42pm
Oh! birthday!
Wall-to-Wall - Write on Patrick's Wall

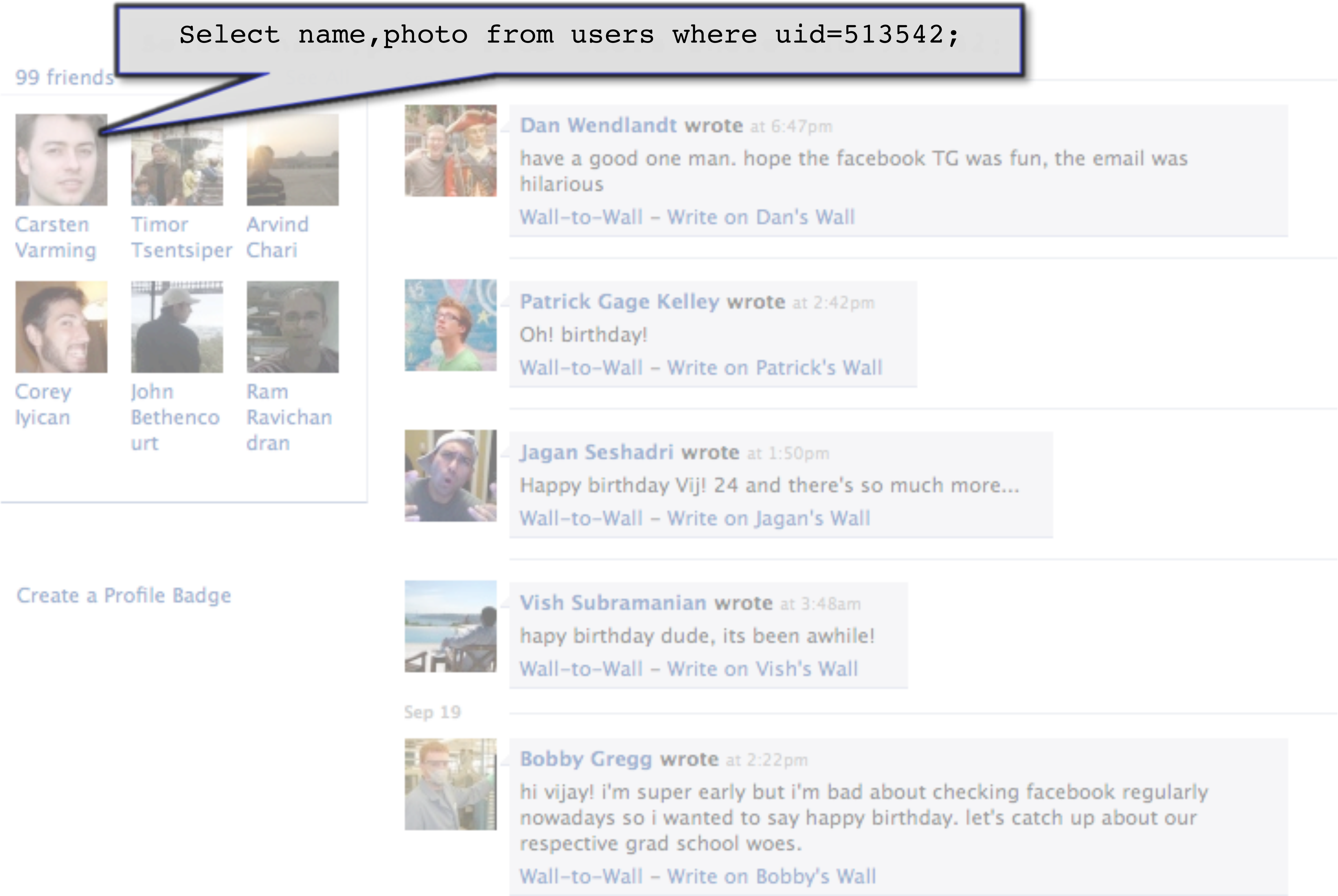
Jagan Seshadri wrote at 1:50pm
Happy birthday Vij! 24 and there's so much more...
Wall-to-Wall - Write on Jagan's Wall

Vish Subramanian wrote at 3:48am
hapy birthday dude, its been awhile!
Wall-to-Wall - Write on Vish's Wall

Sep 19

Bobby Gregg wrote at 2:22pm
hi vijay! i'm super early but i'm bad about checking facebook regularly nowadays so i wanted to say happy birthday. let's catch up about our respective grad school woes.
Wall-to-Wall - Write on Bobby's Wall

Small record, random access



```
Select name,photo from users where uid=513542;
```

99 friends

Carsten Varming

Timor Tsentsiper

Arvind Chari

Corey Iyican

John Bethencourt

Ram Ravichandran

Create a Profile Badge

Dan Wendlandt wrote at 6:47pm
have a good one man. hope the facebook TG was fun, the email was hilarious
Wall-to-Wall - Write on Dan's Wall

Patrick Gage Kelley wrote at 2:42pm
Oh! birthday!
Wall-to-Wall - Write on Patrick's Wall

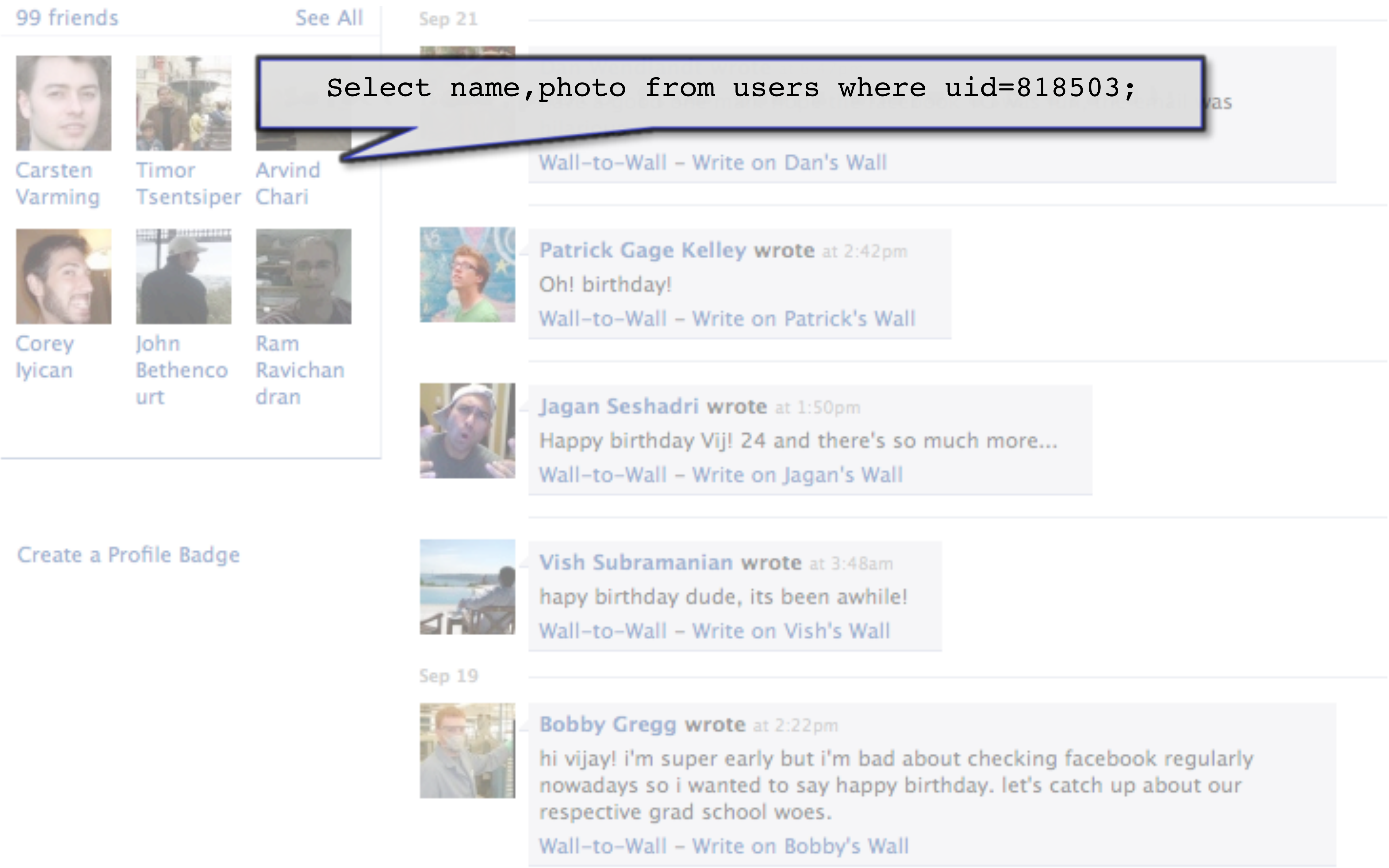
Jagan Seshadri wrote at 1:50pm
Happy birthday Vij! 24 and there's so much more...
Wall-to-Wall - Write on Jagan's Wall

Vish Subramanian wrote at 3:48am
hapy birthday dude, its been awhile!
Wall-to-Wall - Write on Vish's Wall

Sep 19

Bobby Gregg wrote at 2:22pm
hi vijay! i'm super early but i'm bad about checking facebook regularly nowadays so i wanted to say happy birthday. let's catch up about our respective grad school woes.
Wall-to-Wall - Write on Bobby's Wall

Small record, random access



```
Select name,photo from users where uid=818503;
```

99 friends See All

Carsten Varming

Timor Tsentsiper

Arvind Chari

Corey Iyican

John Bethencourt

Ram Ravichandran

Create a Profile Badge

Sep 21

Wall-to-Wall – Write on Dan's Wall

Patrick Gage Kelley wrote at 2:42pm

Oh! birthday!

Wall-to-Wall – Write on Patrick's Wall

Jagan Seshadri wrote at 1:50pm

Happy birthday Vij! 24 and there's so much more...

Wall-to-Wall – Write on Jagan's Wall

Vish Subramanian wrote at 3:48am

hapy birthday dude, its been awhile!

Wall-to-Wall – Write on Vish's Wall

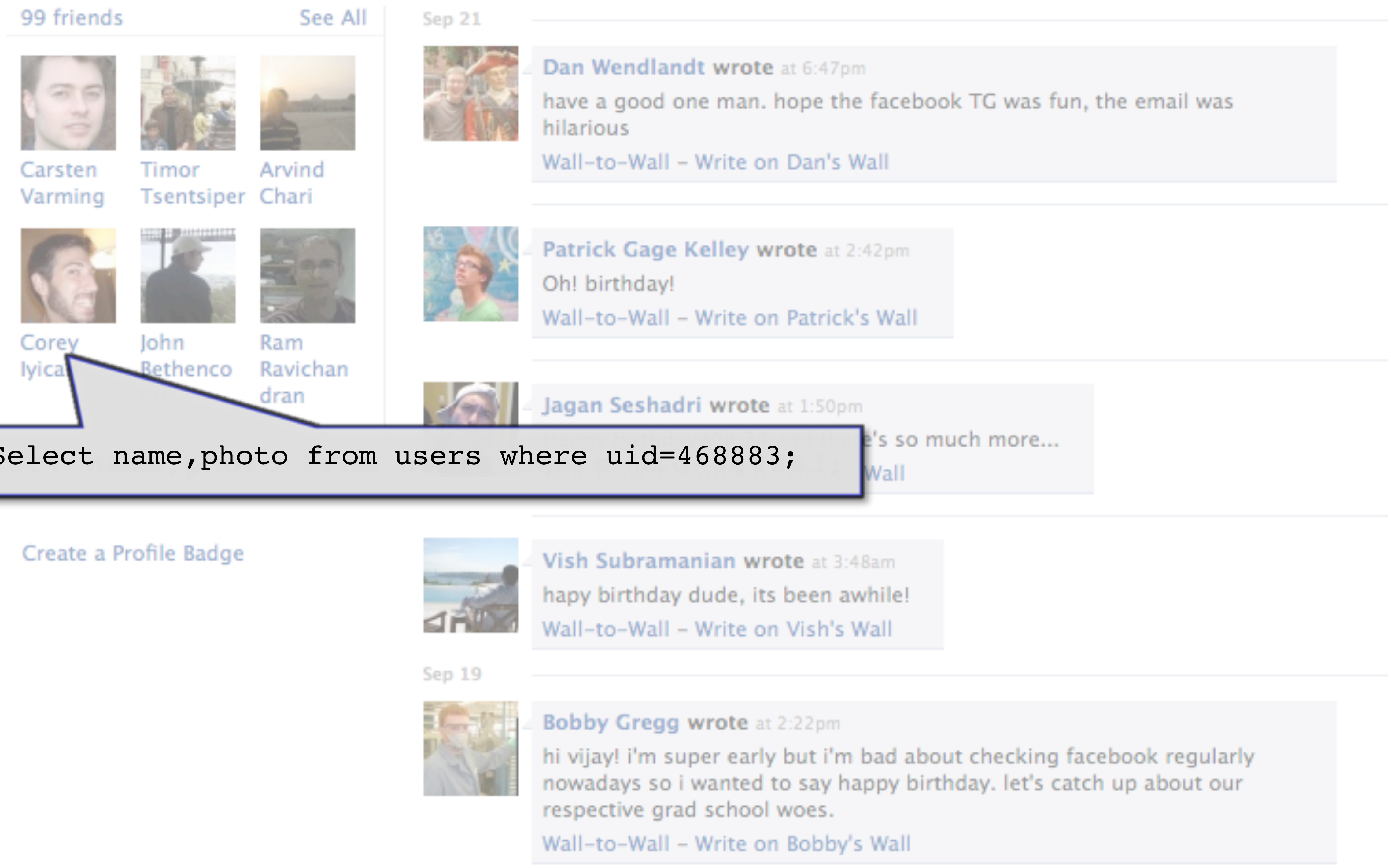
Sep 19

Bobby Gregg wrote at 2:22pm

hi vijay! i'm super early but i'm bad about checking facebook regularly nowadays so i wanted to say happy birthday. let's catch up about our respective grad school woes.

Wall-to-Wall – Write on Bobby's Wall

Small record, random access



99 friends See All

Carsten Varming

Timor Tsentsiper

Arvind Chari

Corey Iyica

John Bethenco

Ram Ravichandran

Create a Profile Badge

Sep 21

Dan Wendlandt wrote at 6:47pm
have a good one man. hope the facebook TG was fun, the email was hilarious
Wall-to-Wall - Write on Dan's Wall

Patrick Gage Kelley wrote at 2:42pm
Oh! birthday!
Wall-to-Wall - Write on Patrick's Wall

Jagan Seshadri wrote at 1:50pm
's so much more...
Wall

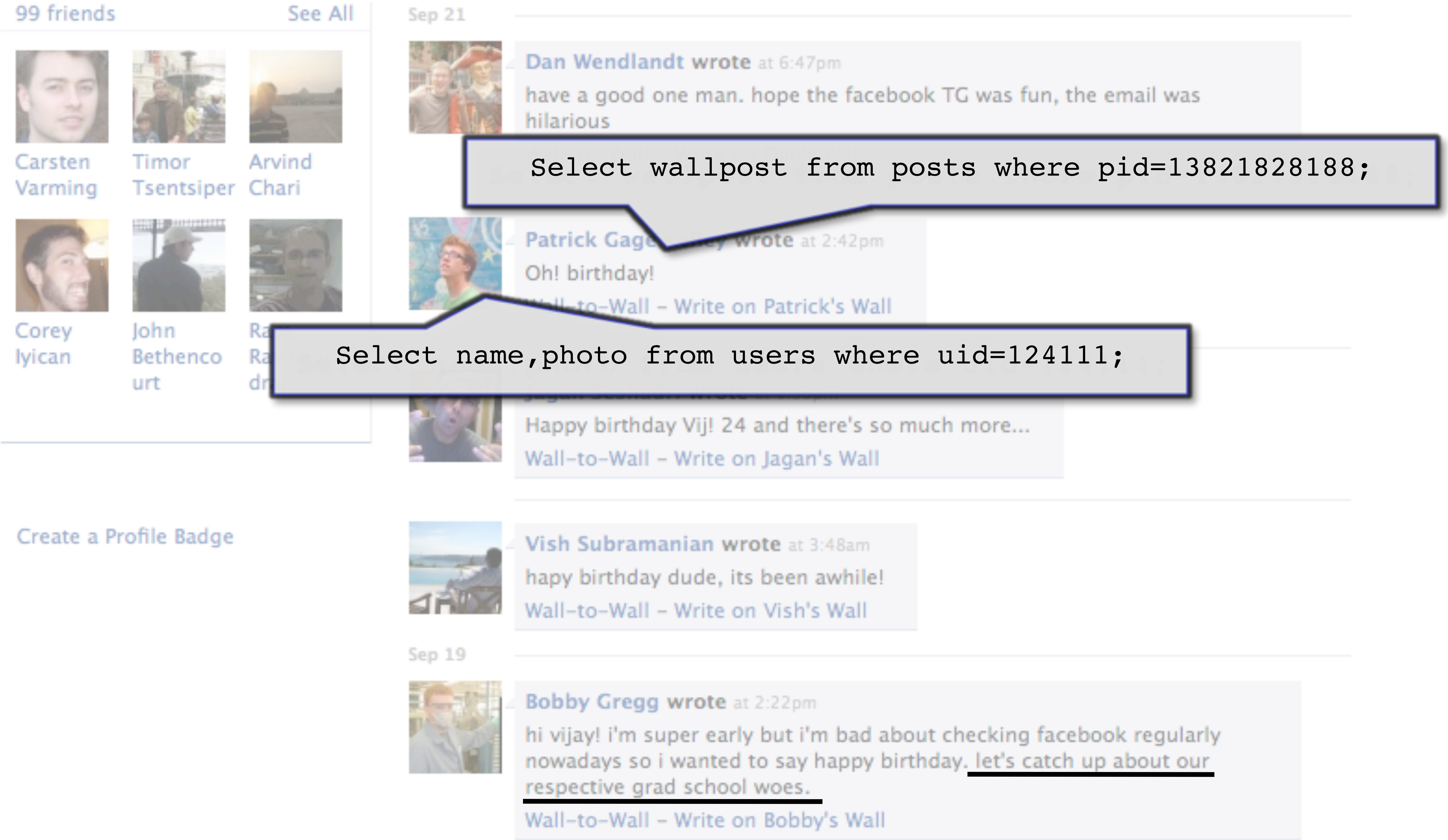
Vish Subramanian wrote at 3:48am
hapy birthday dude, its been awhile!
Wall-to-Wall - Write on Vish's Wall

Sep 19

Bobby Gregg wrote at 2:22pm
hi vijay! i'm super early but i'm bad about checking facebook regularly nowadays so i wanted to say happy birthday. let's catch up about our respective grad school woes.
Wall-to-Wall - Write on Bobby's Wall

```
Select name,photo from users where uid=468883;
```

Small record, random access



99 friends See All

Carsten Varming

Timor Tsentsiper

Arvind Chari

Corey Iyican

John Bethencourt

Ra Ra dr

Create a Profile Badge

Sep 21

Dan Wendlandt wrote at 6:47pm

have a good one man. hope the facebook TG was fun, the email was hilarious

Patrick Gage ... wrote at 2:42pm

Oh! birthday!

Wall-to-Wall – Write on Patrick's Wall

Happy birthday Vij! 24 and there's so much more...

Wall-to-Wall – Write on Jagan's Wall

Vish Subramanian wrote at 3:48am

hapy birthday dude, its been awhile!

Wall-to-Wall – Write on Vish's Wall

Sep 19

Bobby Gregg wrote at 2:22pm

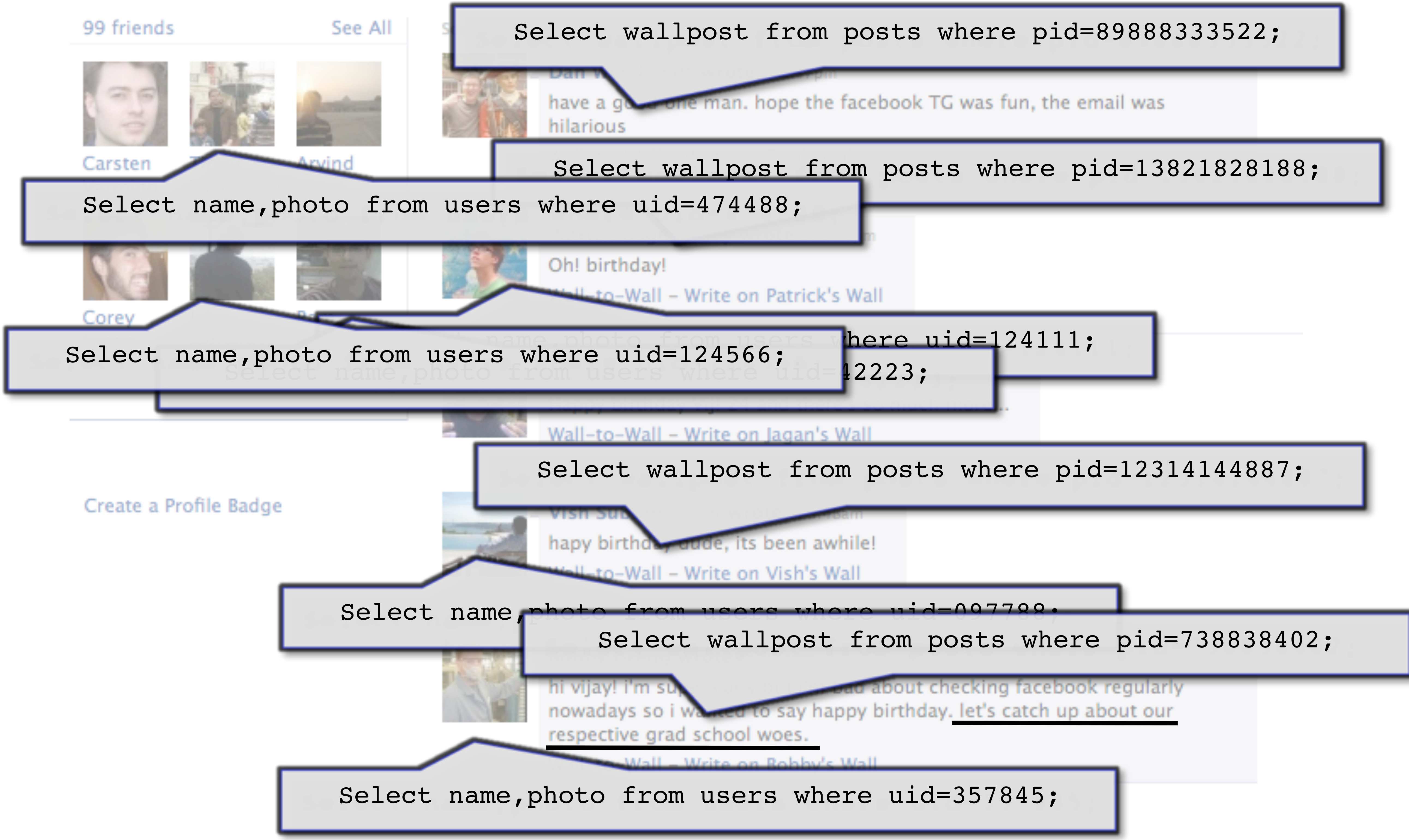
hi vijay! i'm super early but i'm bad about checking facebook regularly nowadays so i wanted to say happy birthday. let's catch up about our respective grad school woes.

Wall-to-Wall – Write on Bobby's Wall

```
Select wallpost from posts where pid=13821828188;
```

```
Select name,photo from users where uid=124111;
```

Small record, random access



have a good one man. hope the facebook TG was fun, the email was hilarious

Oh! birthday!

Wall-to-Wall – Write on Patrick's Wall

Wall-to-Wall – Write on Jagan's Wall

hapy birthday dude, its been awhile!

Wall-to-Wall – Write on Vish's Wall

hi vijay! i'm su... about checking facebook regularly nowadays so i wanted to say happy birthday. let's catch up about our respective grad school woes.

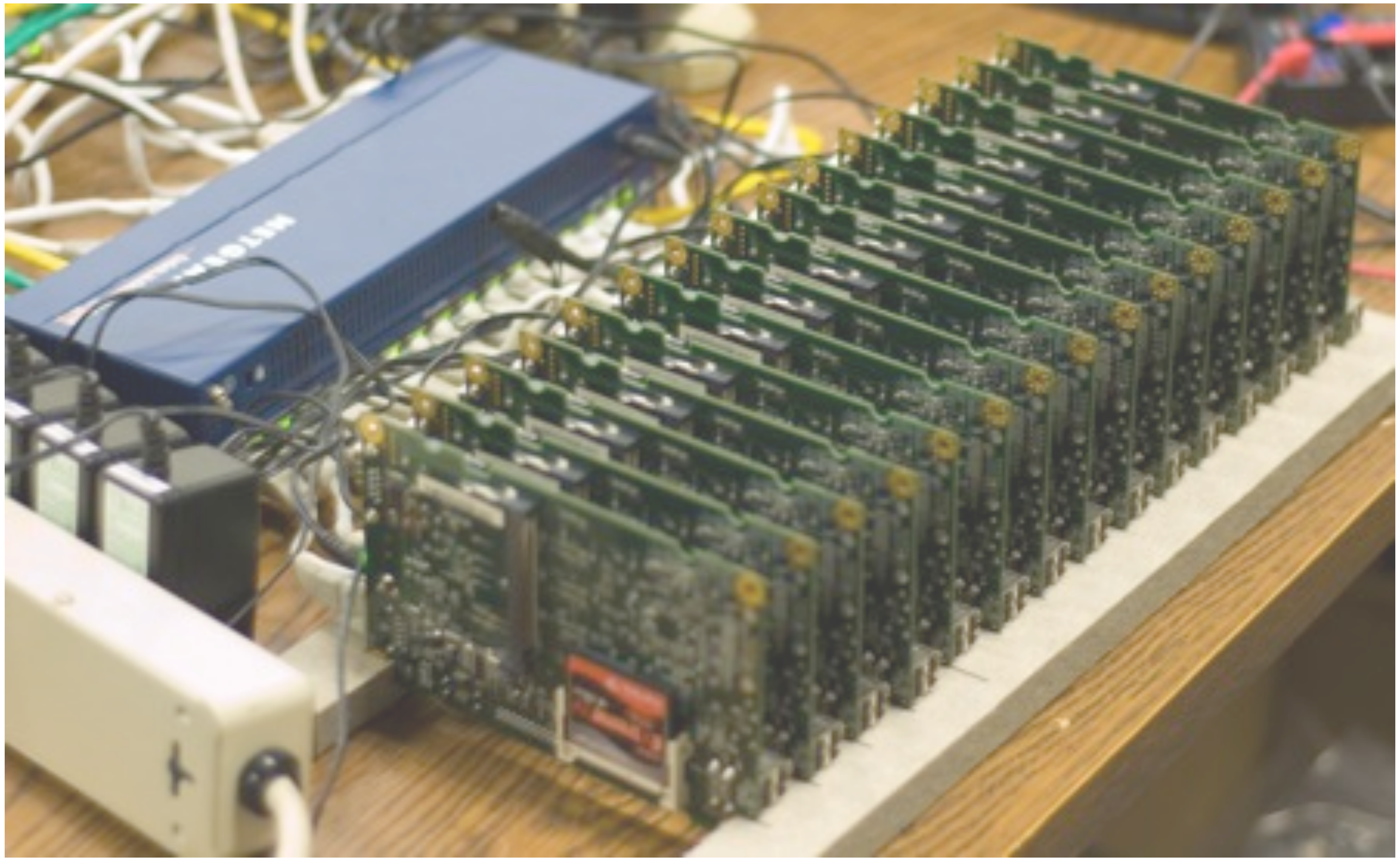
Wall-to-Wall – Write on Robby's Wall

```
Select wallpost from posts where pid=89888333522;
Select wallpost from posts where pid=13821828188;
Select name,photo from users where uid=474488;
Select name,photo from users where uid=124111;
Select name,photo from users where uid=124566;
Select name,photo from users where uid=42223;
Select wallpost from posts where pid=12314144887;
Select name,photo from users where uid=097788;
Select wallpost from posts where pid=738838402;
Select name,photo from users where uid=357845;
```

FAWN-DS

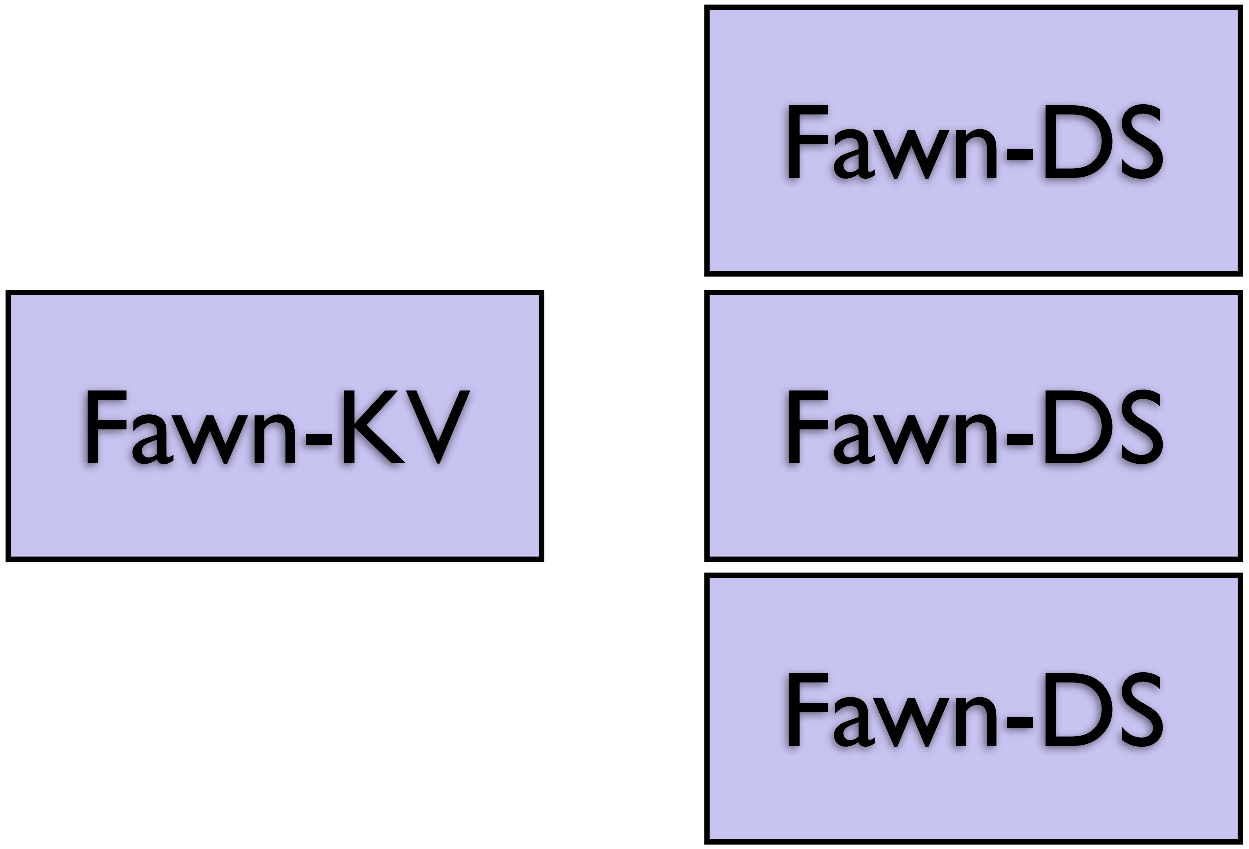
key-value backend store
one node
optimized for wimpy
nodes and flash

Fawn-DS



FAWN-DS FAWN-KV

A cluster-distributed
key-value store
Minimizes work on churn

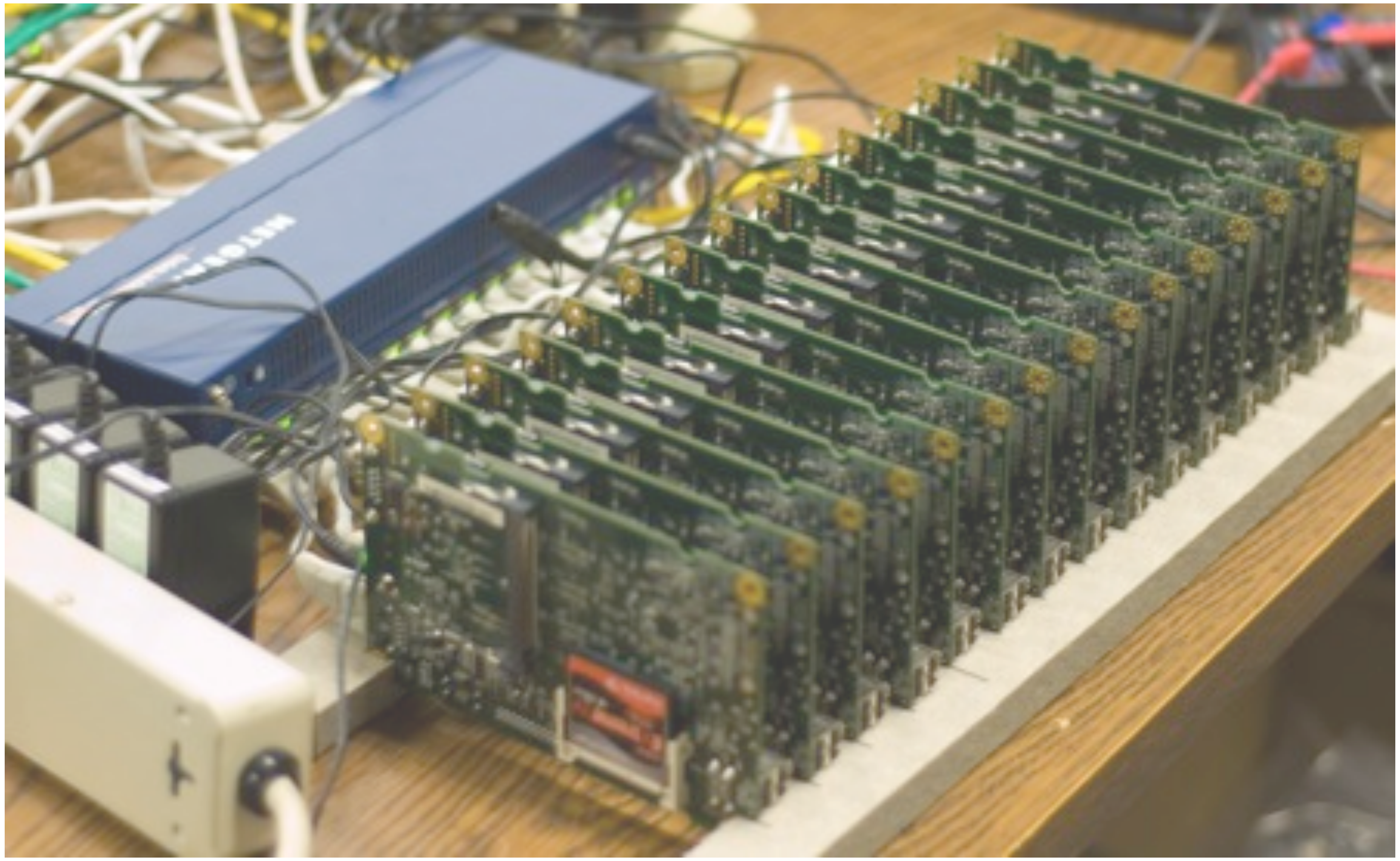


Fawn-KV

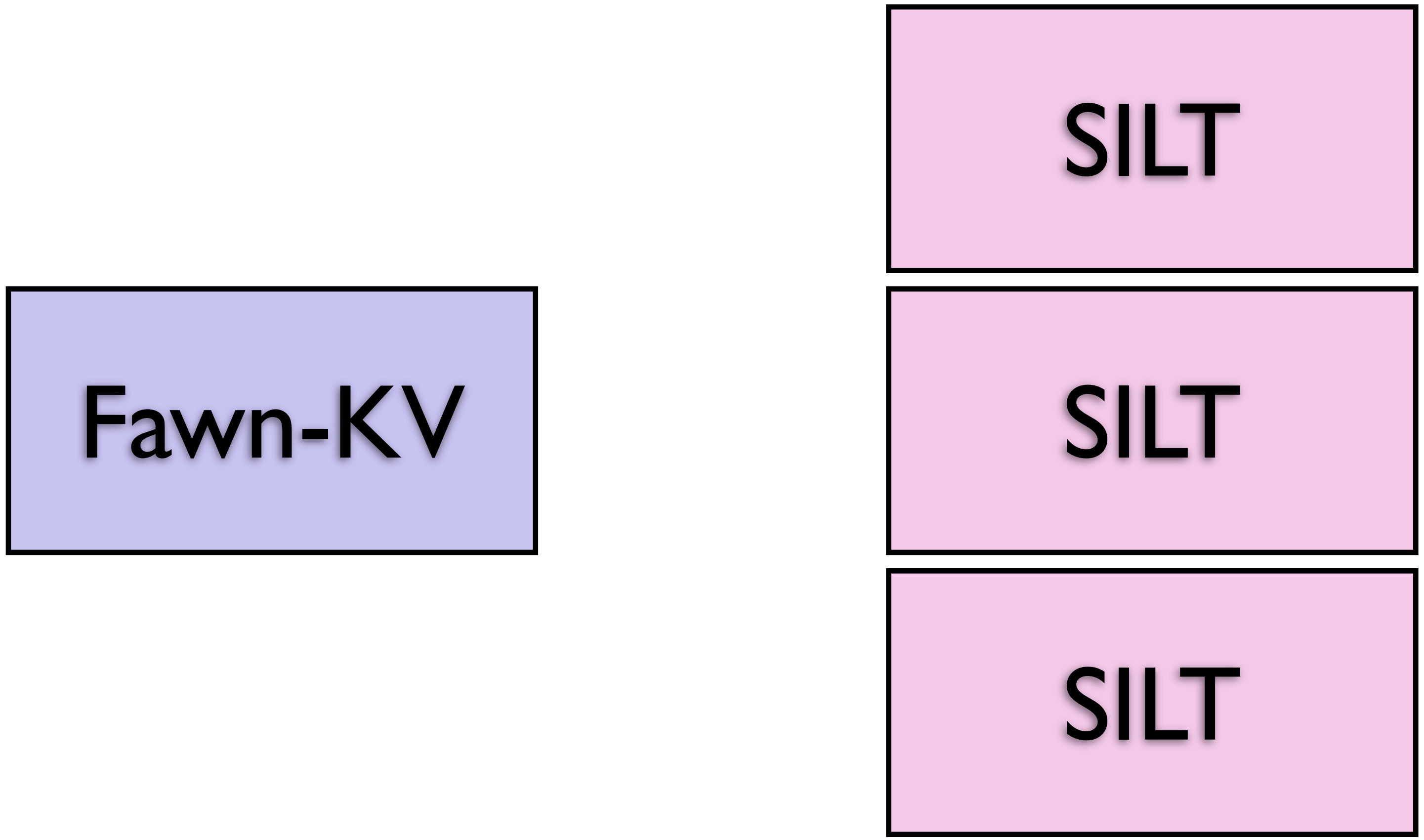
Fawn-DS

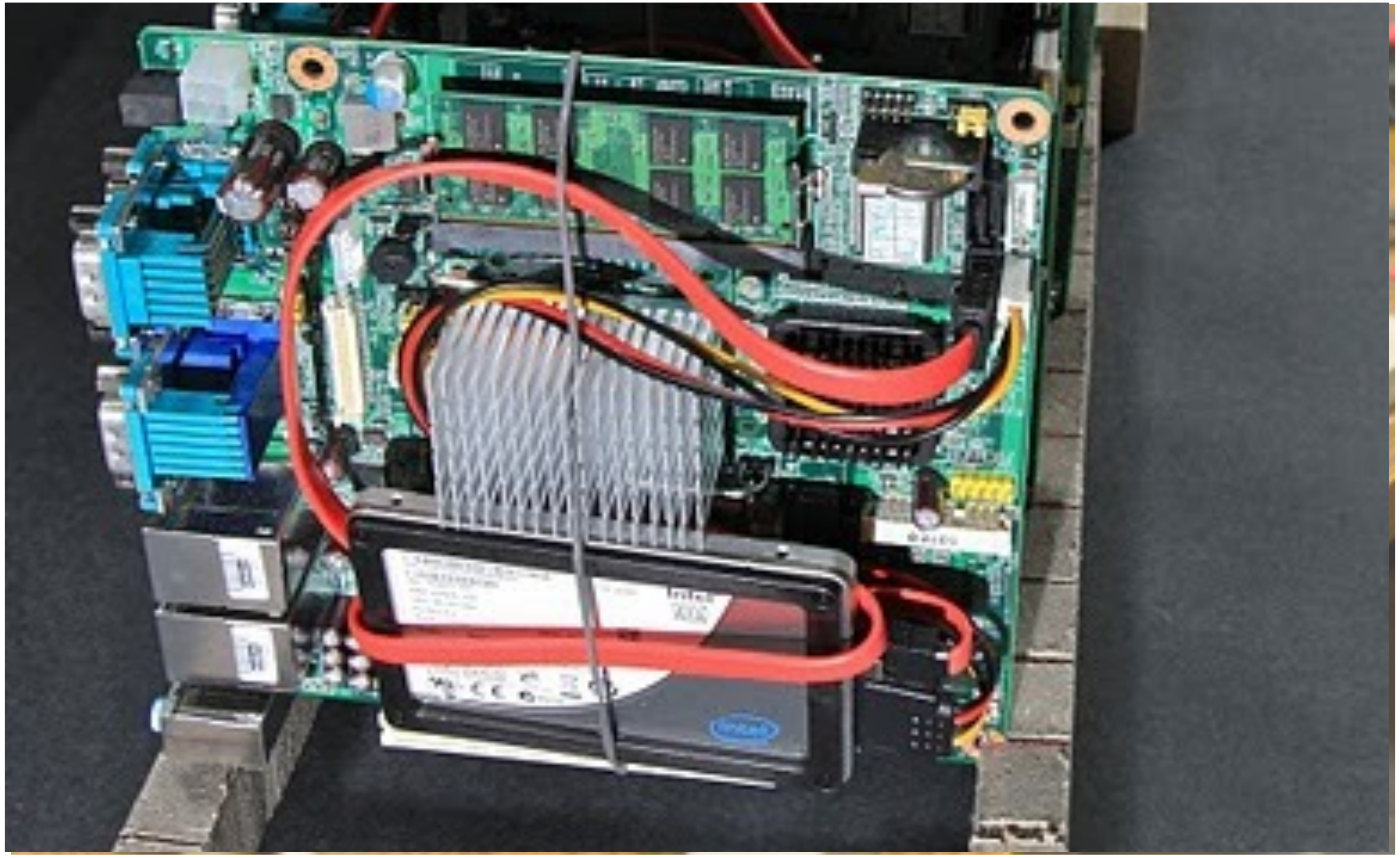
Fawn-DS

Fawn-DS

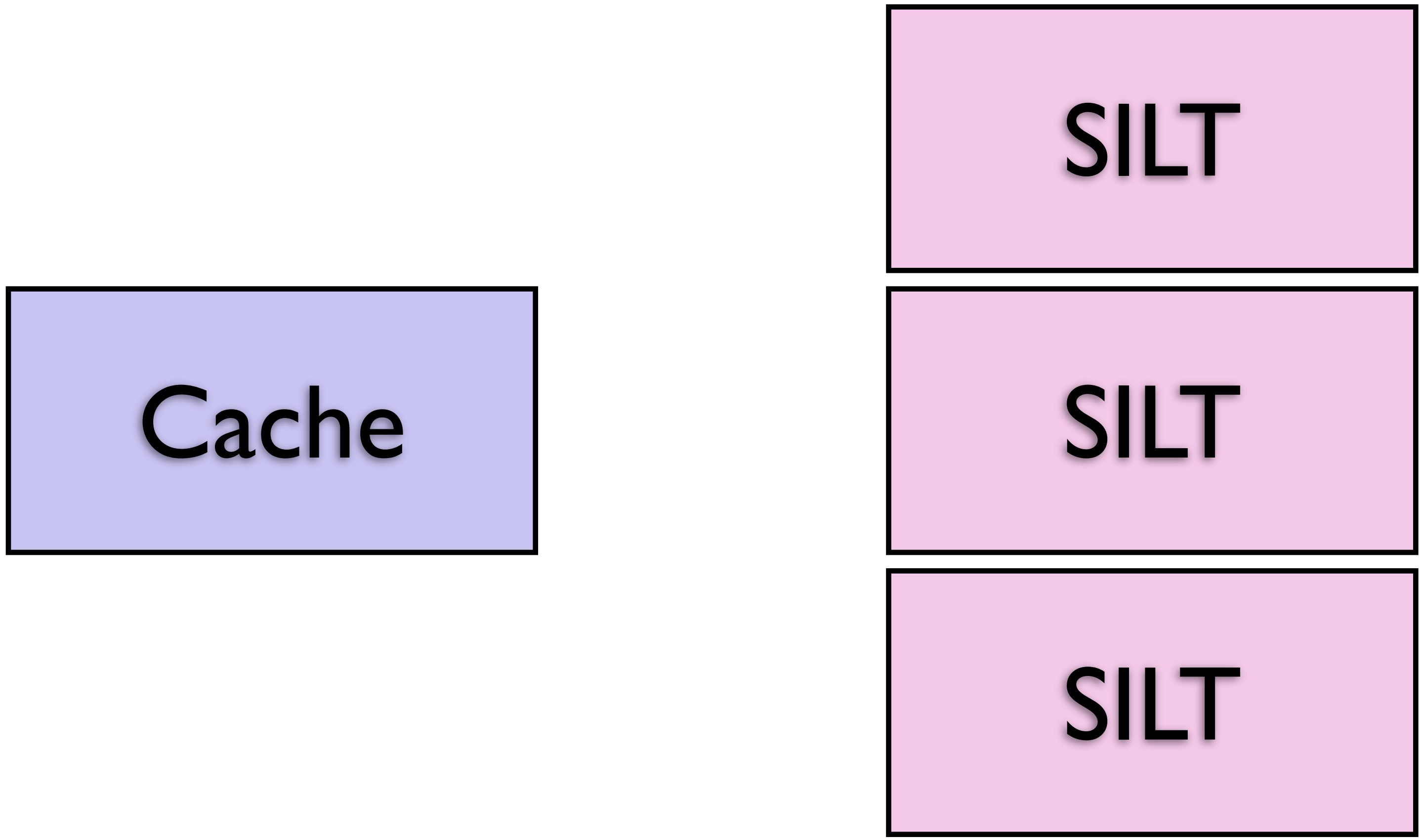


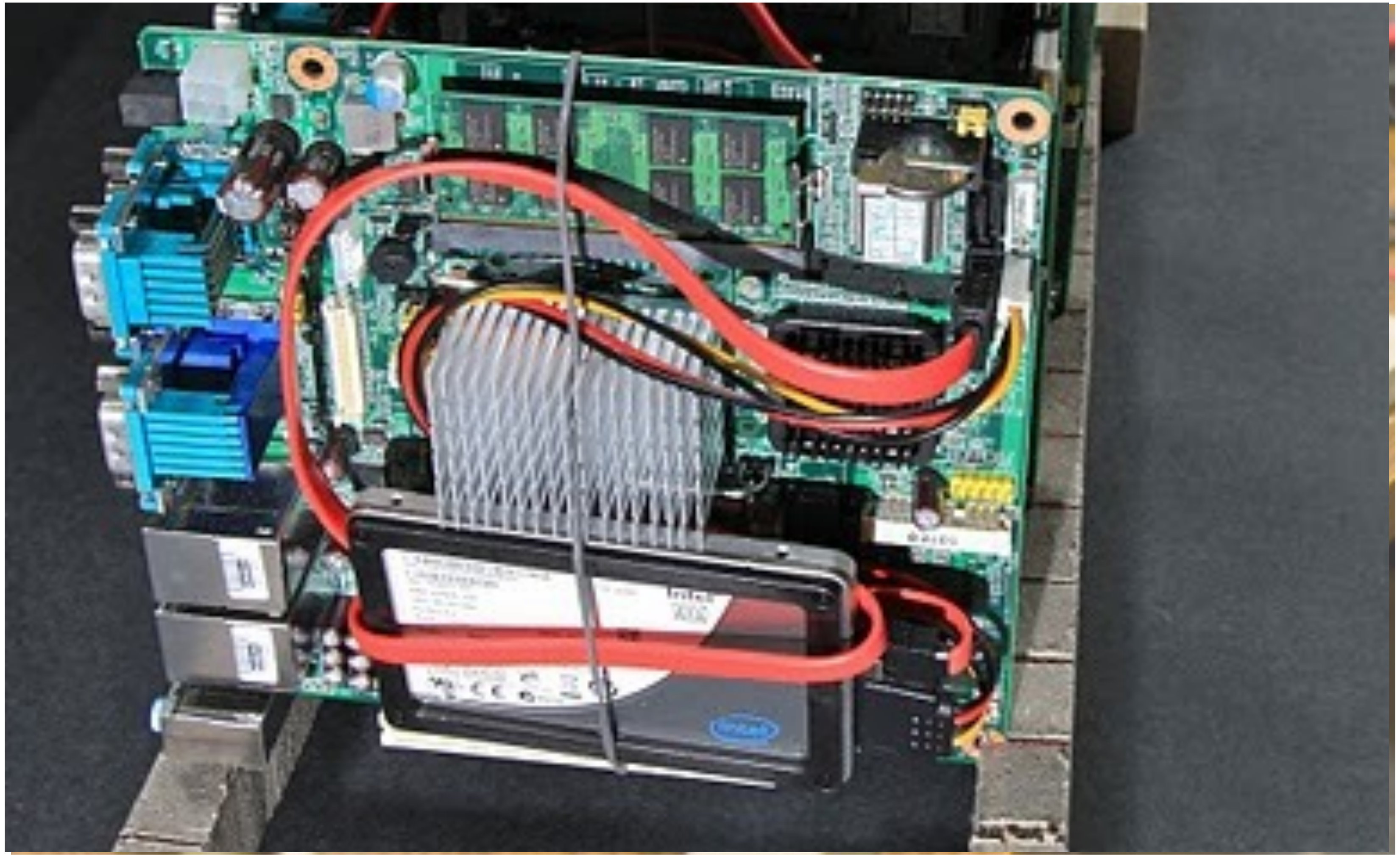
backend store
hyper-optimized
for low DRAM
and large flash



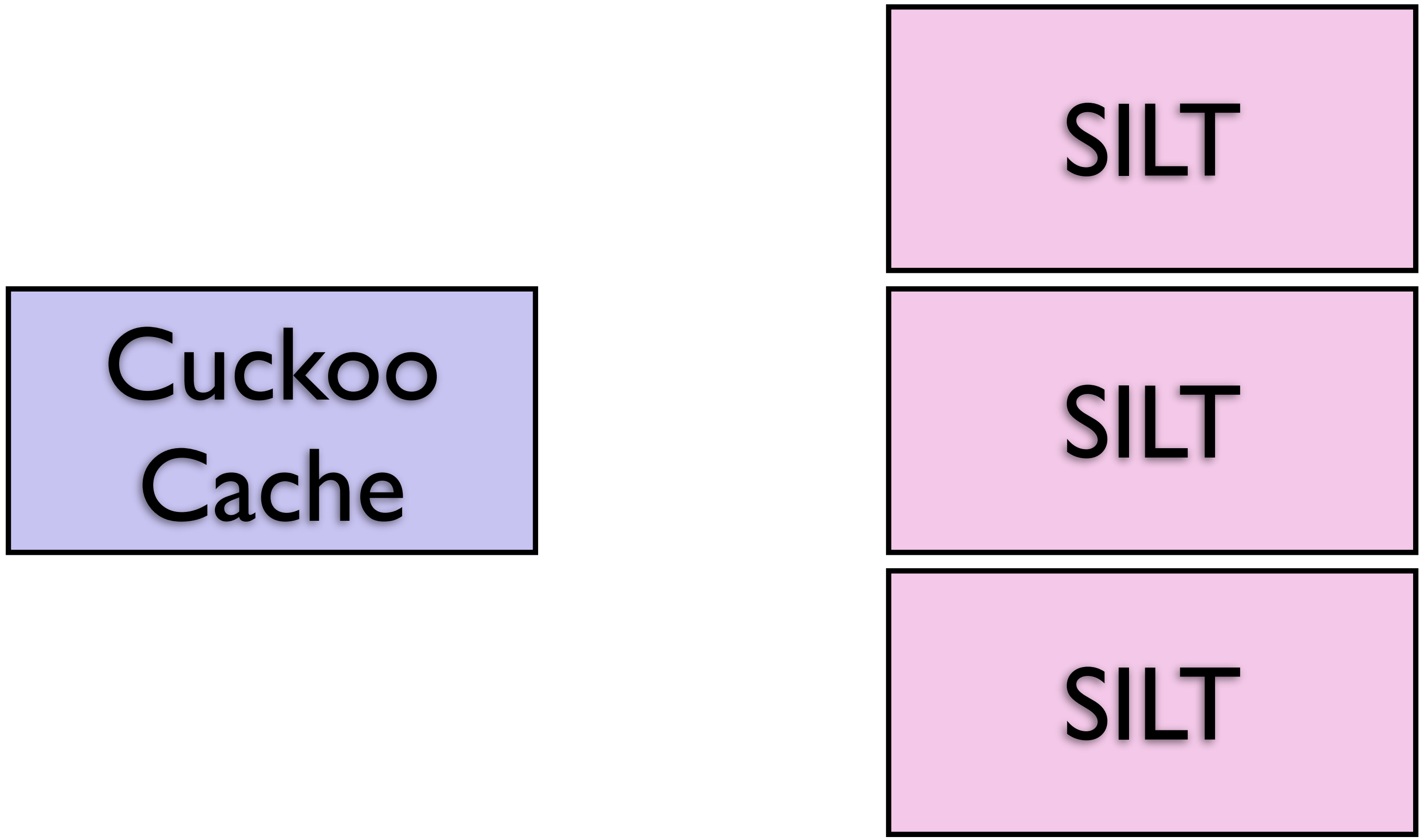


Provable load balancing using a tiny cache





Parallel, fast, memory-efficient memcached using *optimistic cuckoo hashing*

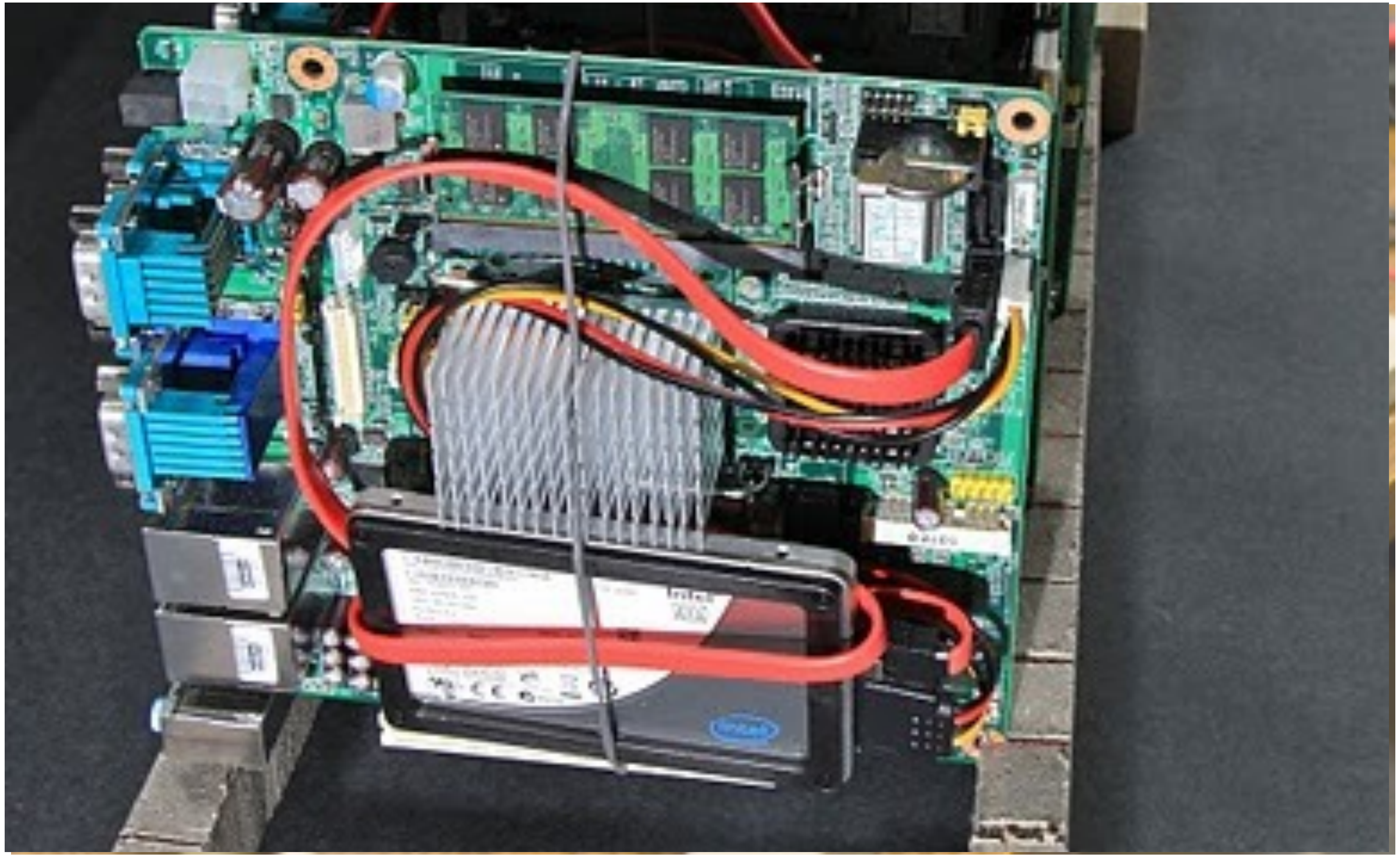


Cuckoo Cache

SILT

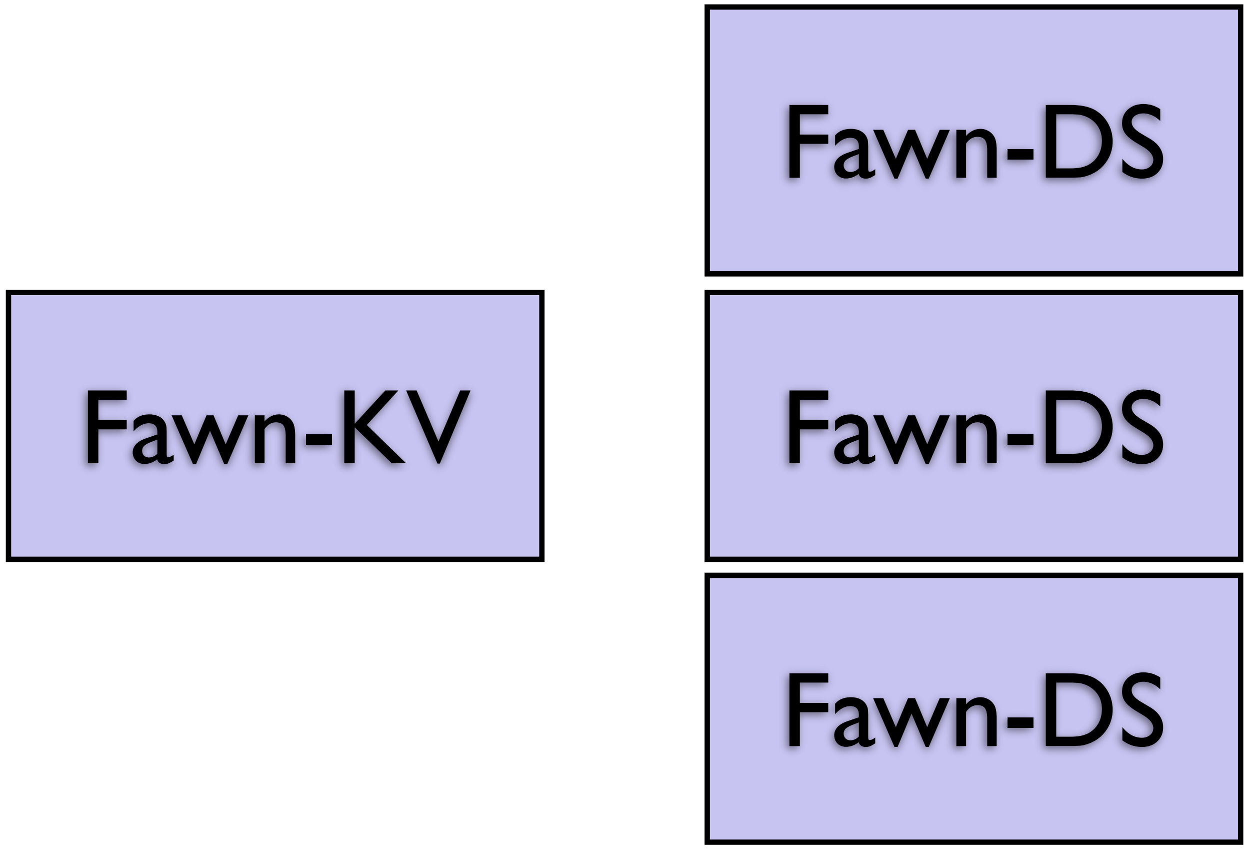
SILT

SILT



After first victory, moved to Atom+SSD

Geode 500Mhz	256MB	4GB CF Card ~2k IOPS
6x	8x	30-60x
Atom 1.6 Ghz single-core	2GB	120GB SSD ~60k IOPS

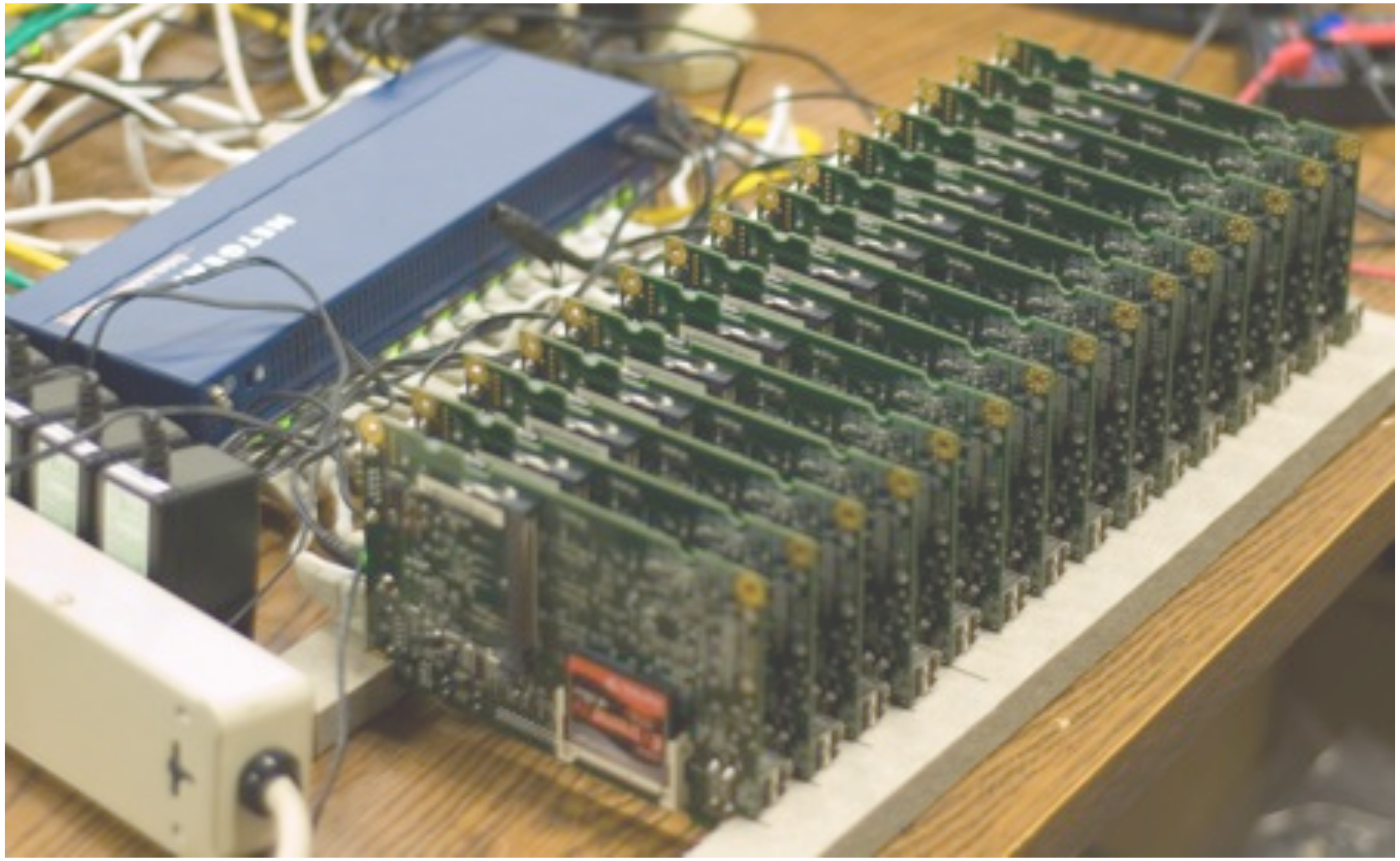


Fawn-DS

Fawn-KV

Fawn-DS

Fawn-DS



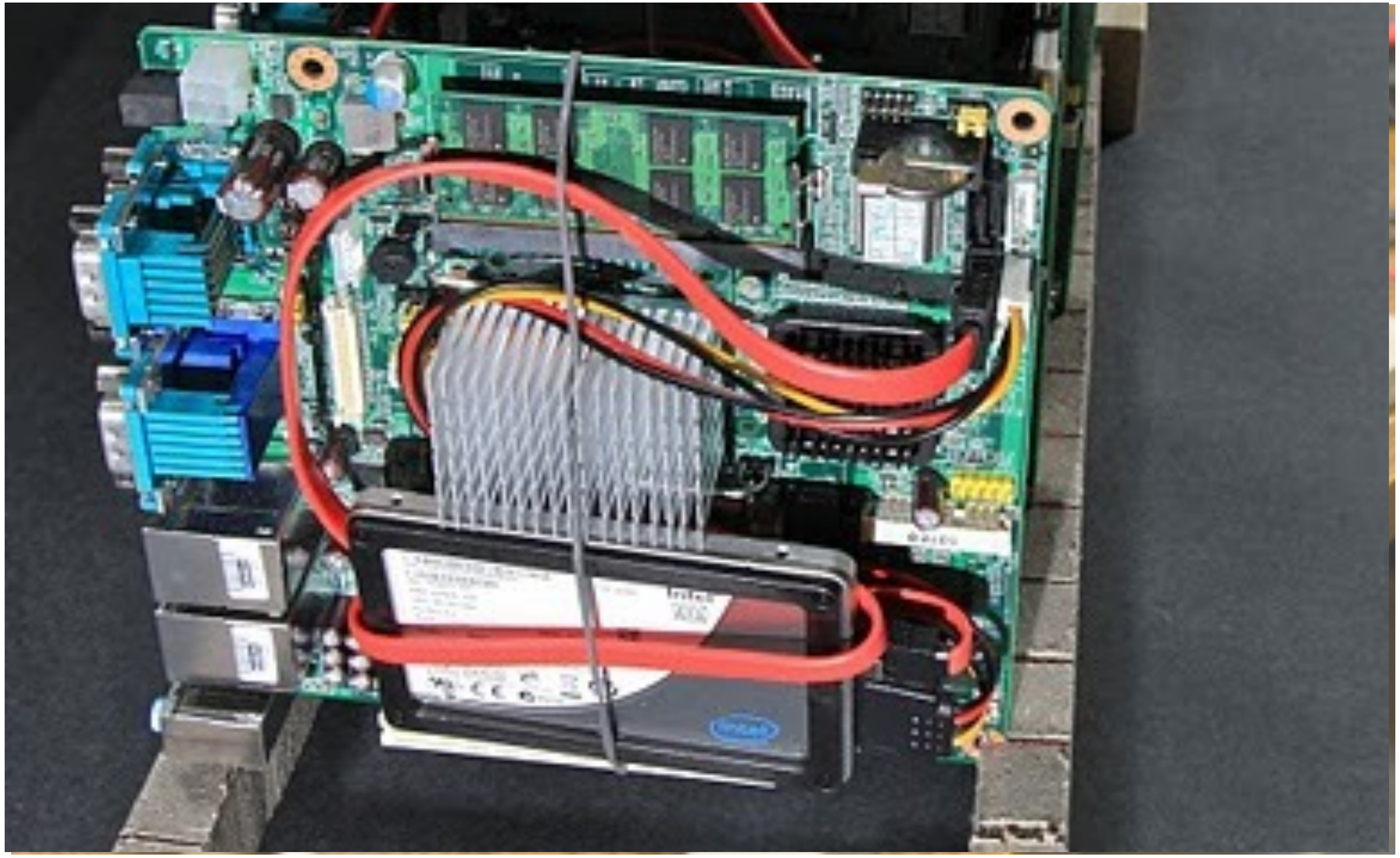
backend store
hyper-optimized
for low DRAM
and large flash

Fawn-KV

SILT

SILT

SILT



Flash Must be Used Carefully

Random reads / sec	48,000

→ Fast, but not THAT fast

Flash Must be Used Carefully

Random reads / sec	48,000

→ Fast, but not THAT fast

$ / GB	1.83

→ Space is precious

Flash Must be Used Carefully

Random reads / sec	48,000

→ Fast, but not THAT fast

$ / GB	1.83

→ Space is precious

Another long-standing problem:

random writes are slow and bad for flash life (wearout)

Three Metrics to Minimize

Memory overhead = Index size per entry

- Ideally 0 bytes/entry (no memory overhead)

Three Metrics to Minimize

Memory overhead = Index size per entry

- Ideally 0 bytes/entry (no memory overhead)

Read amplification = Flash reads per query

- Limits **query throughput**
- Ideally 1 (no wasted flash reads)

Three Metrics to Minimize

Memory overhead = Index size per entry

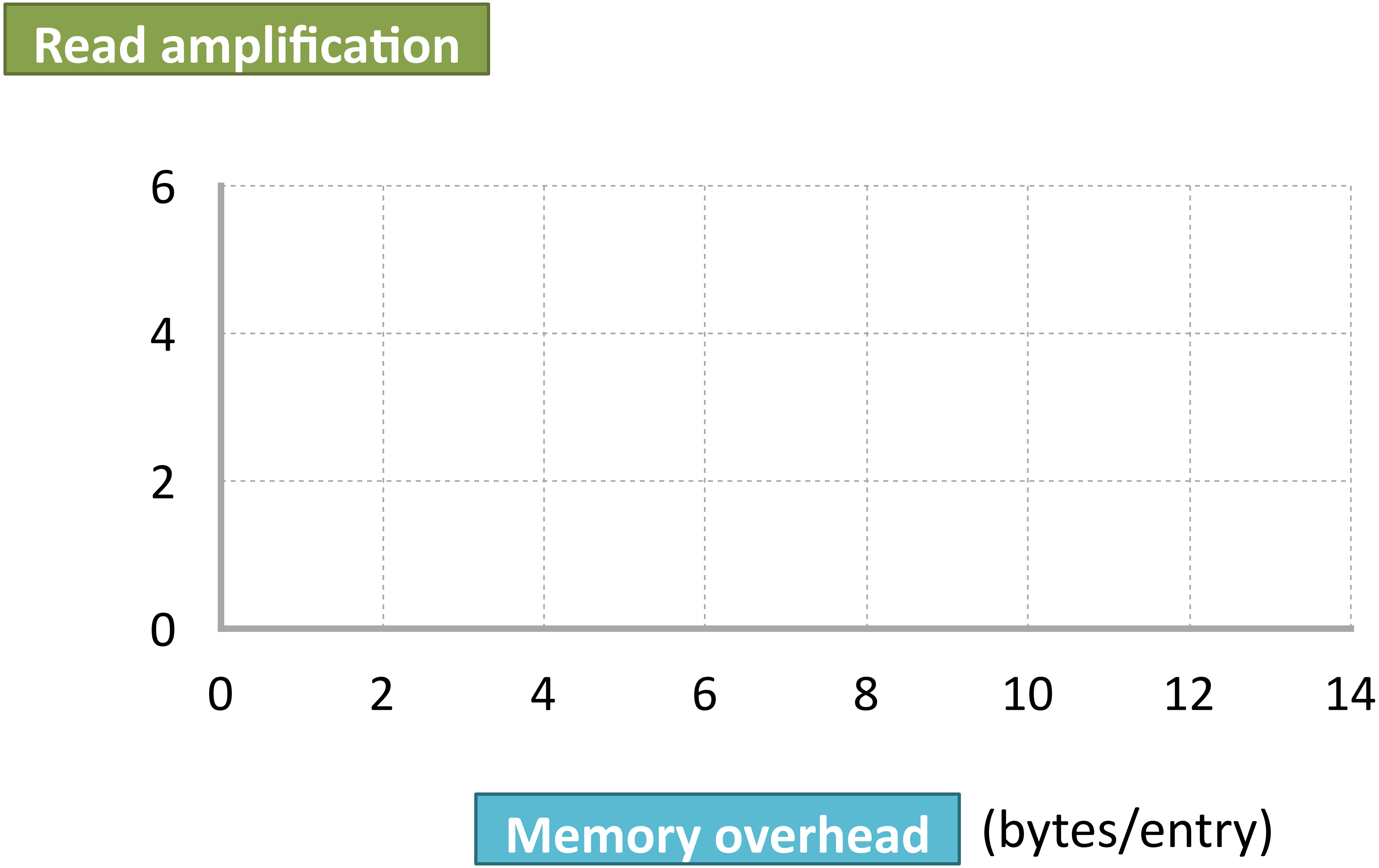
- Ideally 0 bytes/entry (no memory overhead)

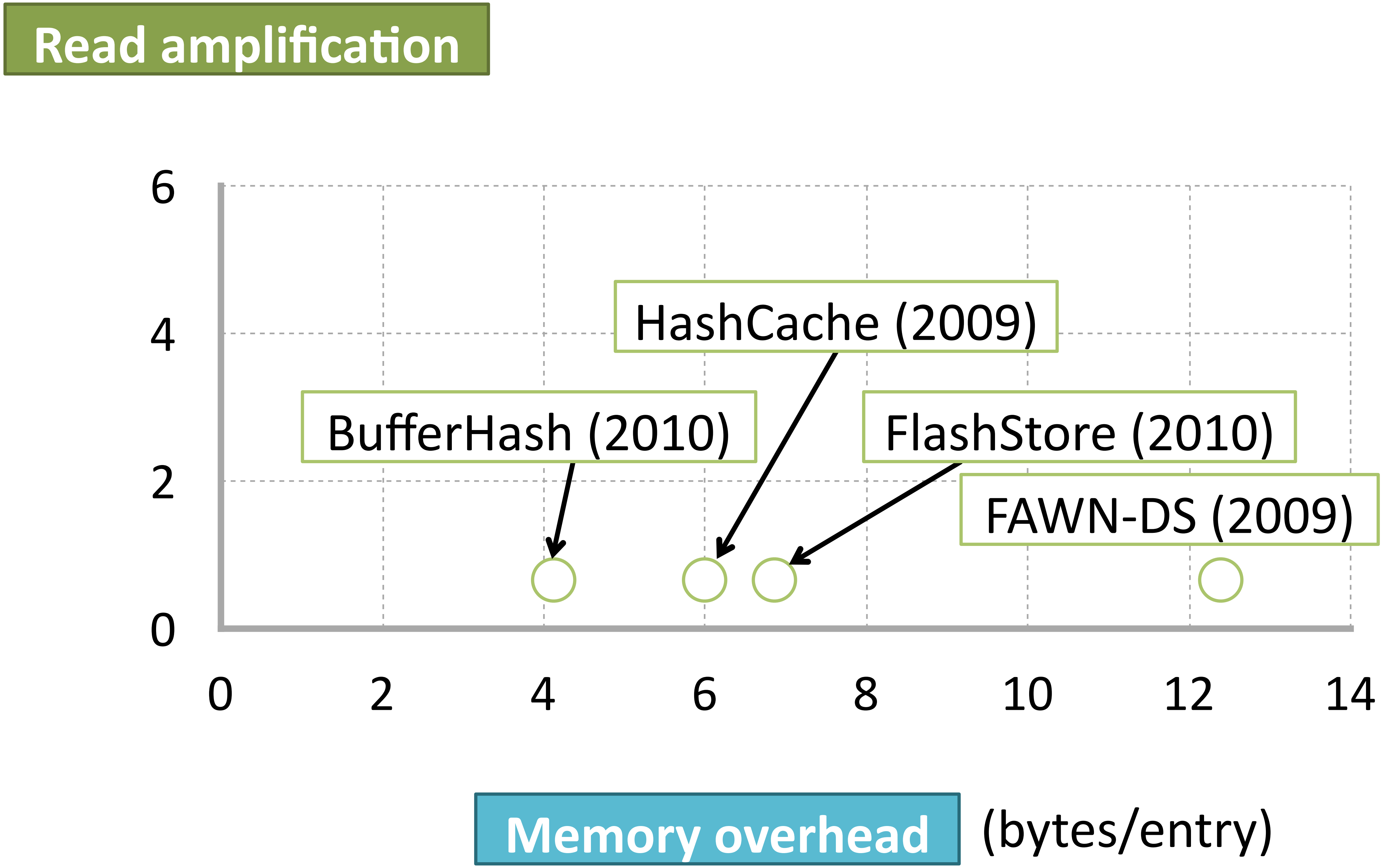
Read amplification = Flash reads per query

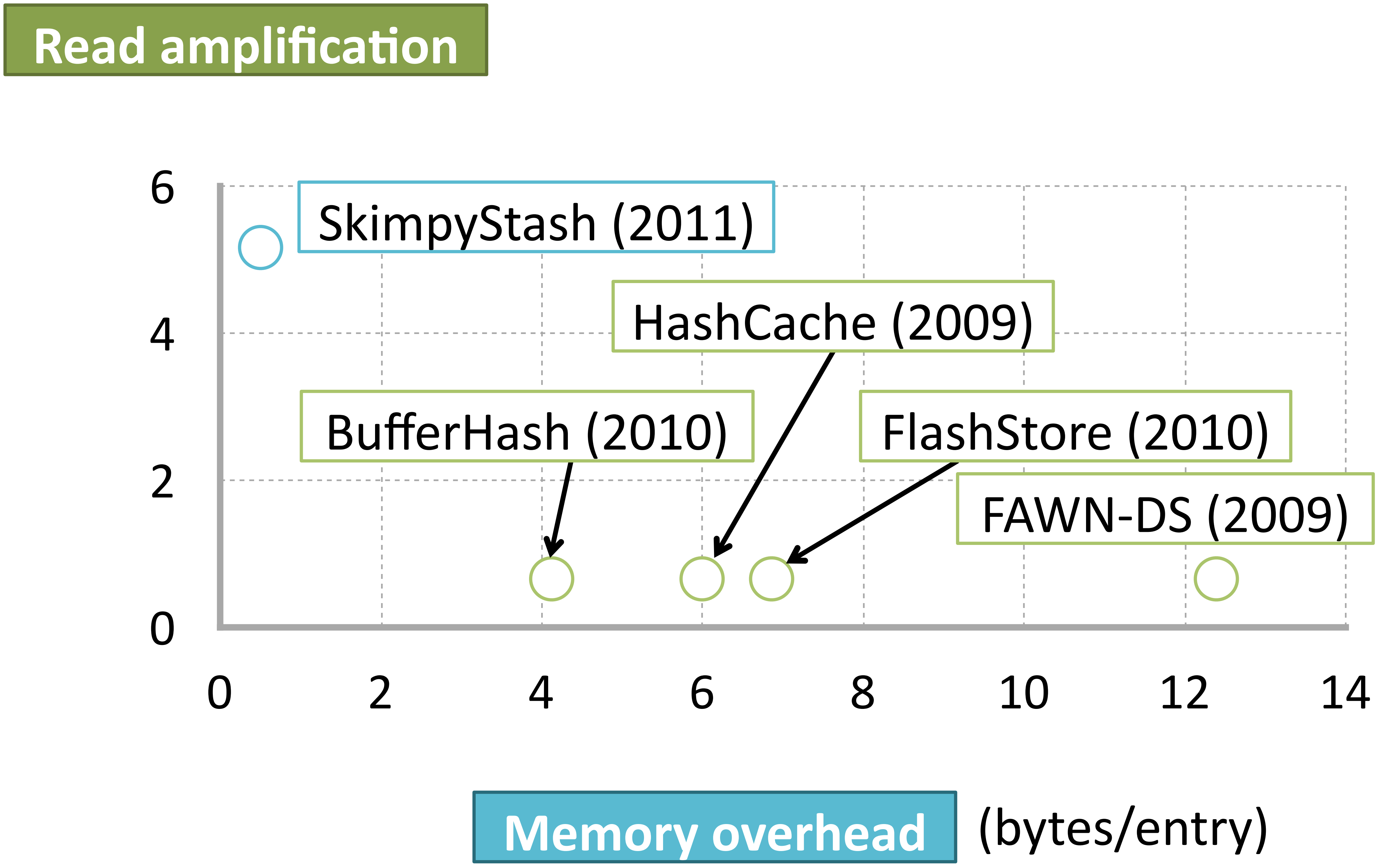
- Limits **query throughput**
- Ideally 1 (no wasted flash reads)

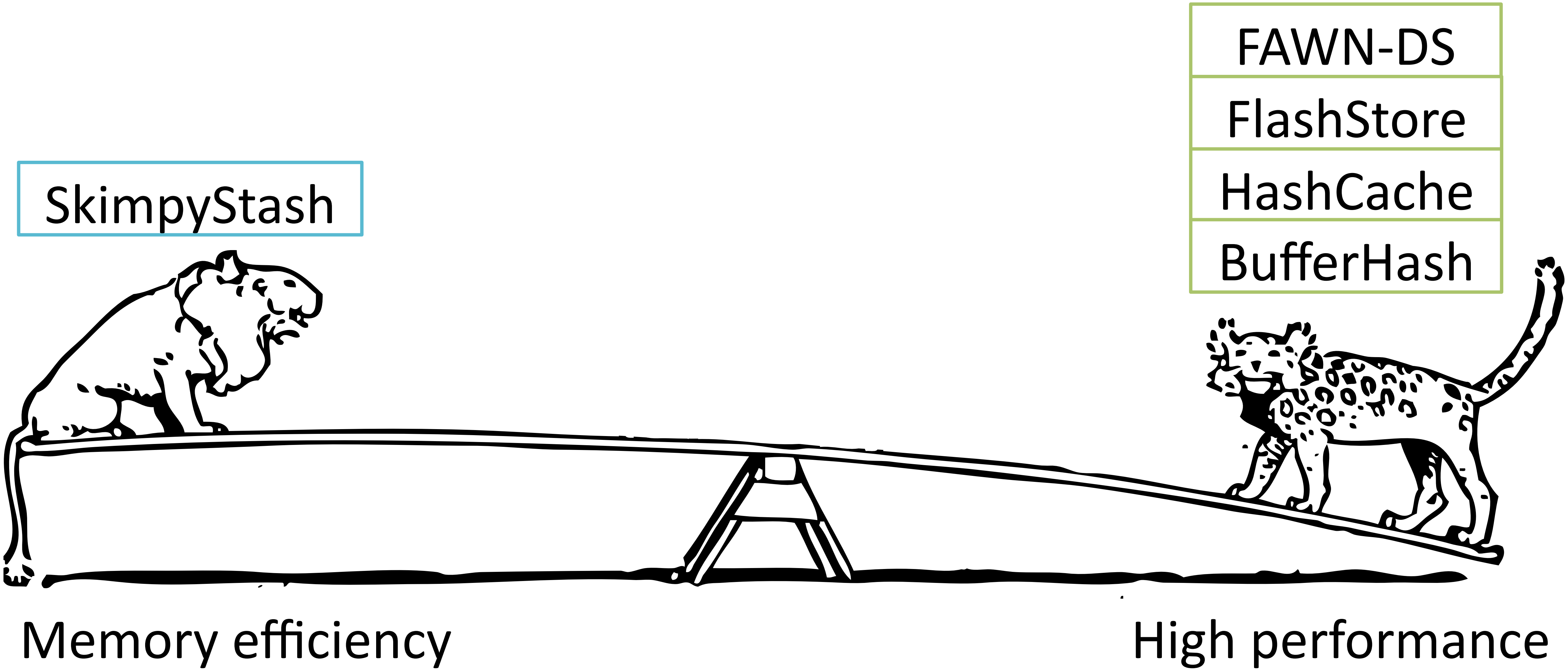
Write amplification = Flash writes per entry

- Limits **insert throughput**
- Also reduces **flash life expectancy**
  - Must be small enough for flash to last a few years


Read amplification
6
4
2
0
0
2
4
6
8
10
12
14
Memory overhead
(bytes/entry)


Read amplification
6
4
2
0
HashCache (2009)
BufferHash (2010)
FlashStore (2010)
FAWN-DS (2009)
0
2
4
6
8
10
12
14
Memory overhead
(bytes/entry)


Read amplification
6
4
2
0
0
2
4
6
8
10
12
14
SkimpyStash (2011)
HashCache (2009)
BufferHash (2010)
FlashStore (2010)
FAWN-DS (2009)
Memory overhead
(bytes/entry)



SkimpyStash

FAWN-DS

FlashStore

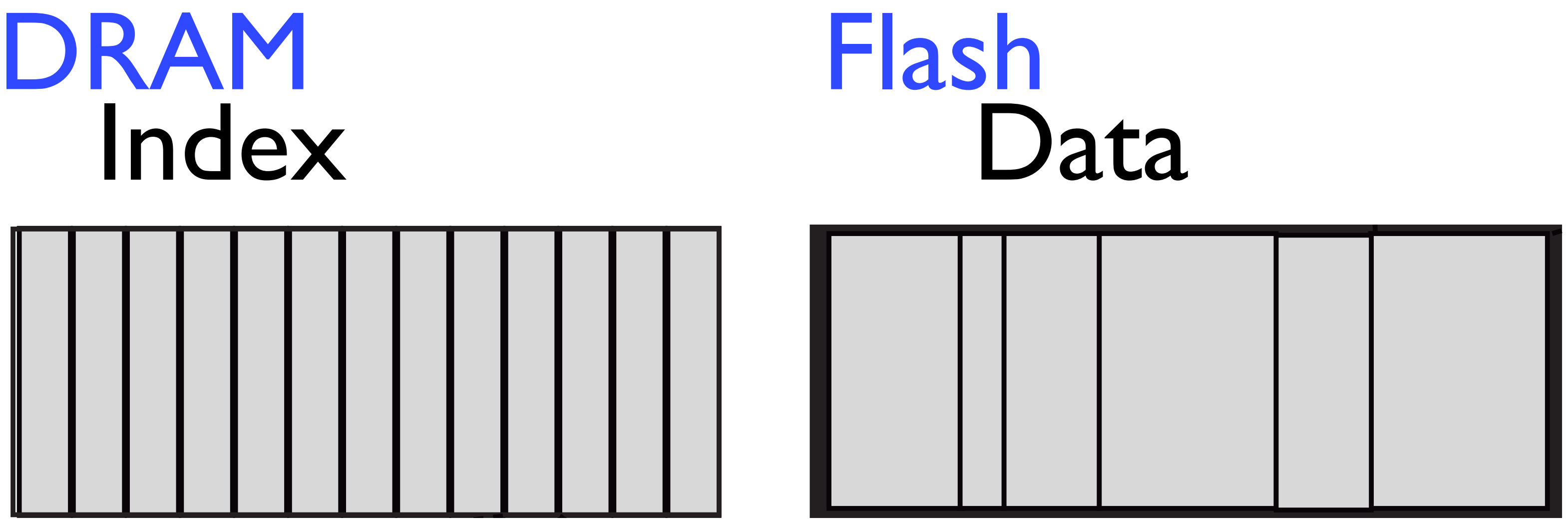
HashCache

BufferHash

Memory efficiency

High performance

(static) “External Dictionary”



DRAM

Index

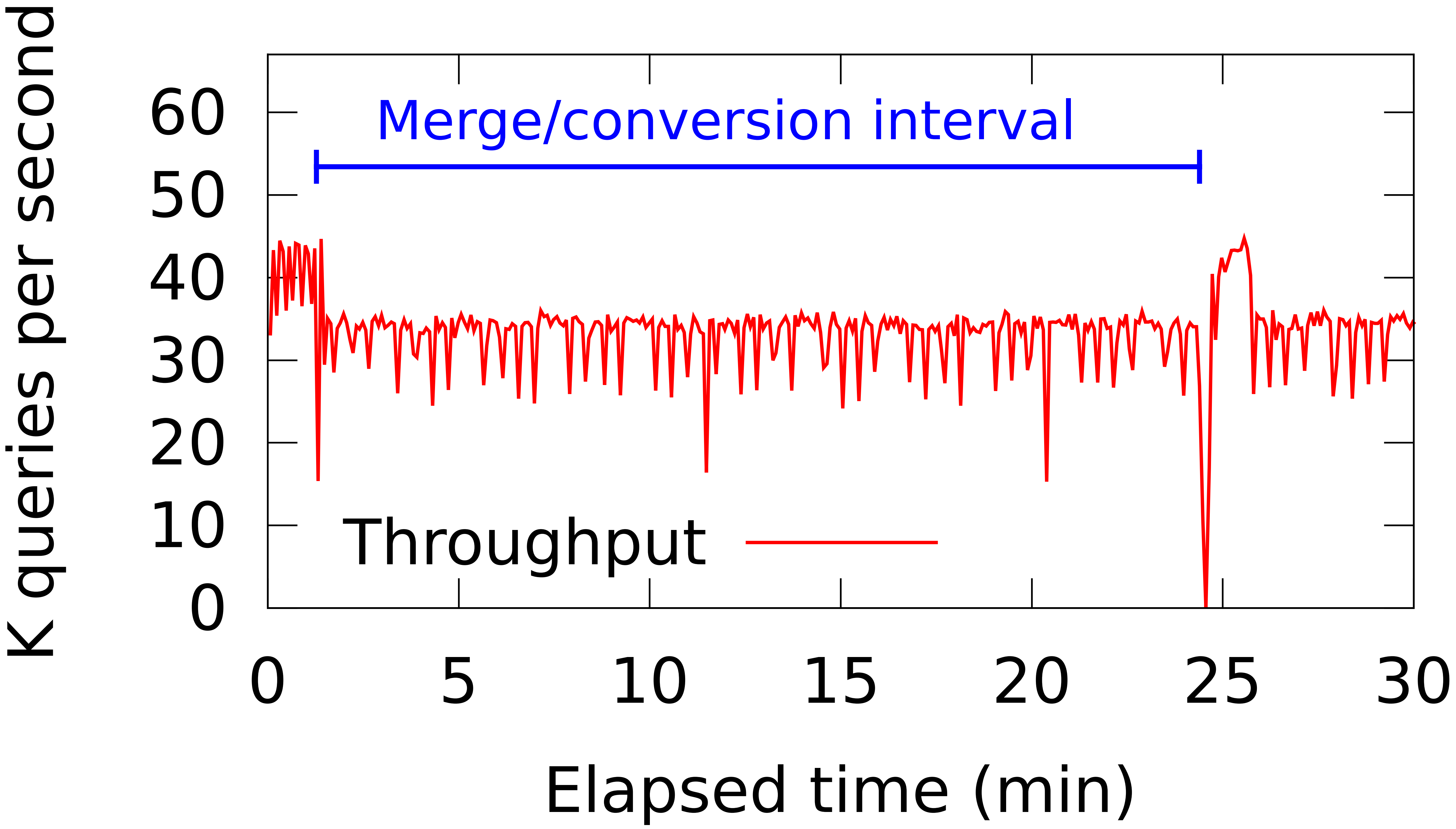
Flash

Data

- Prior state of the art: “EPH”: ~3.8 bits/entry
- Ours: Entropy-coded tries, ~2.5 bits/entry

- Important considerations:
  - Construction speed; query speed
  - Aw, it’s read-only... [need a system; built it]

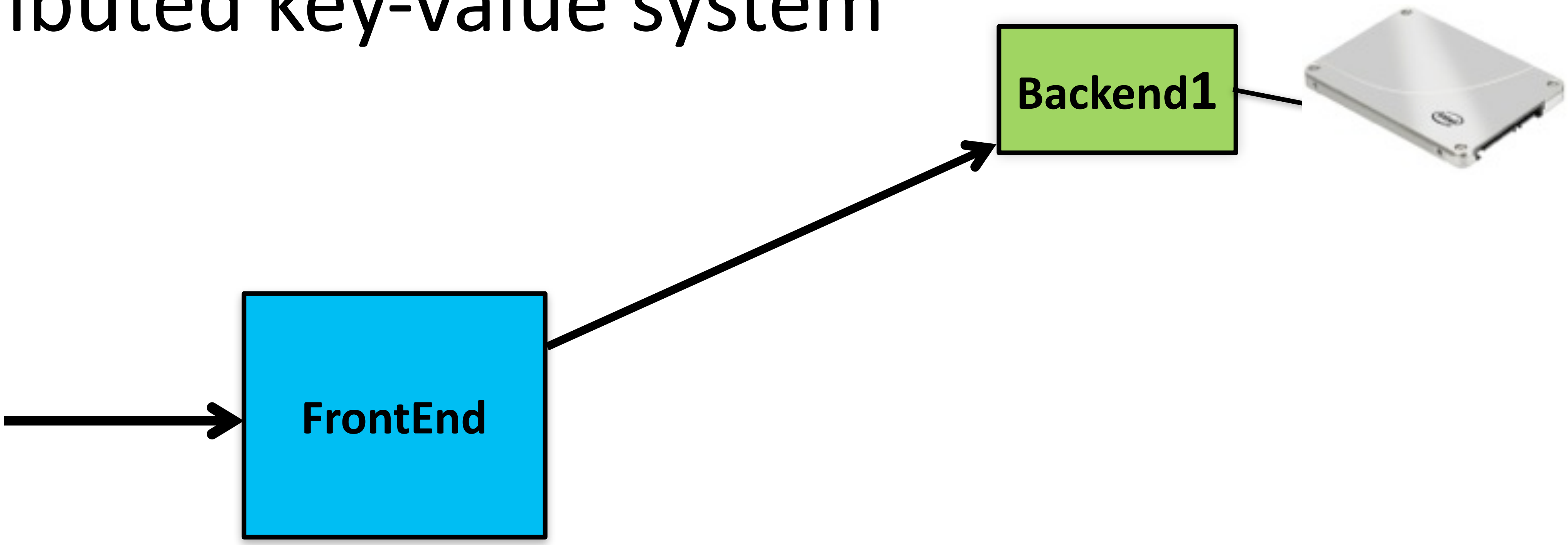
Workload: 90% GET (100~ M keys) + 10% PUT



Caveat: Not on wimpies. Still working on reducing CPU cost! :-)

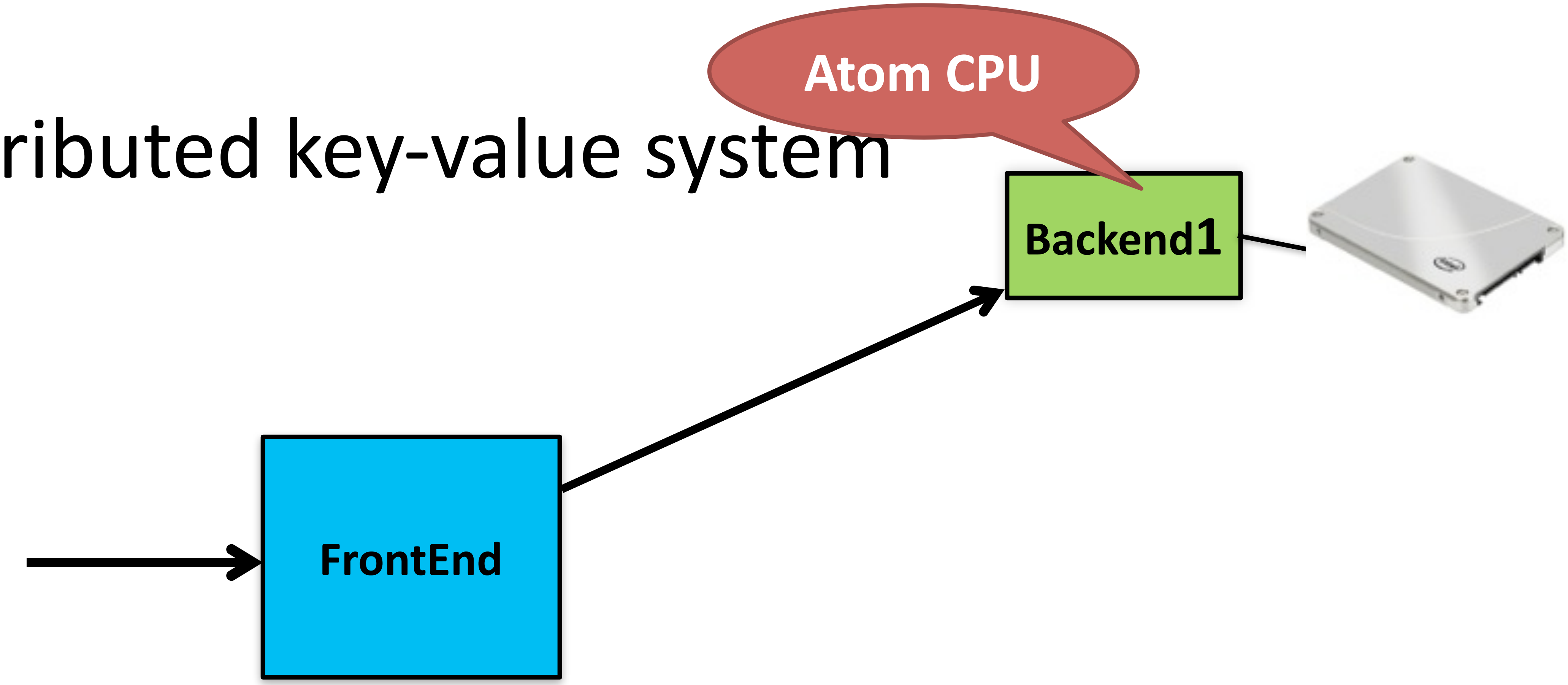
And now... Load imbalance

- Distributed key-value system



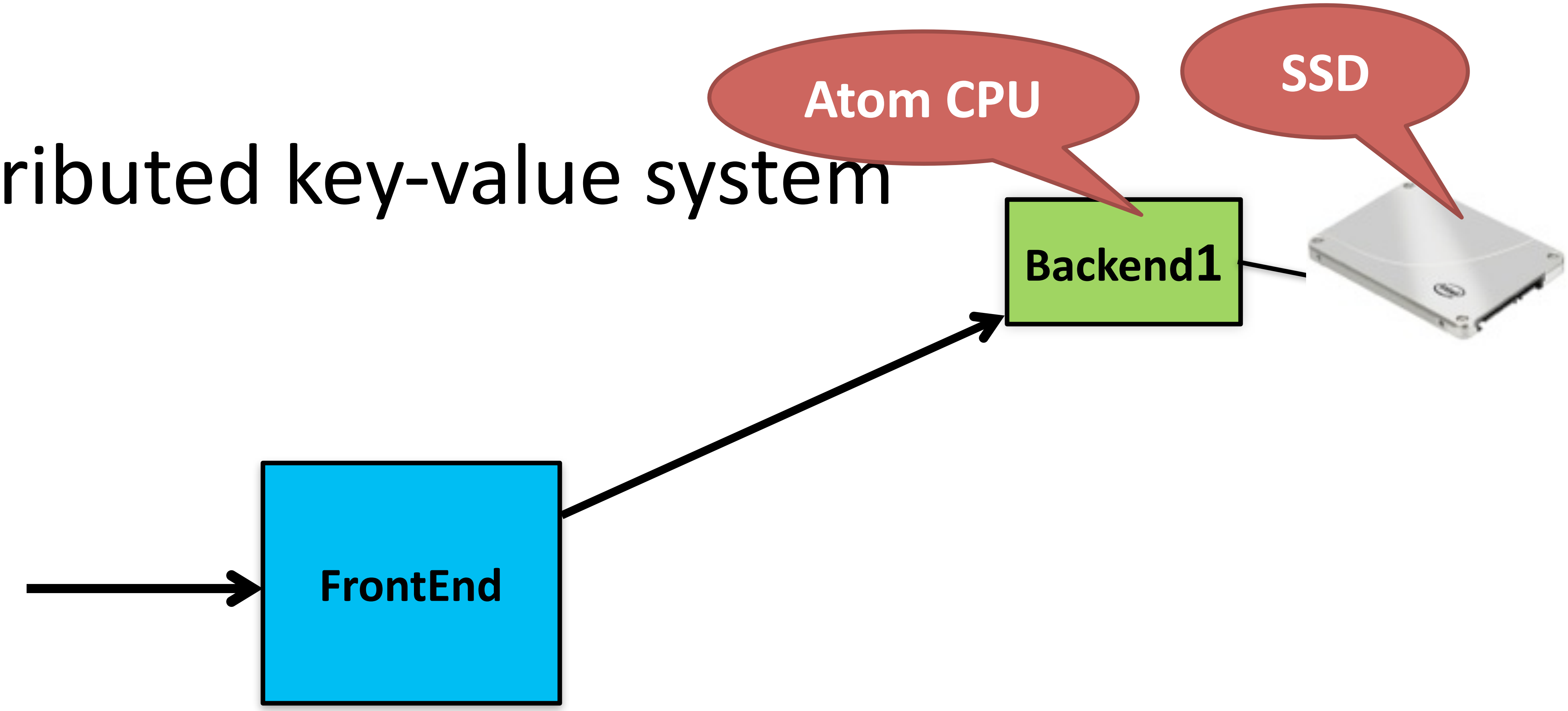
And now... Load imbalance

- Distributed key-value system



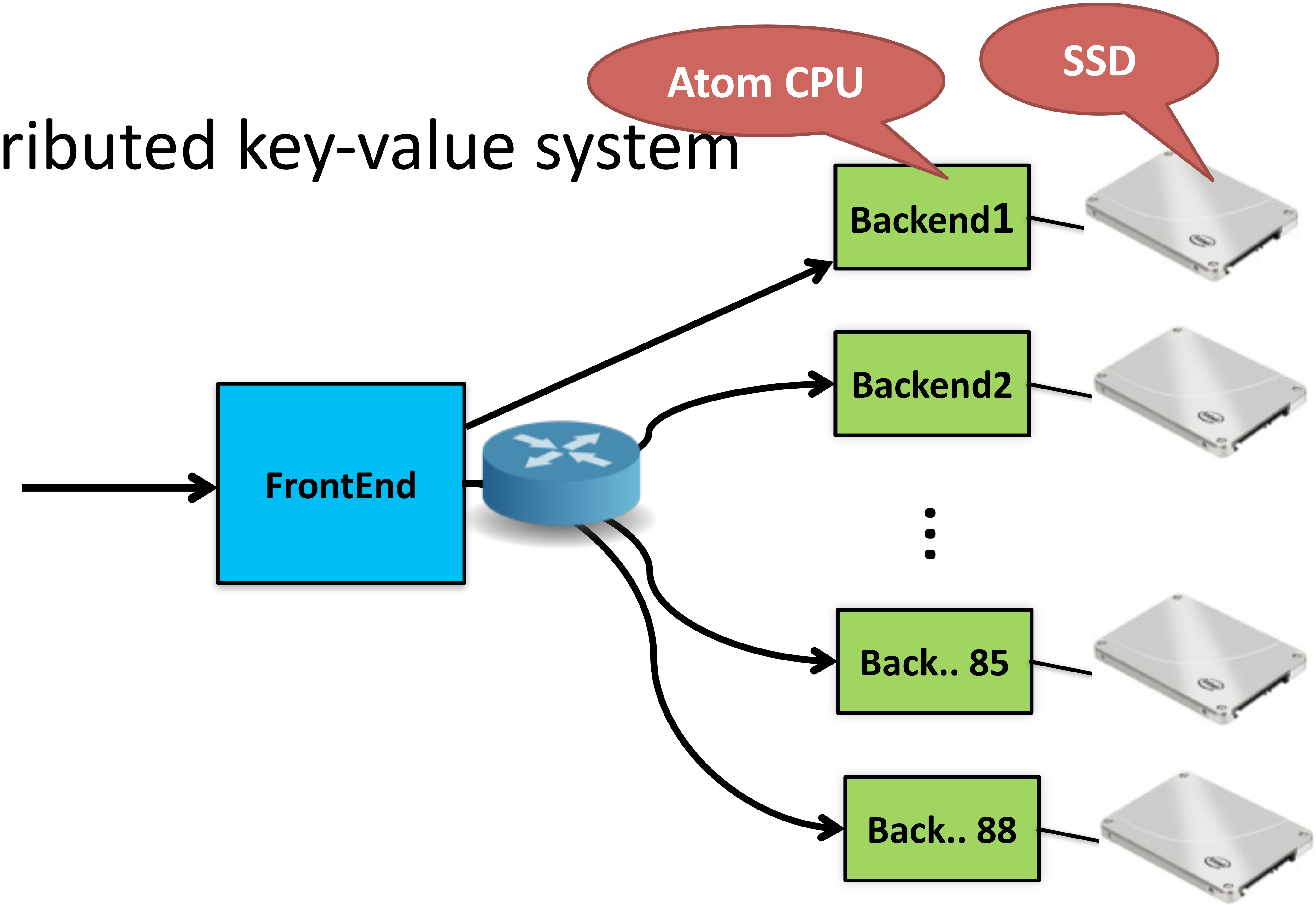
And now... Load imbalance

- Distributed key-value system



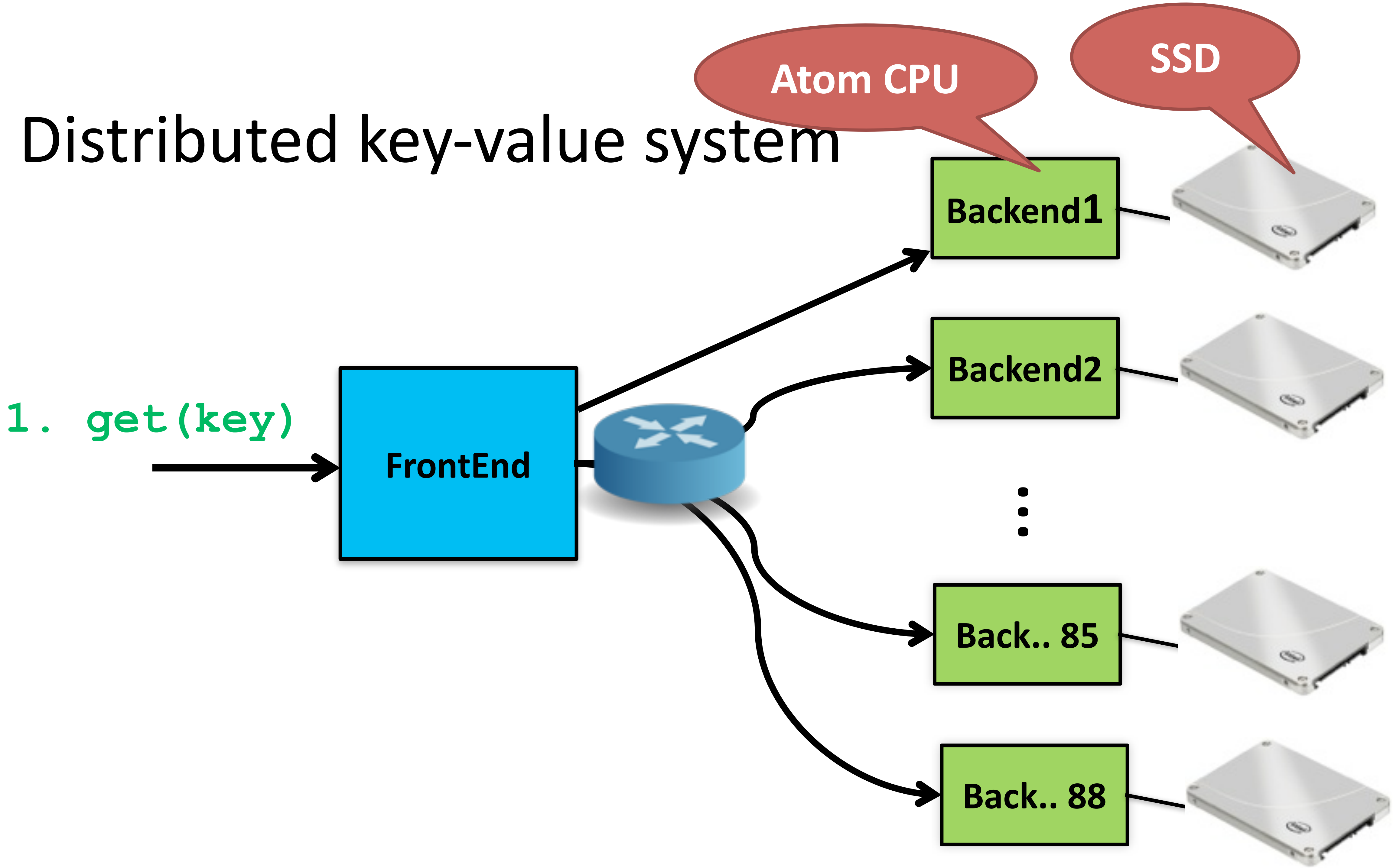
And now... Load imbalance

- Distributed key-value system



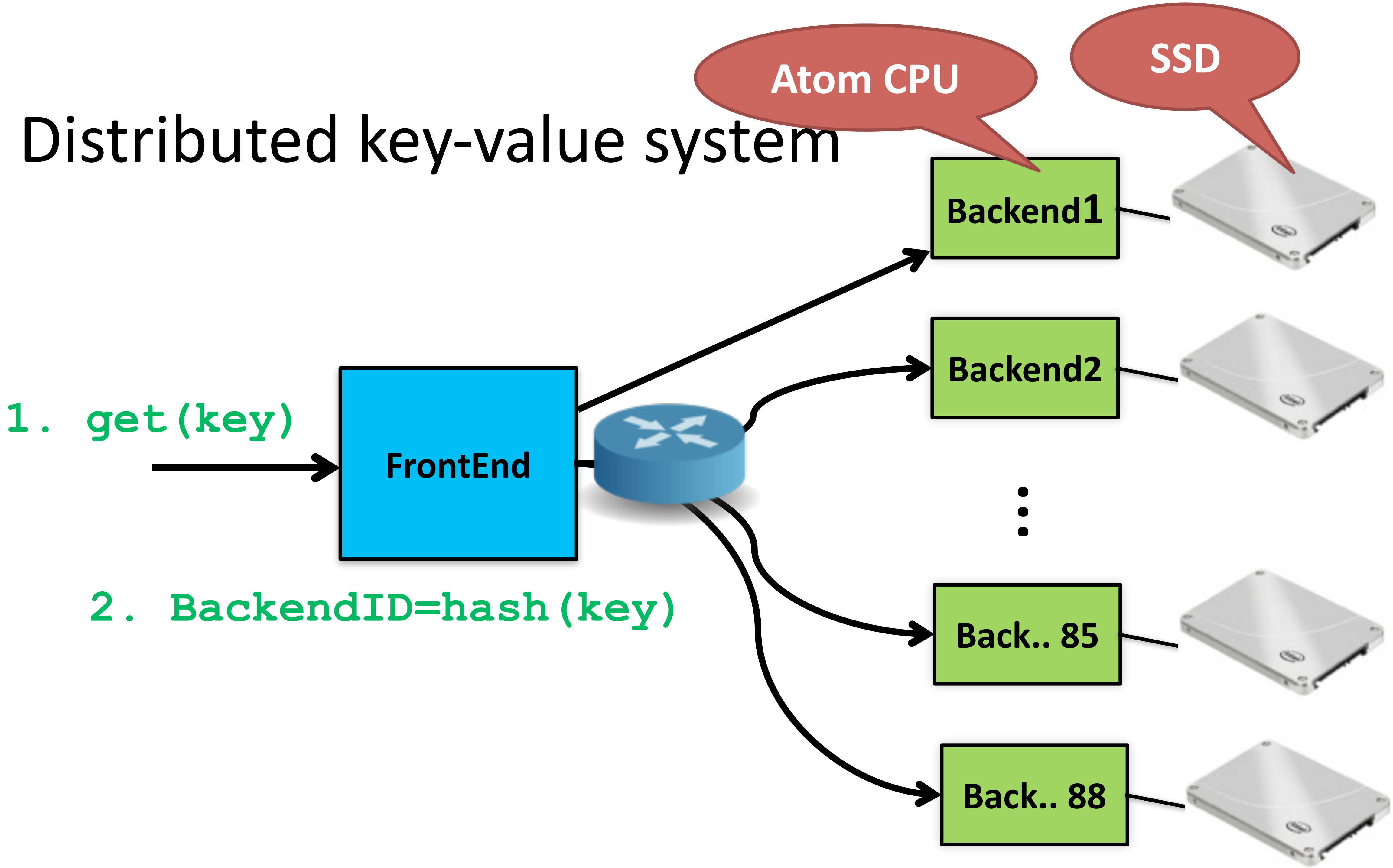
And now… Load imbalance

- Distributed key-value system



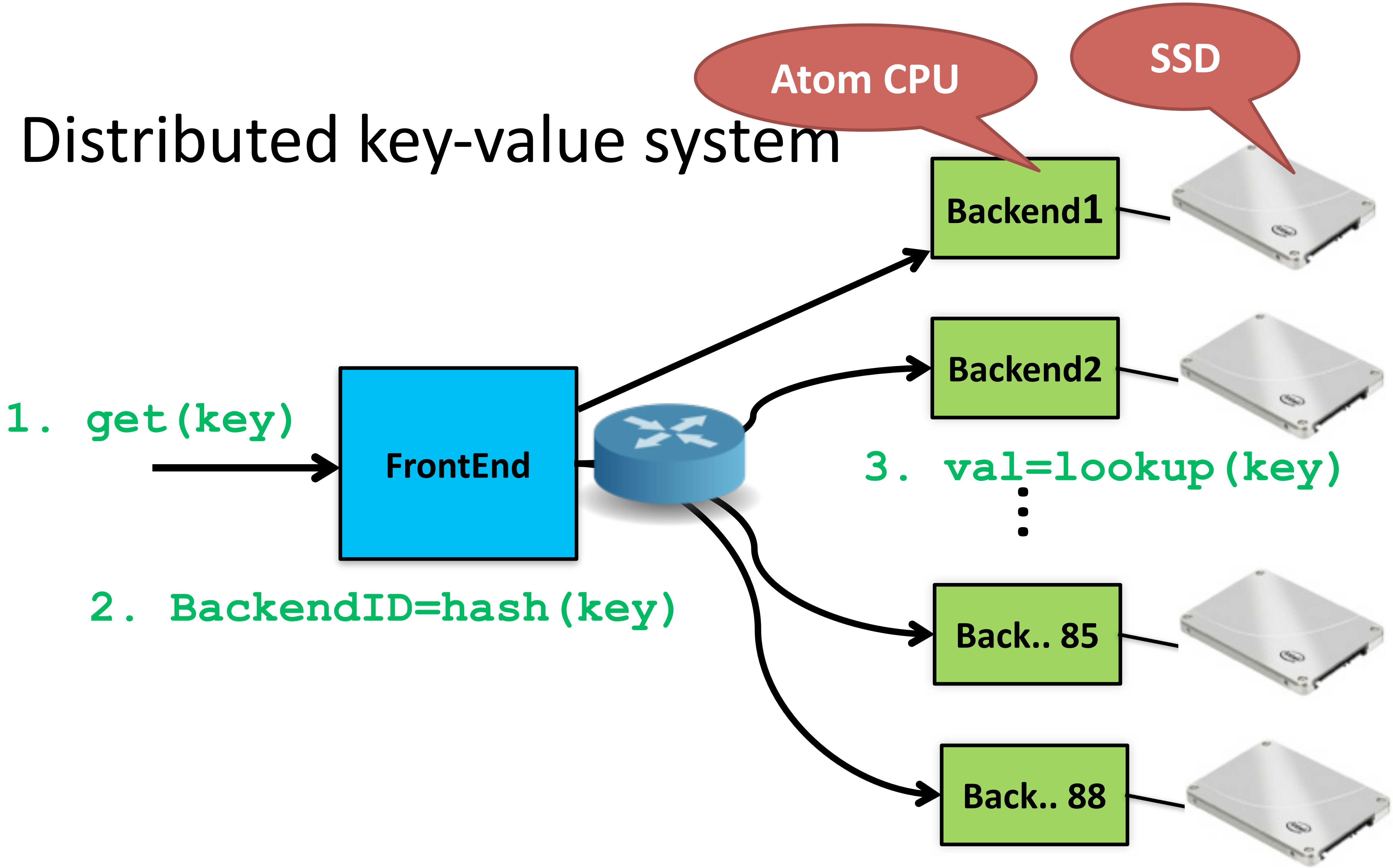
And now… Load imbalance

- Distributed key-value system



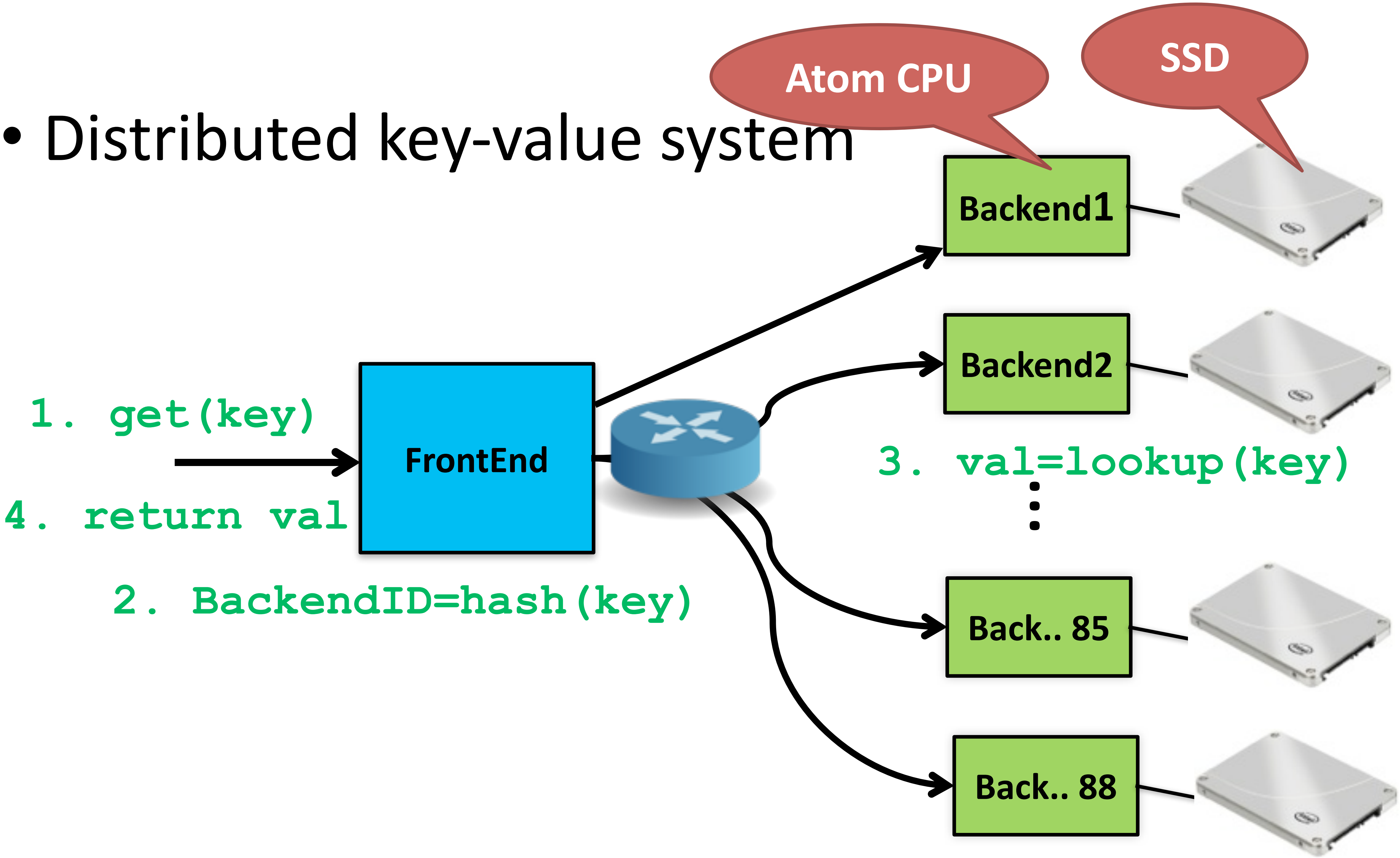
And now… Load imbalance

- Distributed key-value system



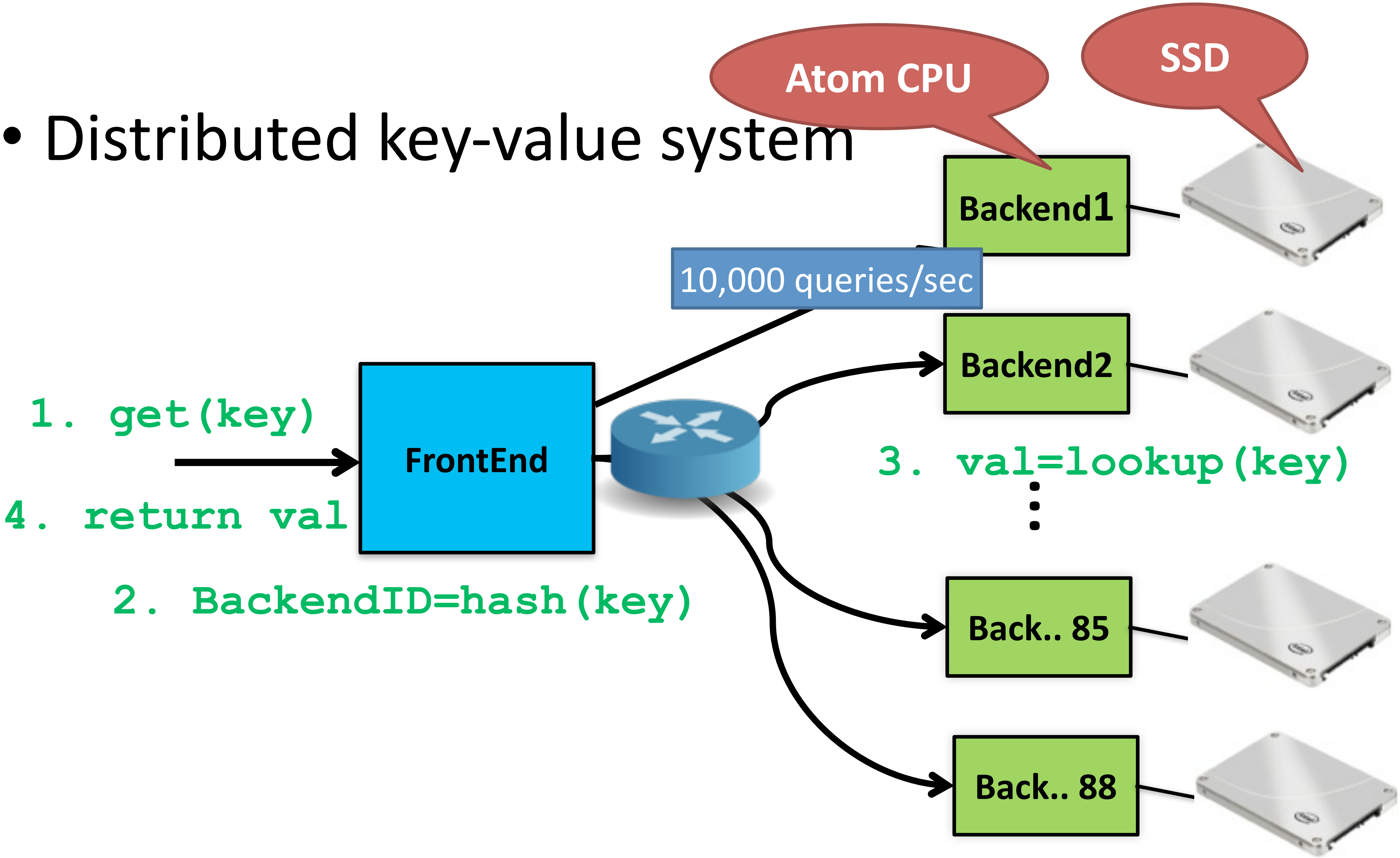
And now… Load imbalance

- Distributed key-value system



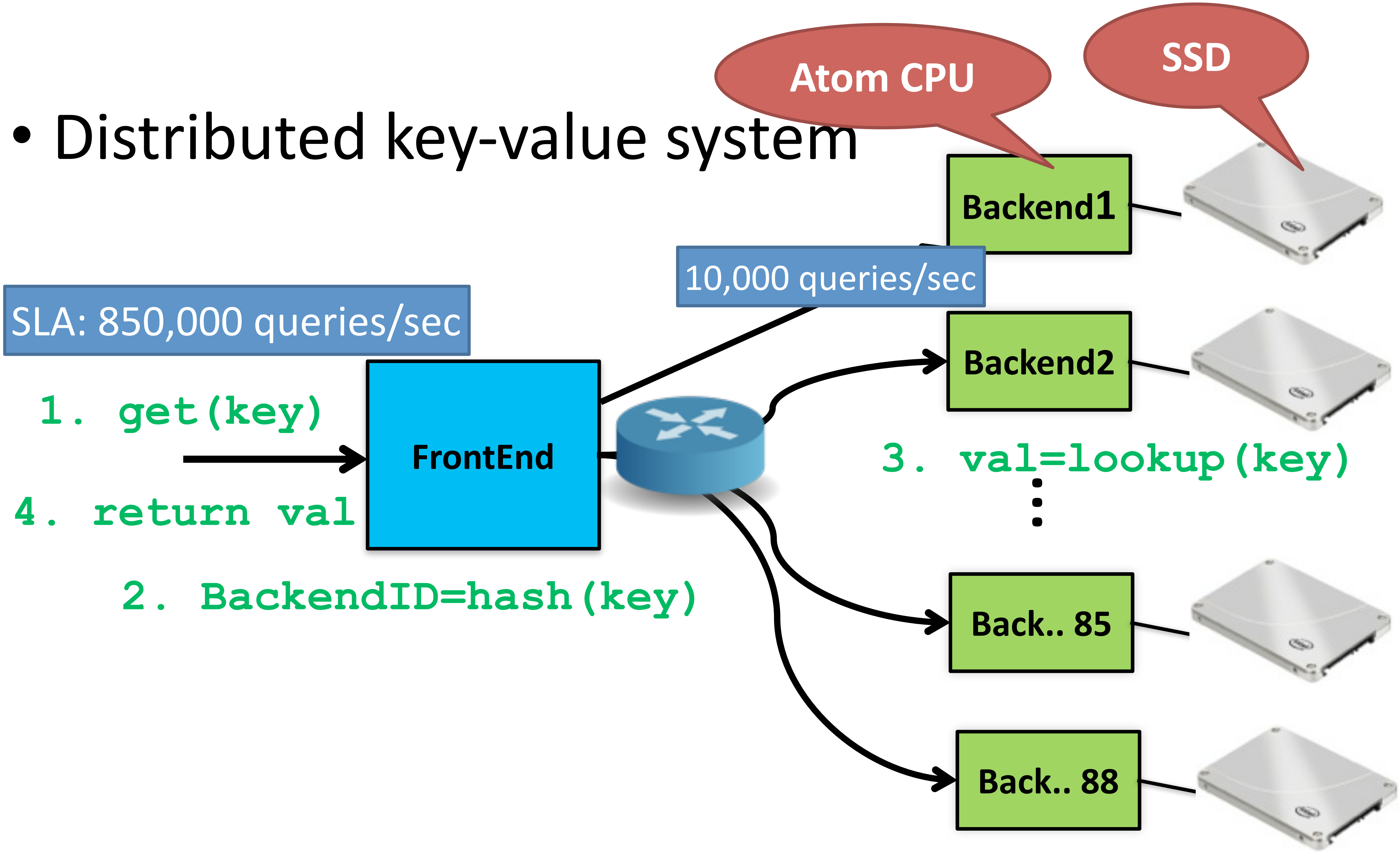
And now... Load imbalance

- Distributed key-value system

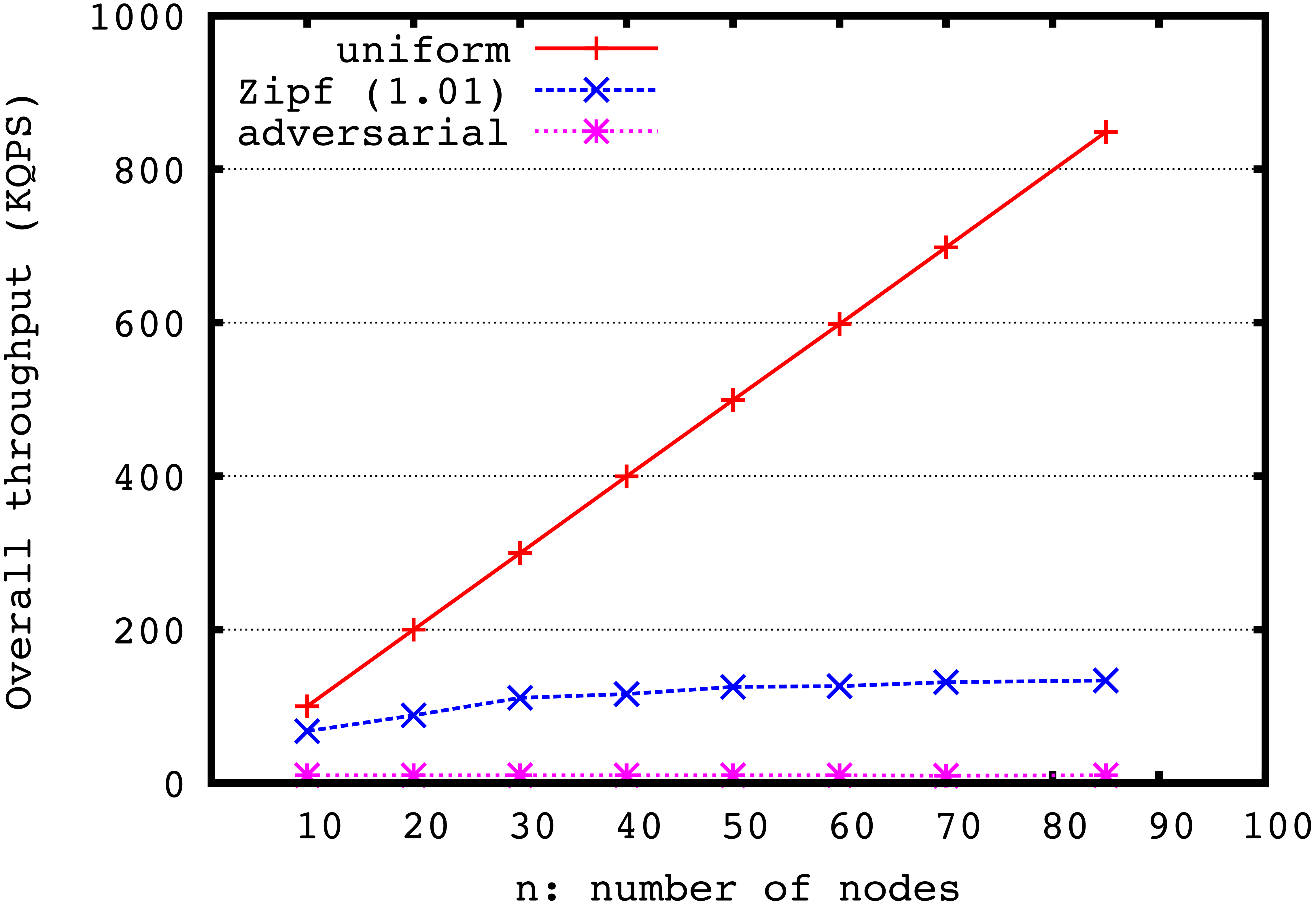


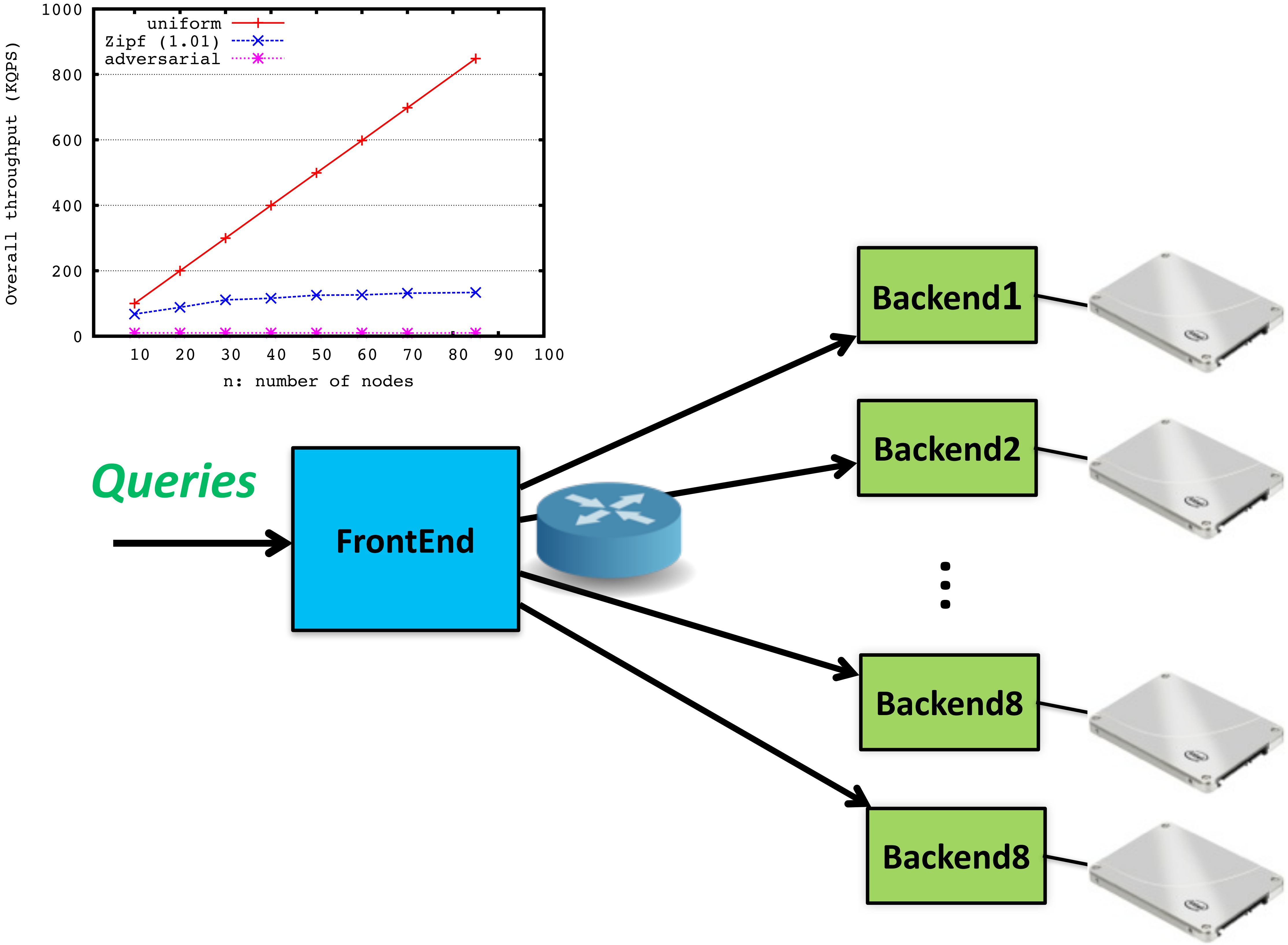
And now… Load imbalance

- Distributed key-value system



Measured tput on FAWN testbed





Queries

FrontEnd

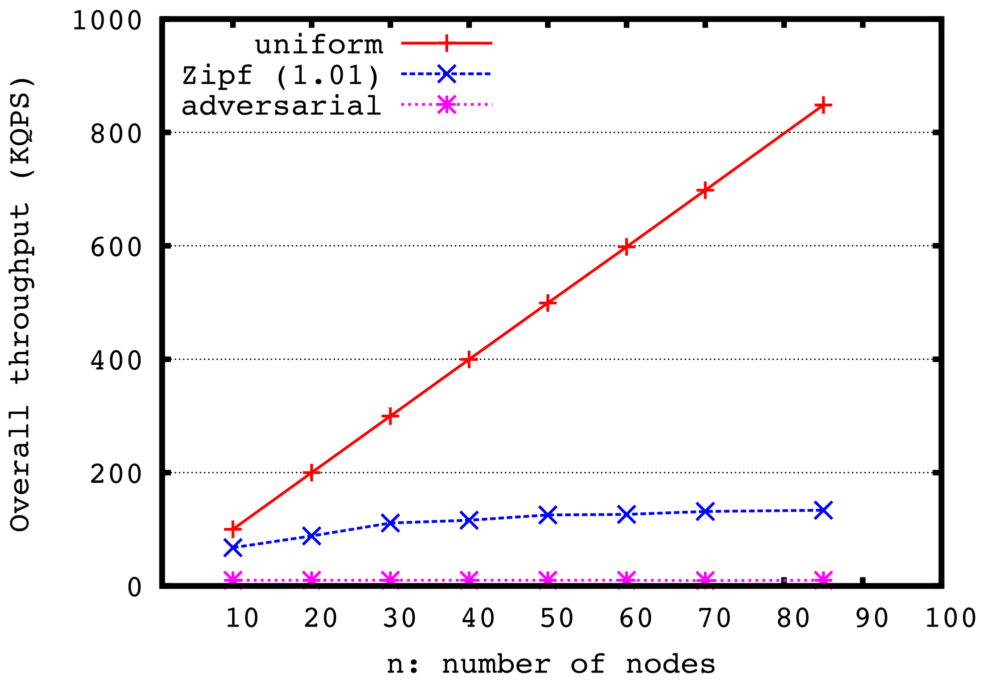
Backend1

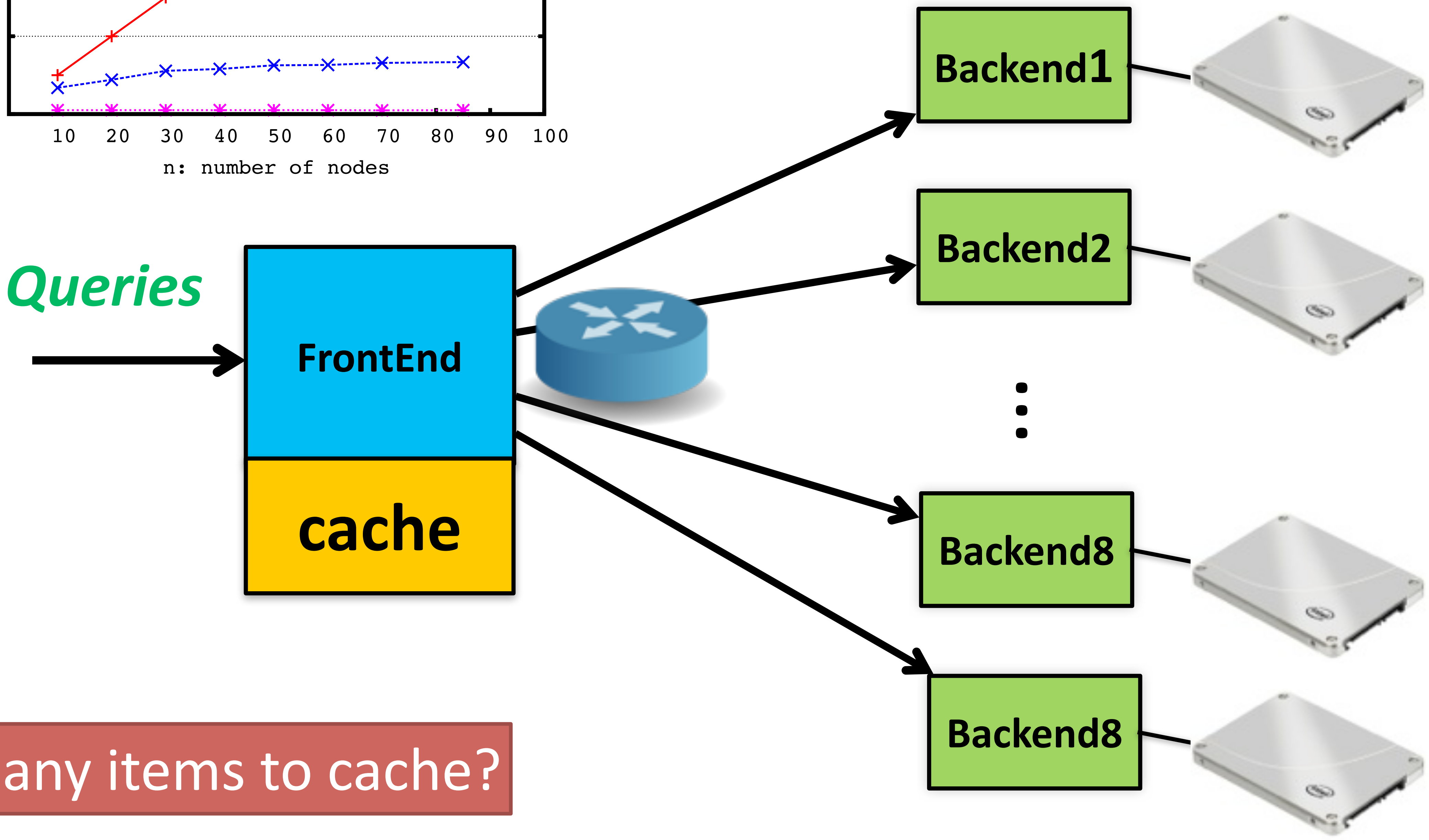
Backend2

⋮

Backend8

Backend8





Queries

FrontEnd

cache

Backend1

Backend2

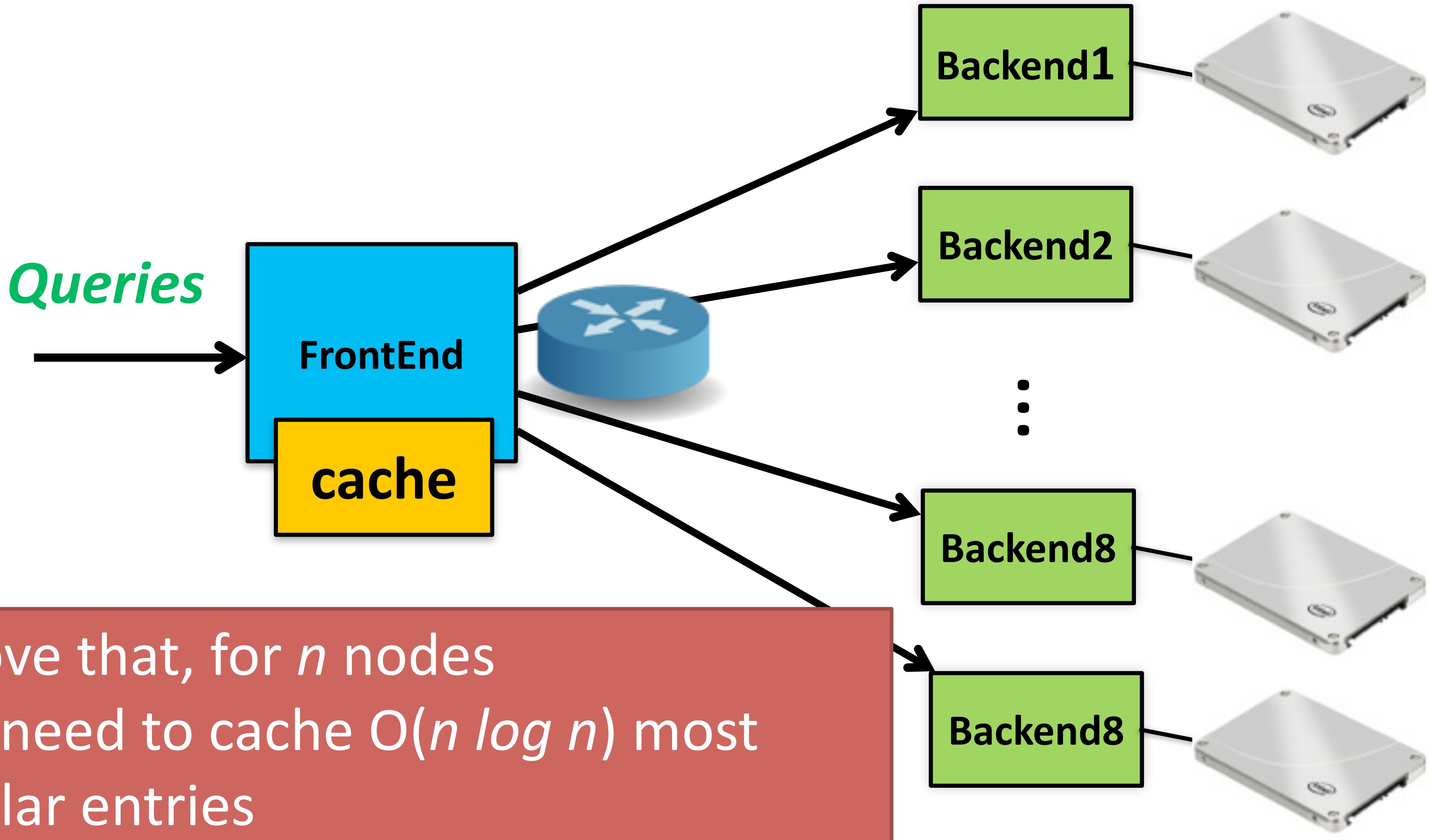
⋮

Backend8

Backend8

How many items to cache?

small/fast cache is enough!

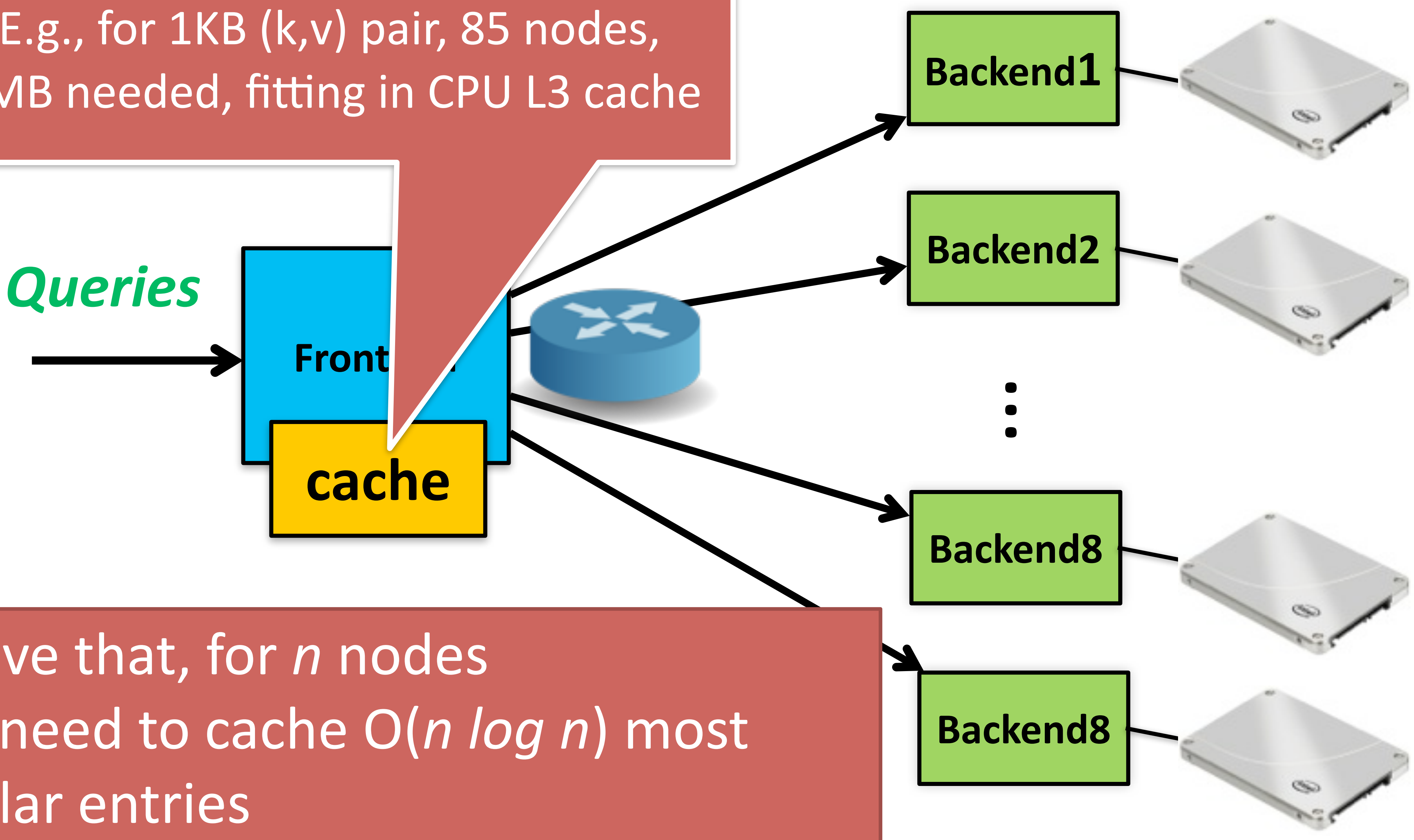


We prove that, for *n* nodes

- Only need to cache O(*n log n*) most popular entries
- worst case perf. = *(1 - ε)* * n * single node capacity

small/fast cache is enough!

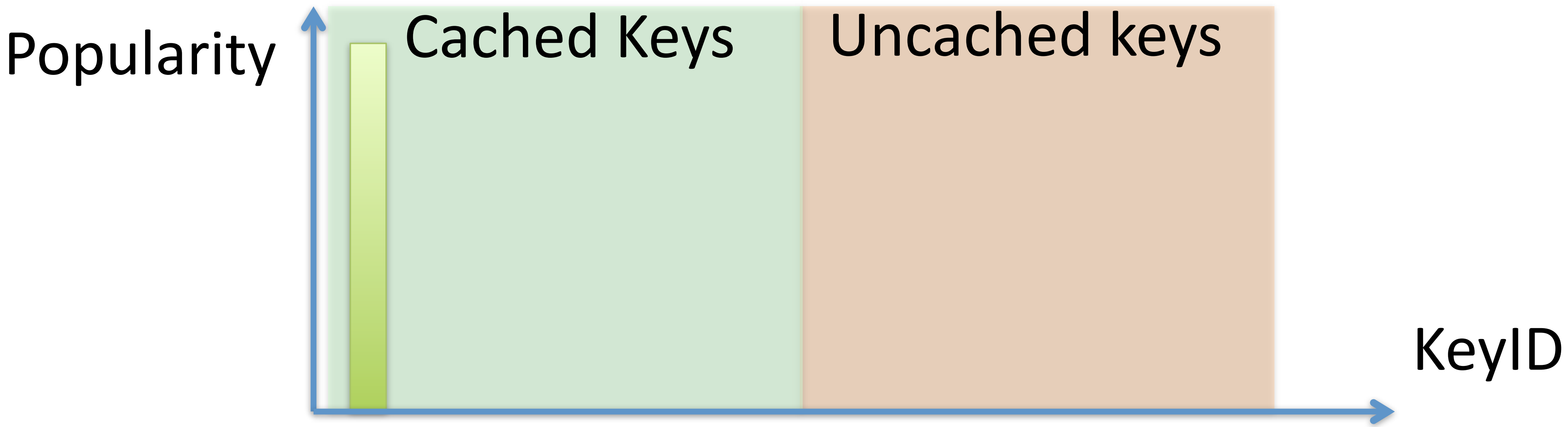
E.g., for 1KB (k,v) pair, 85 nodes, 3MB needed, fitting in CPU L3 cache



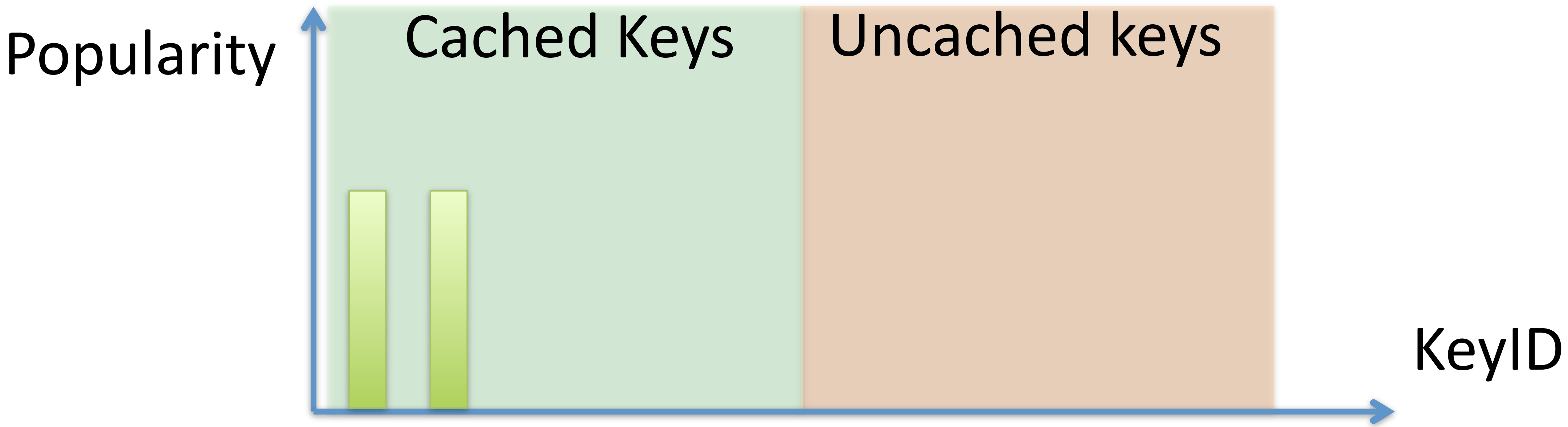
We prove that, for *n* nodes

- Only need to cache O(*n log n*) most popular entries
- worst case perf. = *(1 - ε)* * n * single node capacity

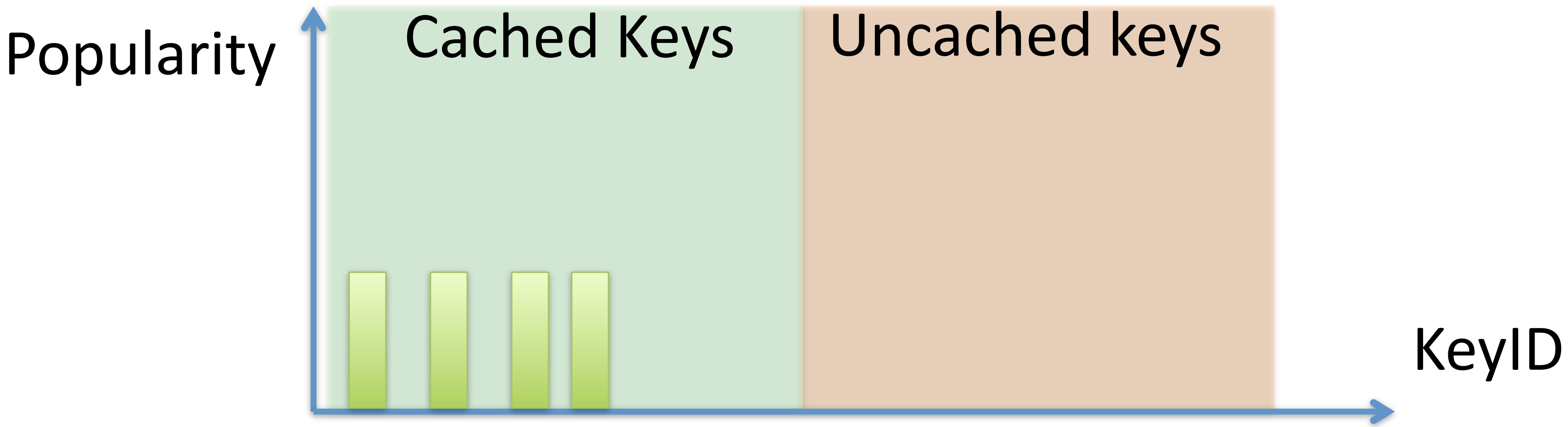
Cache forces near-uniform dist.



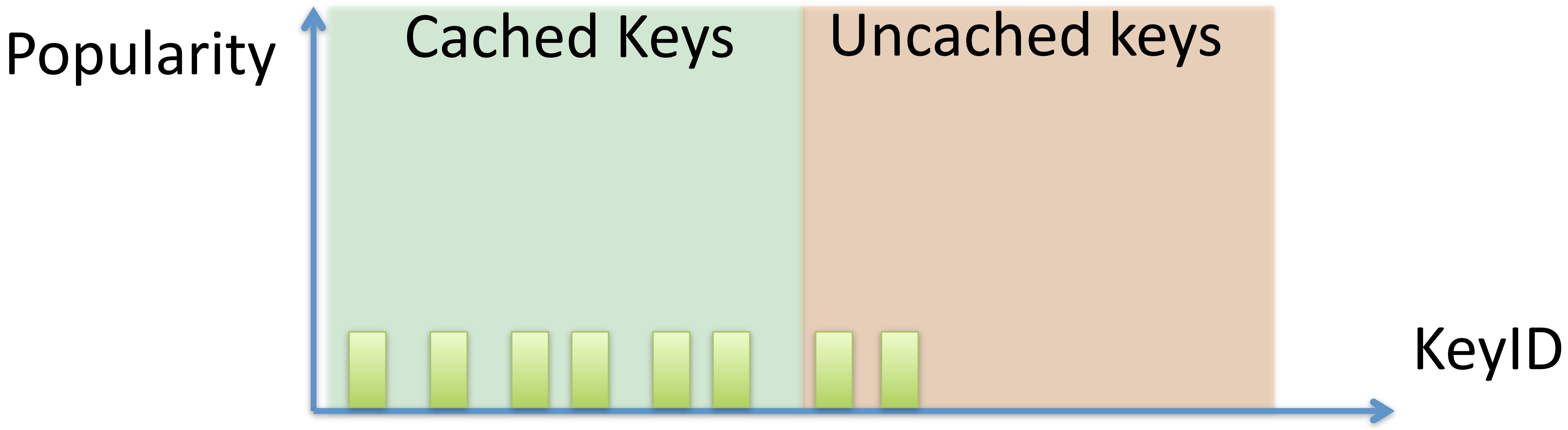
Cache forces near-uniform dist.



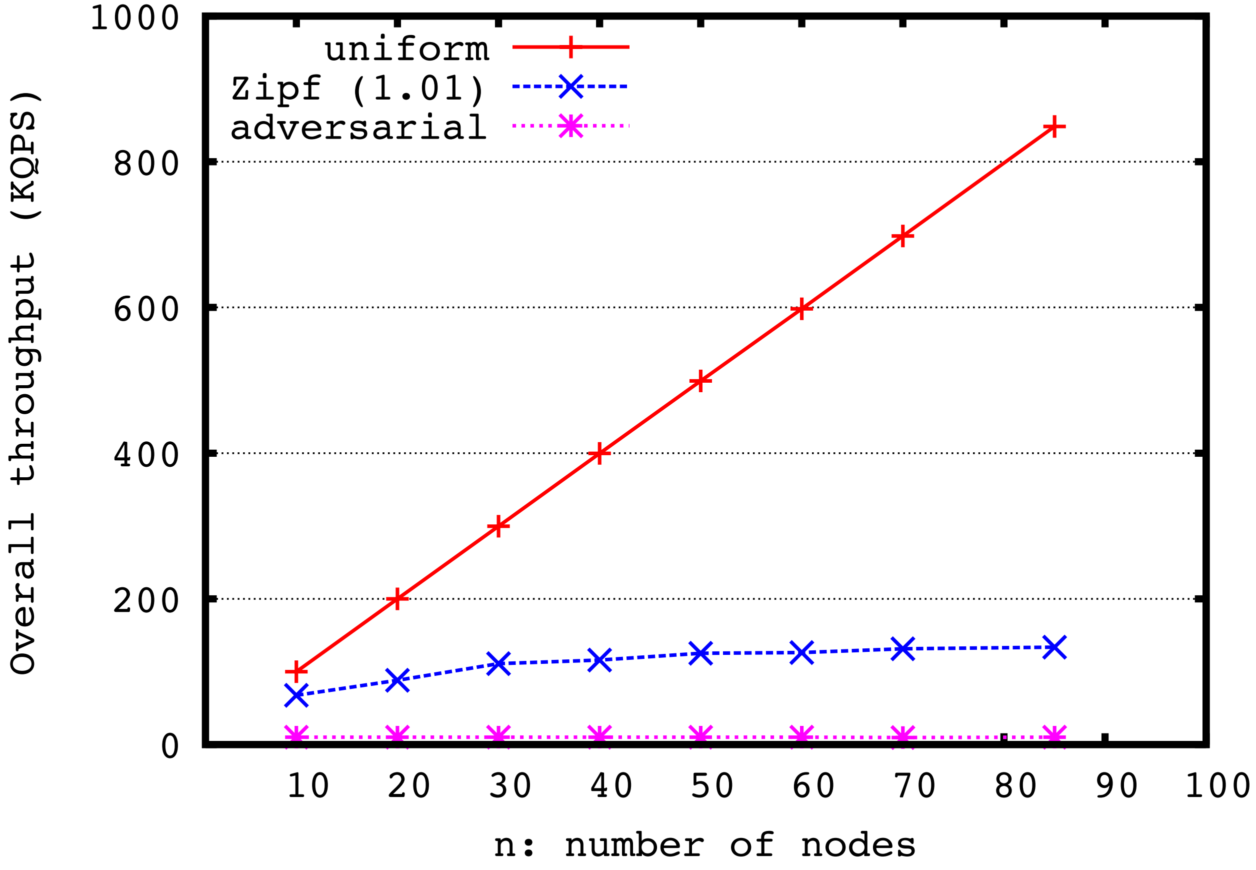
Cache forces near-uniform dist.

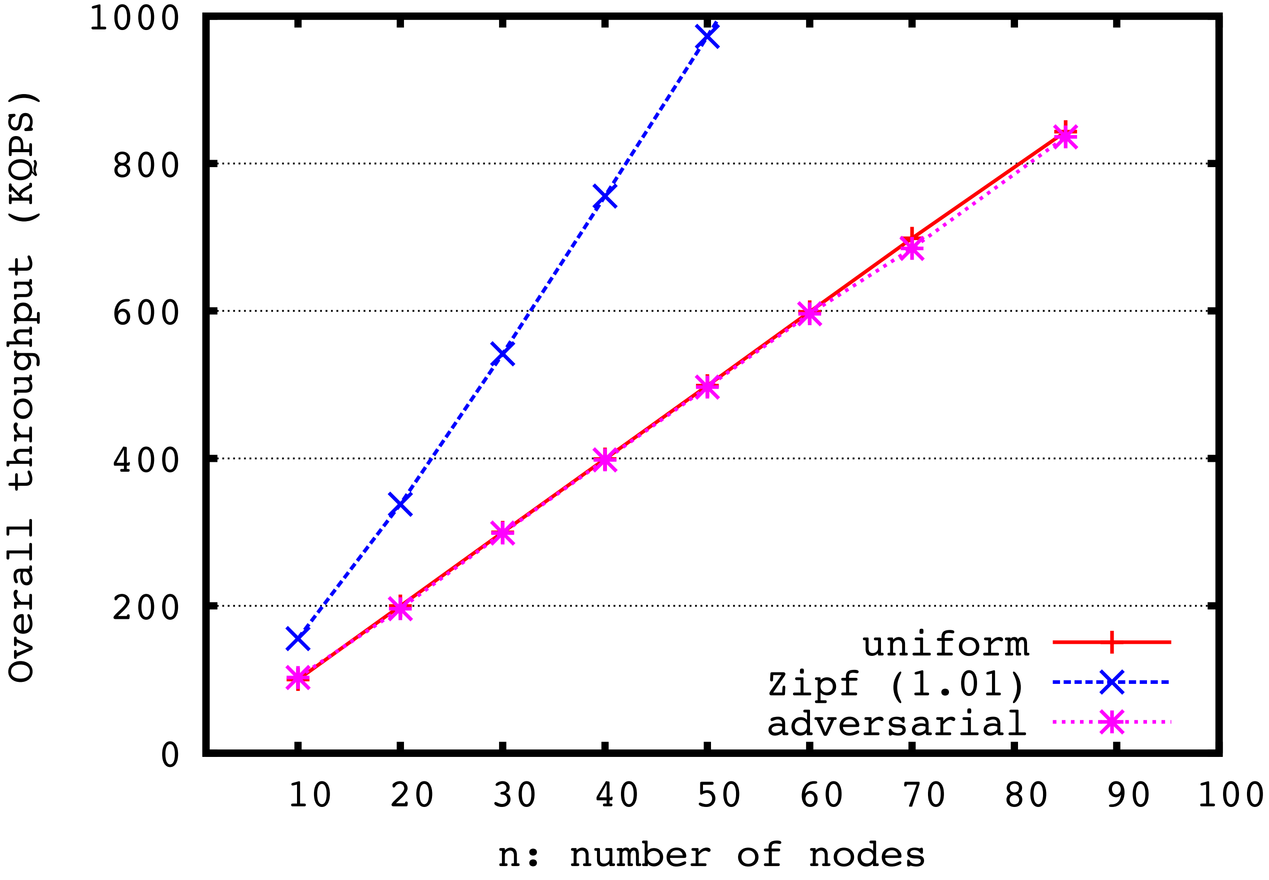


Cache forces near-uniform dist.

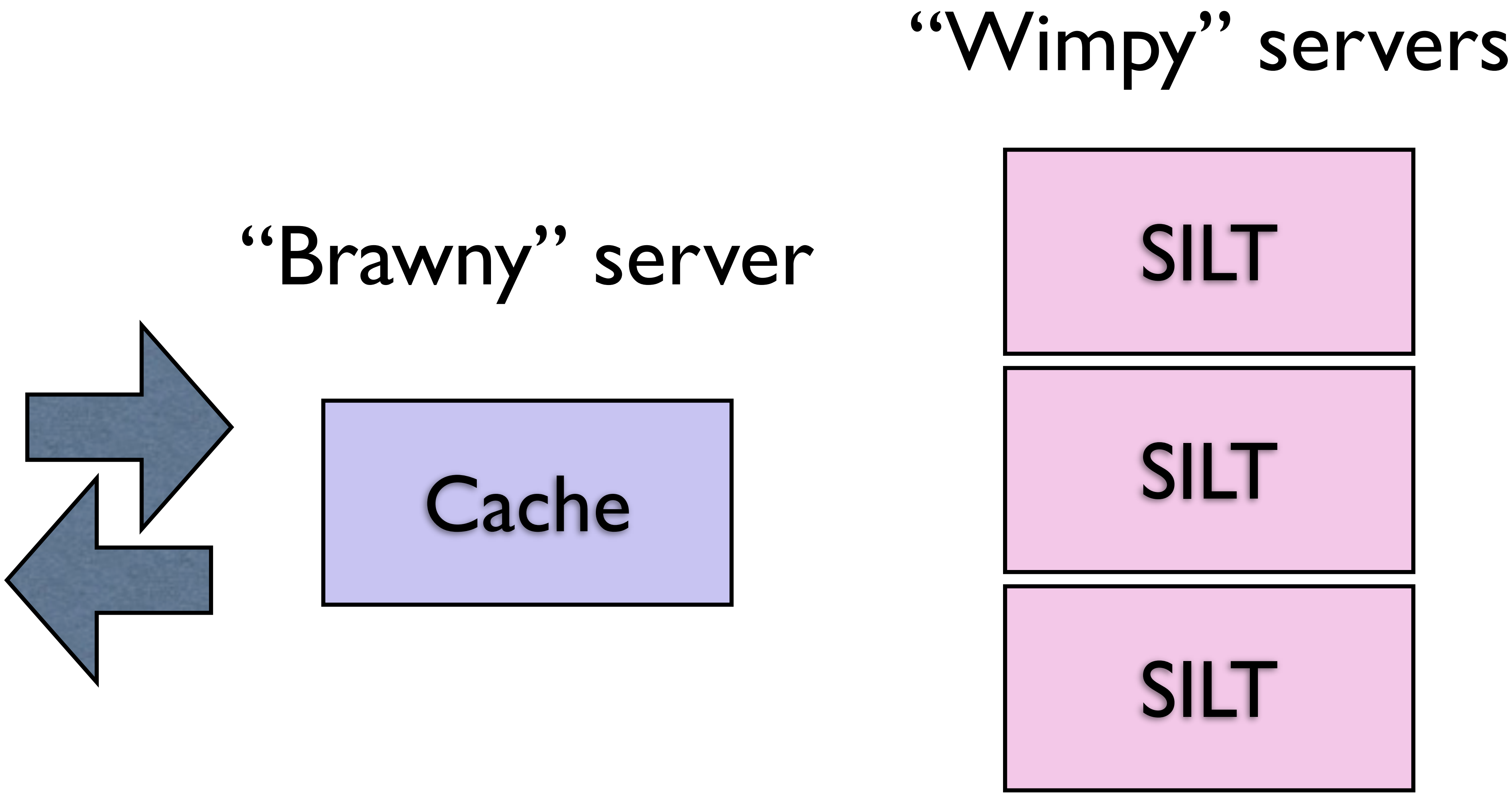


Worst case? Now best case





Thus...



“Wimpy” servers

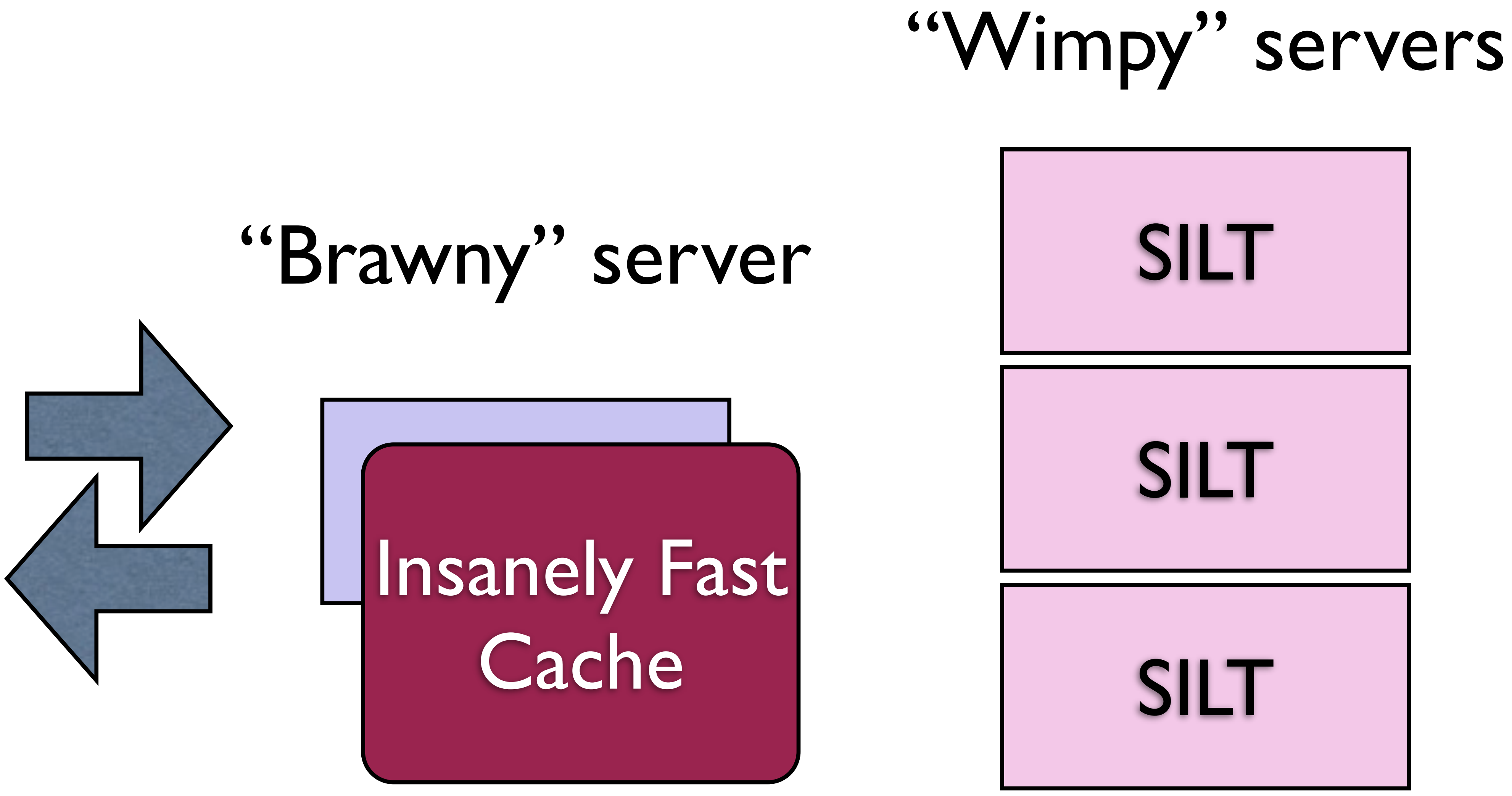
“Brawny” server

Cache

SILT

SILT

SILT



“Wimpy” servers

“Brawny” server

Insanely Fast Cache

SILT

SILT

SILT

Optimistic Cuckoo Hashing

Hashing again

- Well known, old technique, nothing new.... right?

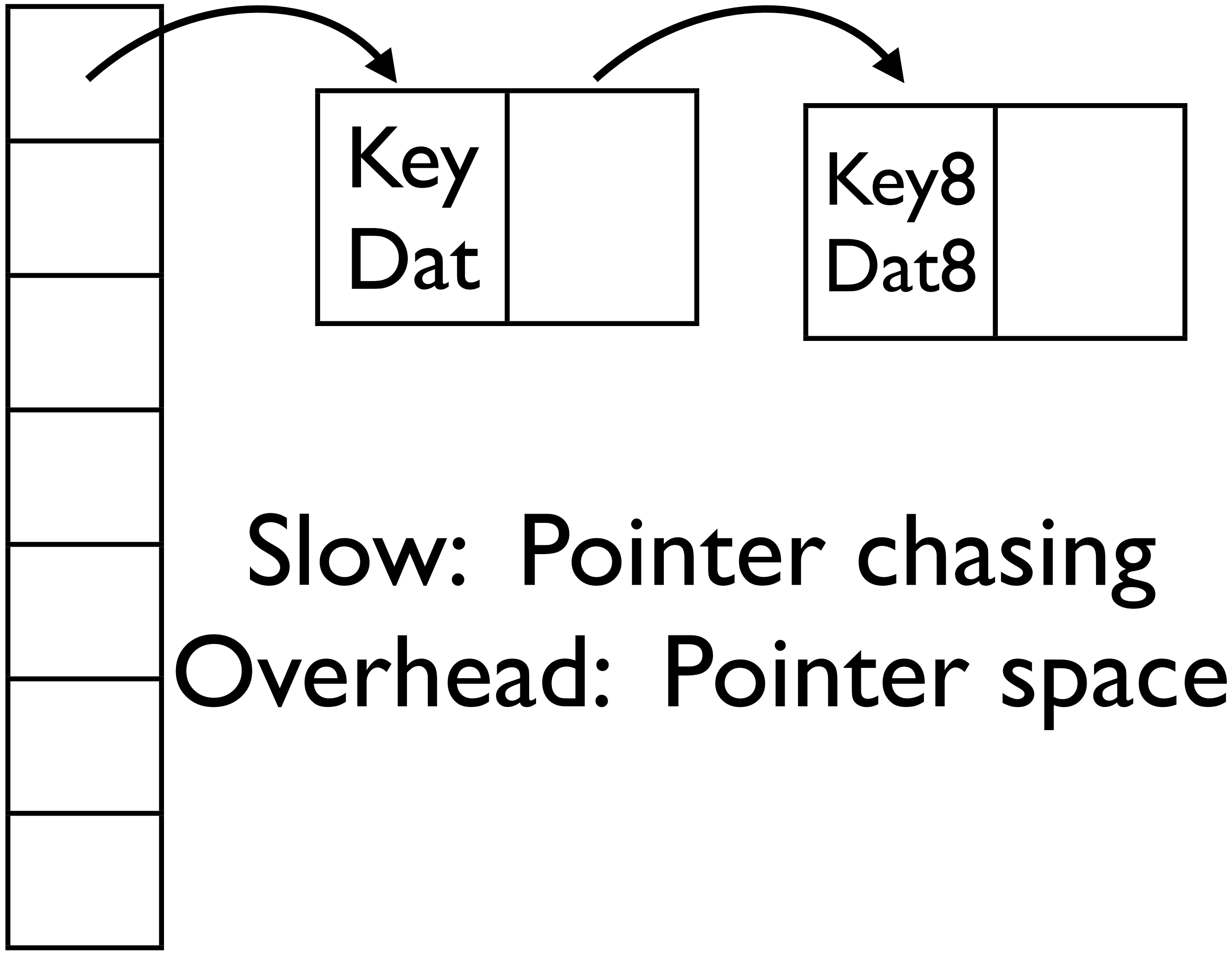
Memory efficiency vs speed

Linear Probing

key	dat
key3	dat3
key2	dat2

Wastes 50% of slots

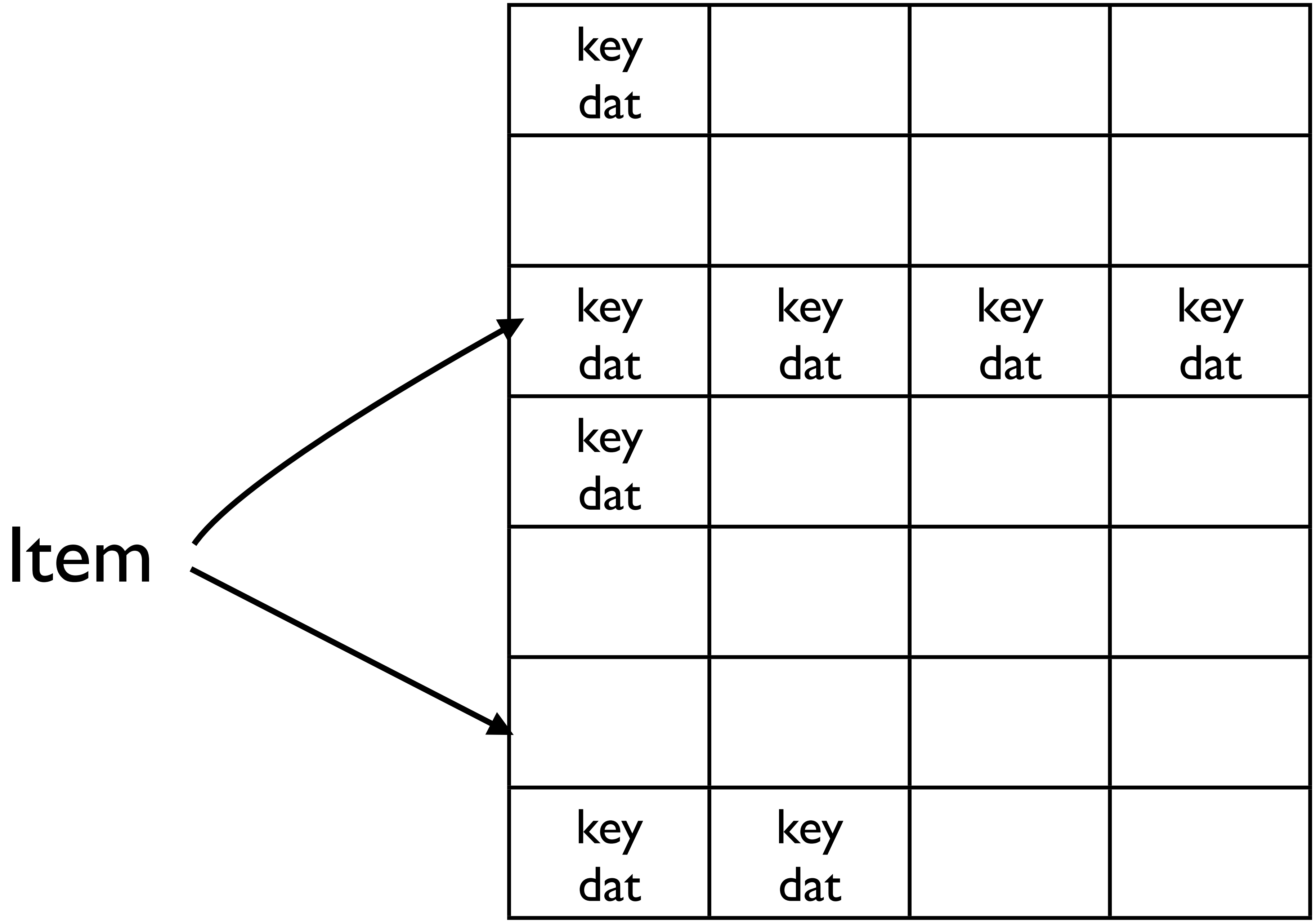
Chaining



Slow: Pointer chasing

Overhead: Pointer space

Cuckoo Hashing



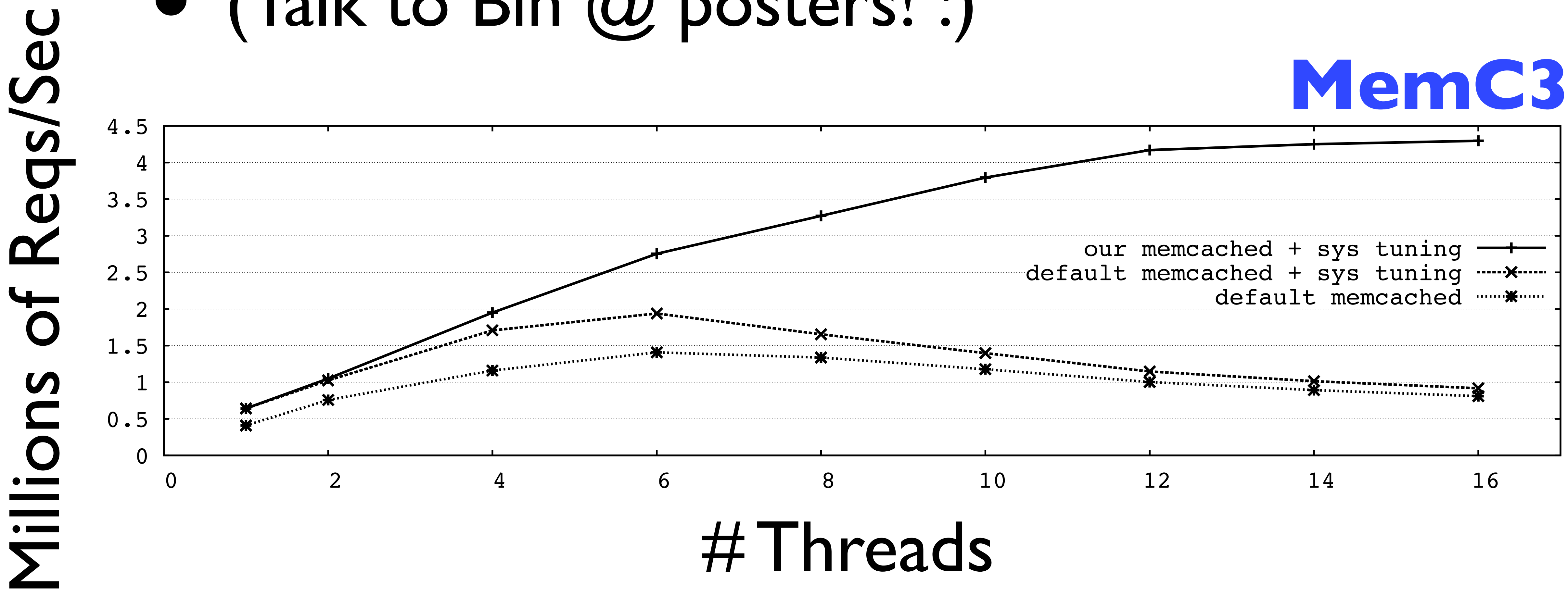
key dat			
key dat	key dat	key dat	key dat
key dat			
key dat	key dat		

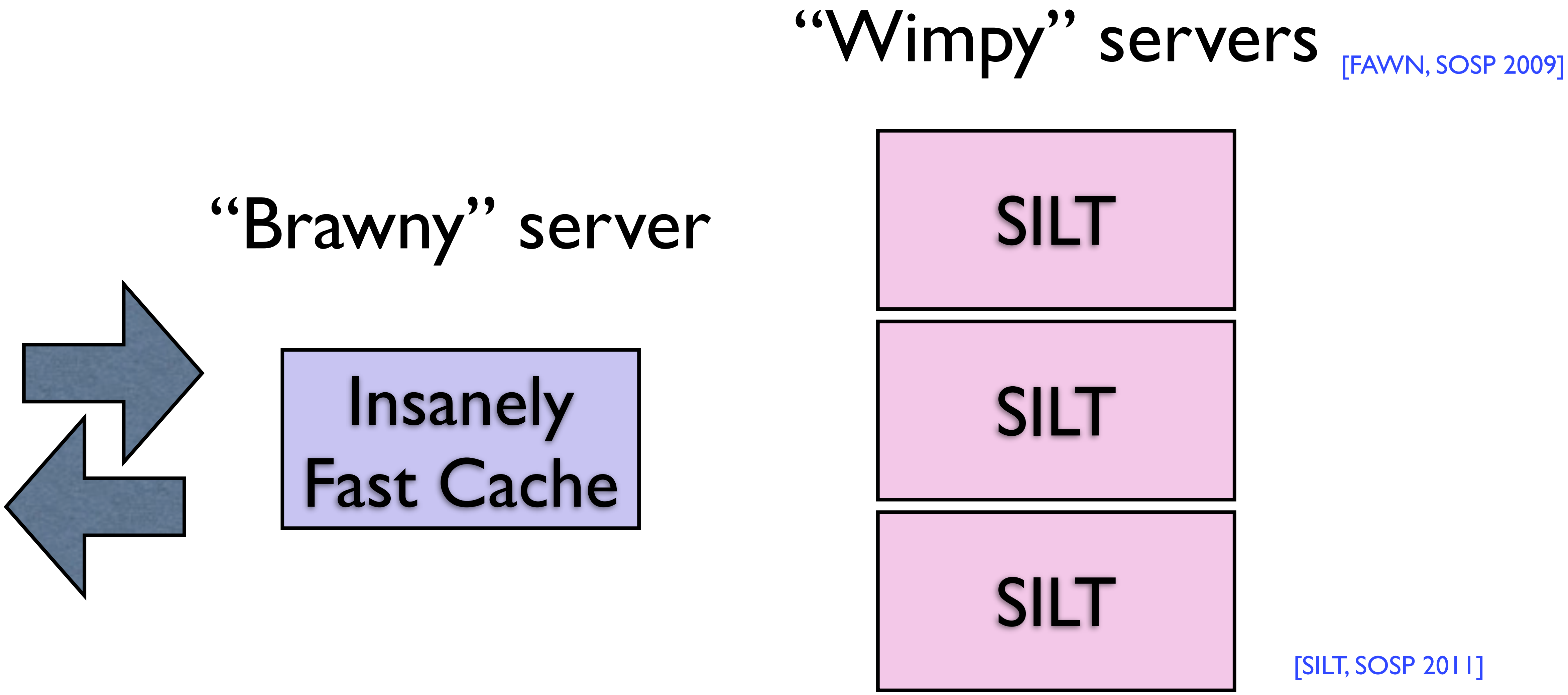
Uses 93% of slots

Fast, compact, but ... Single threaded!

Optimistic Cuckoo

- Multiple reader, lock-free, single-writer
- Applied in Memcached - *Huge* speedup
- (Talk to Bin @ posters! :)





“Wimpy” servers [FAWN, SOSP 2009]

“Brawny” server

Insanely Fast Cache

SILT

SILT

SILT

[SILT, SOSP 2011]

O(N log N) [“small cache” socc 2011]

Multi-reader parallel cuckoo hashing [under submission]

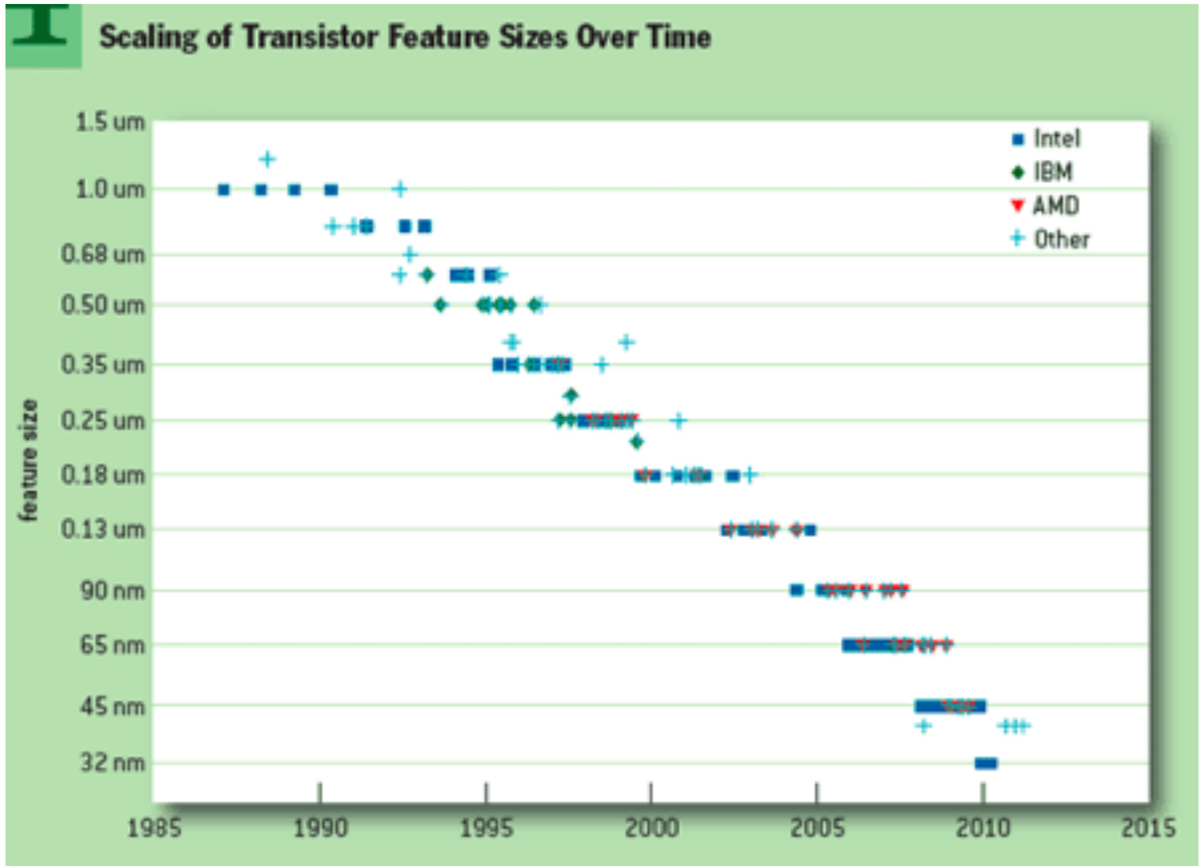
Entropy-coded tries [SILT + under submission]

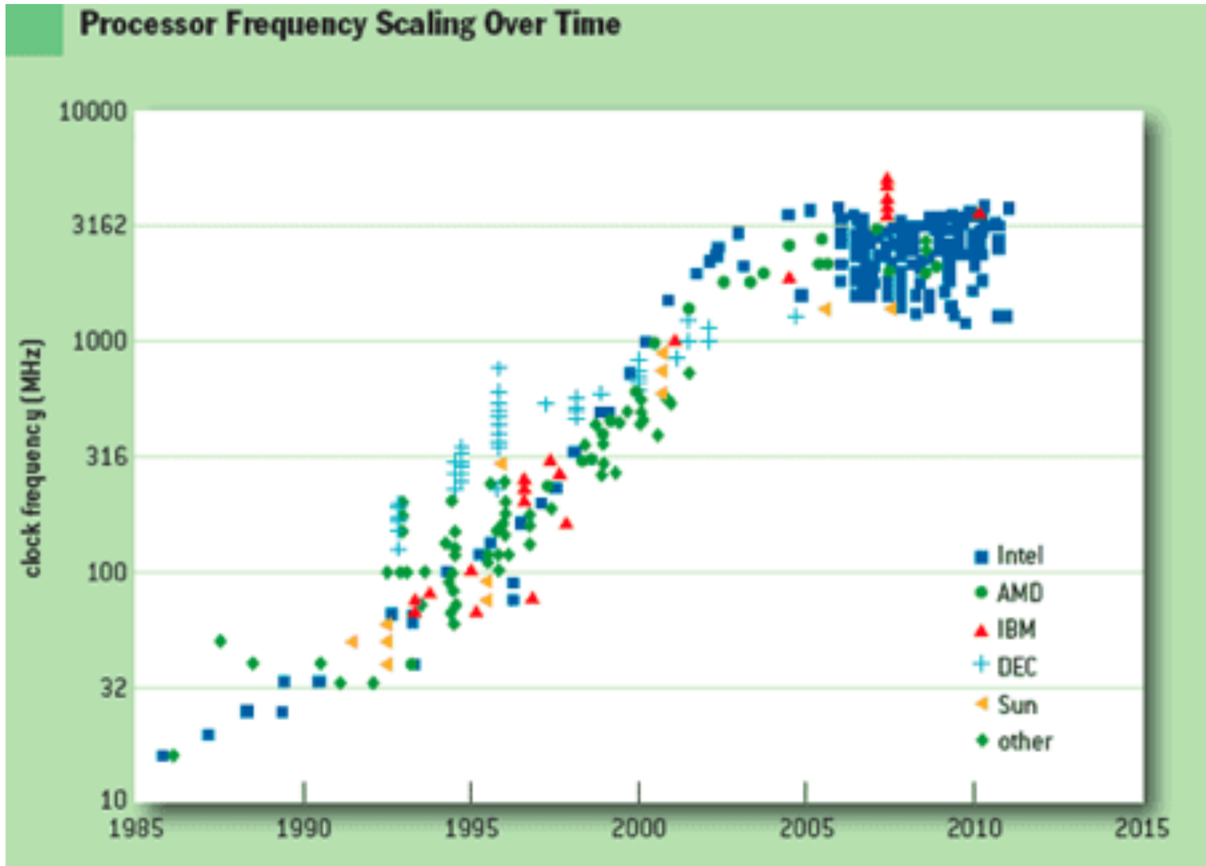
Partial-key cuckoo hashing

Cuckoo filter

Moore

Dennard





highly parallel, lower-GHz, (memory-constrained?):

Architectures, algorithms, and programming