

# Learning to Scale Out By Scaling Down

## The FAWN Project

David Andersen, Vijay Vasudevan, Michael Kaminsky\*, Michael A. Kozuch\*, 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\*\*\*, Padmanabhan Pillali\*, Dong Zhou

Carnegie Mellon University

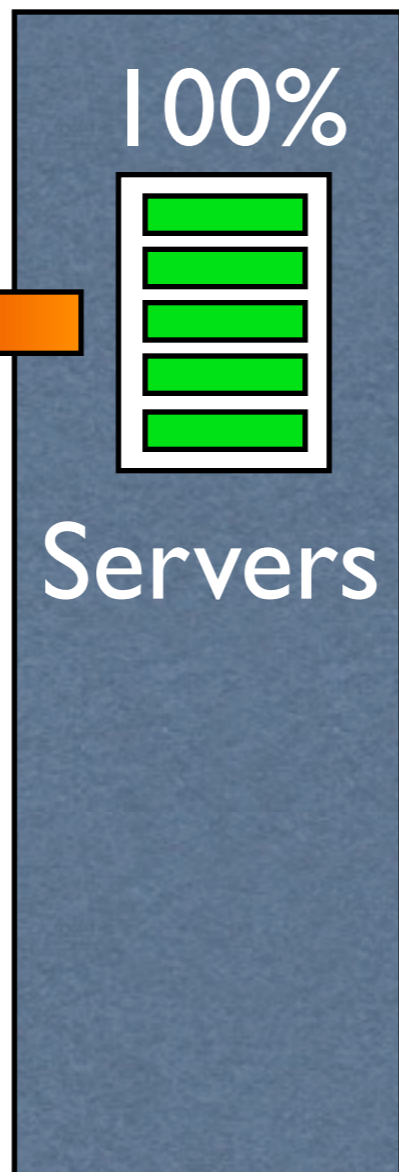
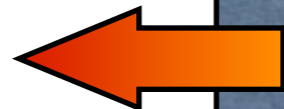
\*\* Princeton University

\*Intel Labs Pittsburgh

\*\*\* Georgia Tech

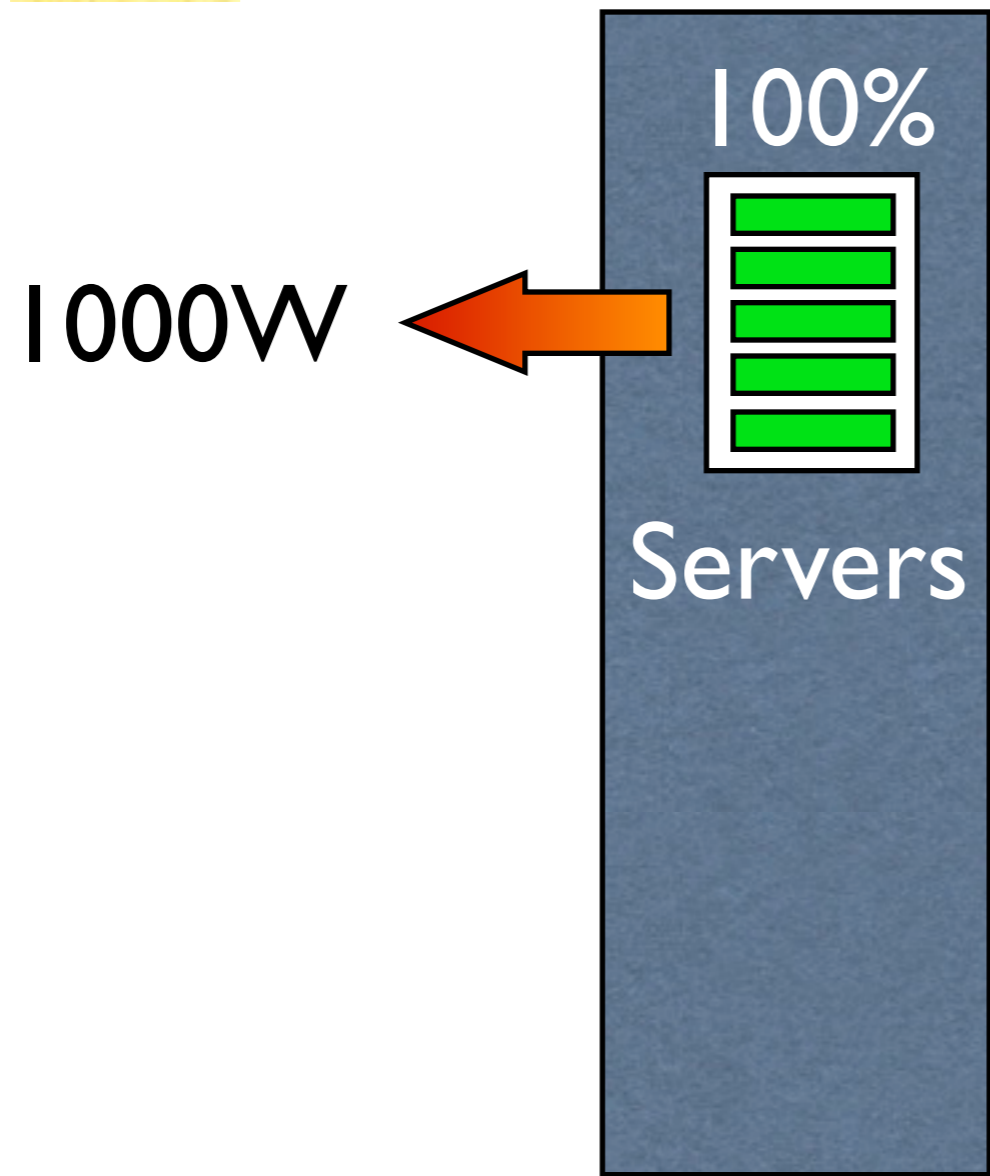


1000W





2000W



1000W

Servers

# Infrastructure: PUE

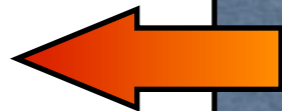
2005: 2–3

2012: ~1.1

*Leave it to industry*



1000W



100%



Servers

# Infrastructure: PUE

2005: 2–3

2012: ~1.1

*Leave it to industry*



1000W



100%

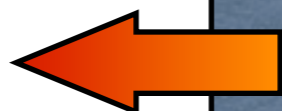


Servers

20%



750W



200W

## Proportionality

# Infrastructure: PUE

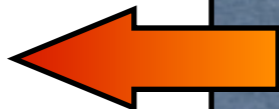
2005: 2–3

2012: ~1.1

*Leave it to industry*



1000W



100%

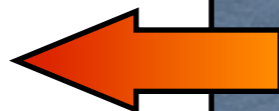


Servers

20%



750W

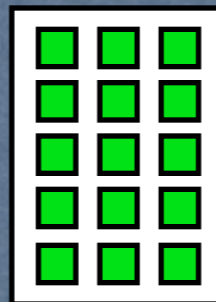


200W

## Proportionality



100%



FAWNs

## Efficiency

300W

# Infrastructure: PUE

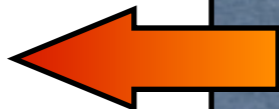
2005: 2-3

2012: ~1.1

*Leave it to industry*



1000W



100%

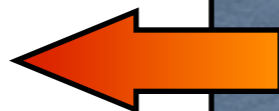


Servers

20%

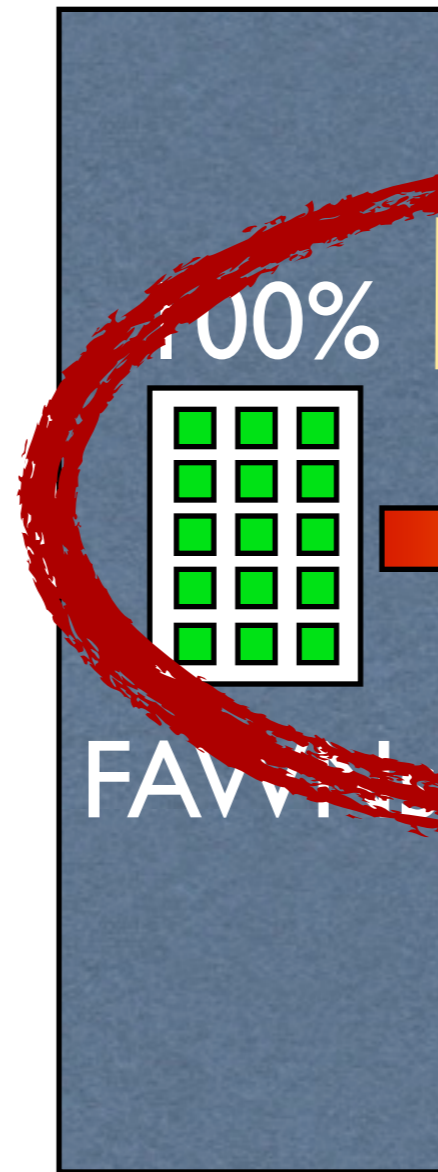


750W



200W

## Proportionality



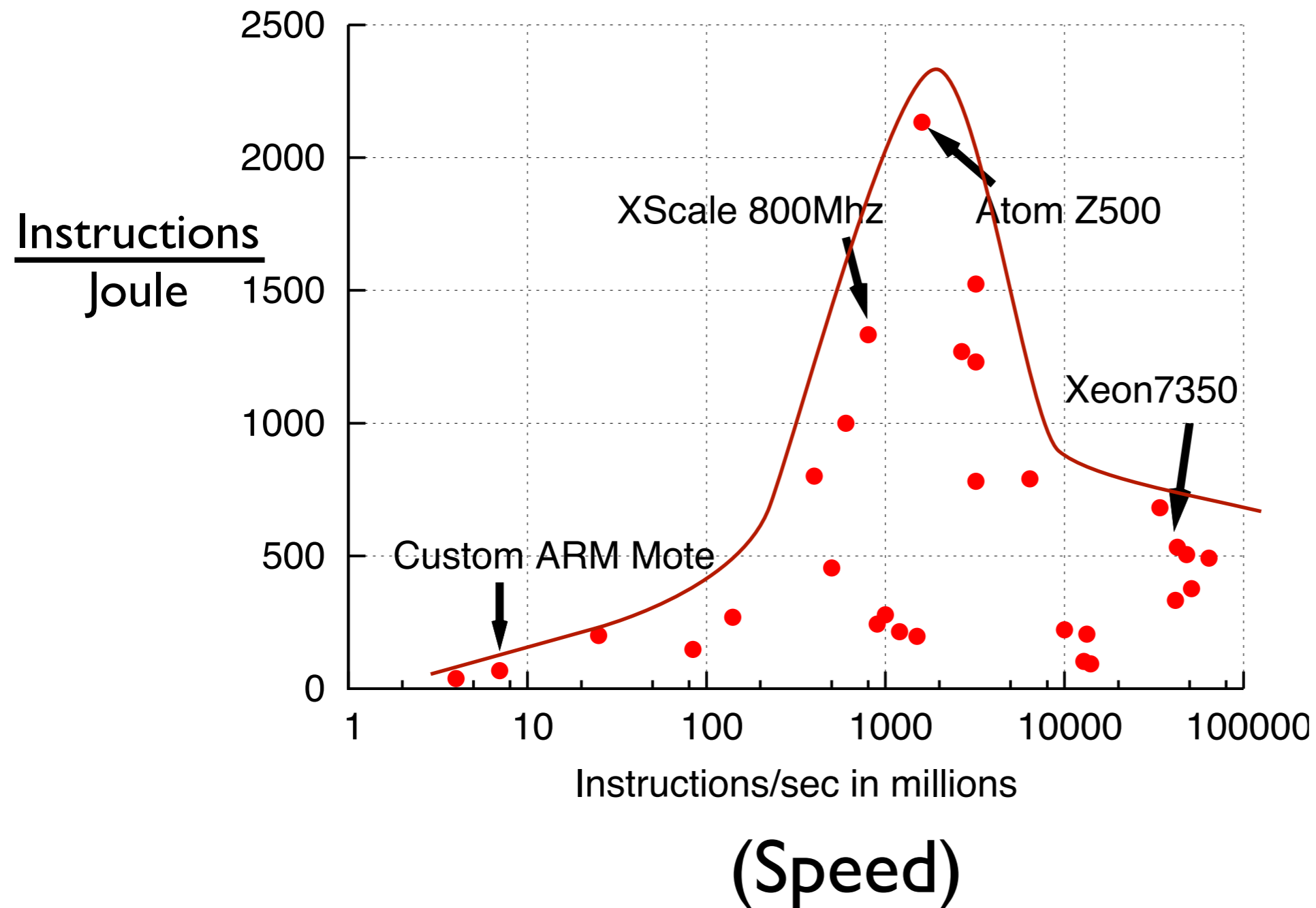
## Efficiency

300W

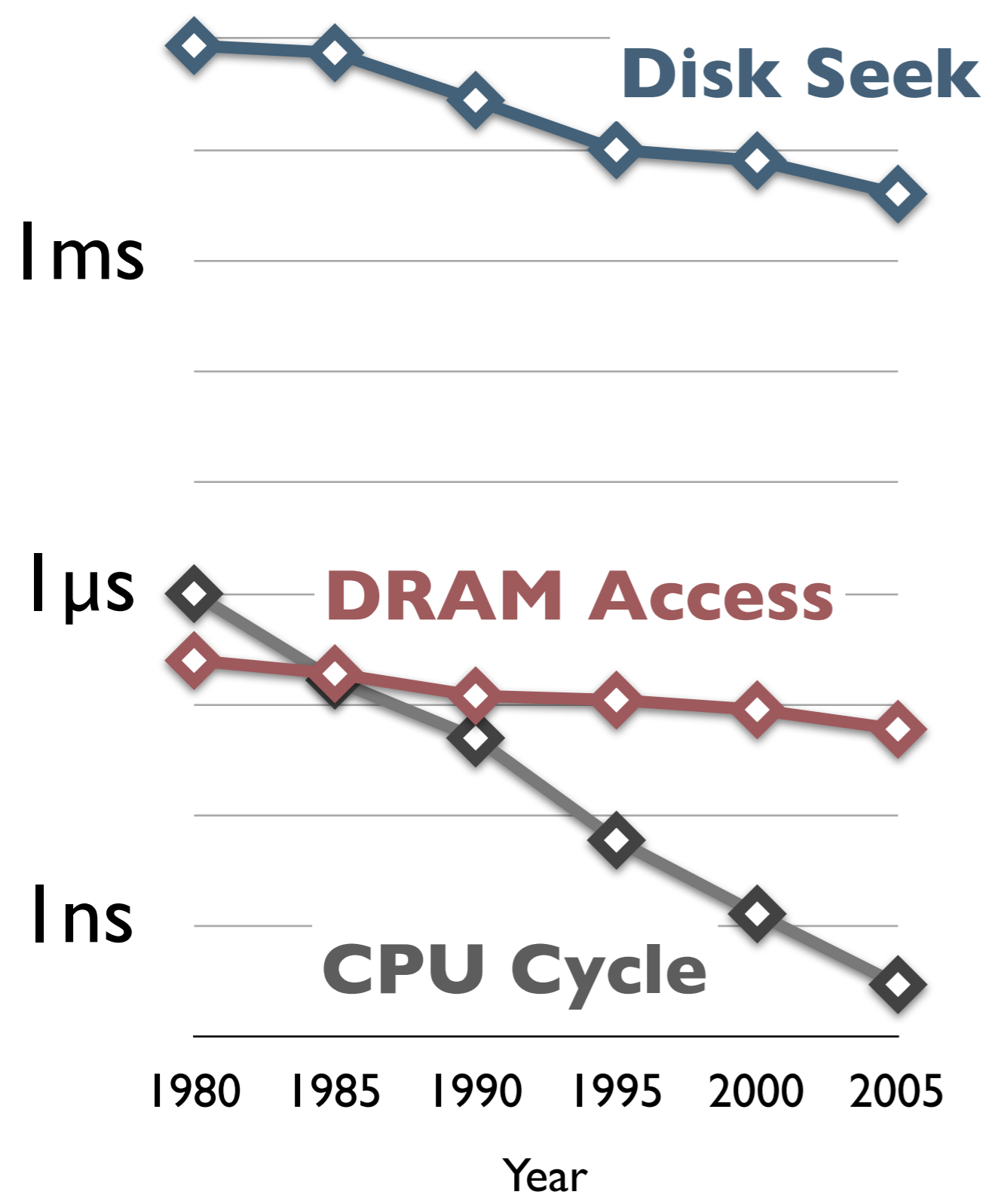


# Gigahertz is not free

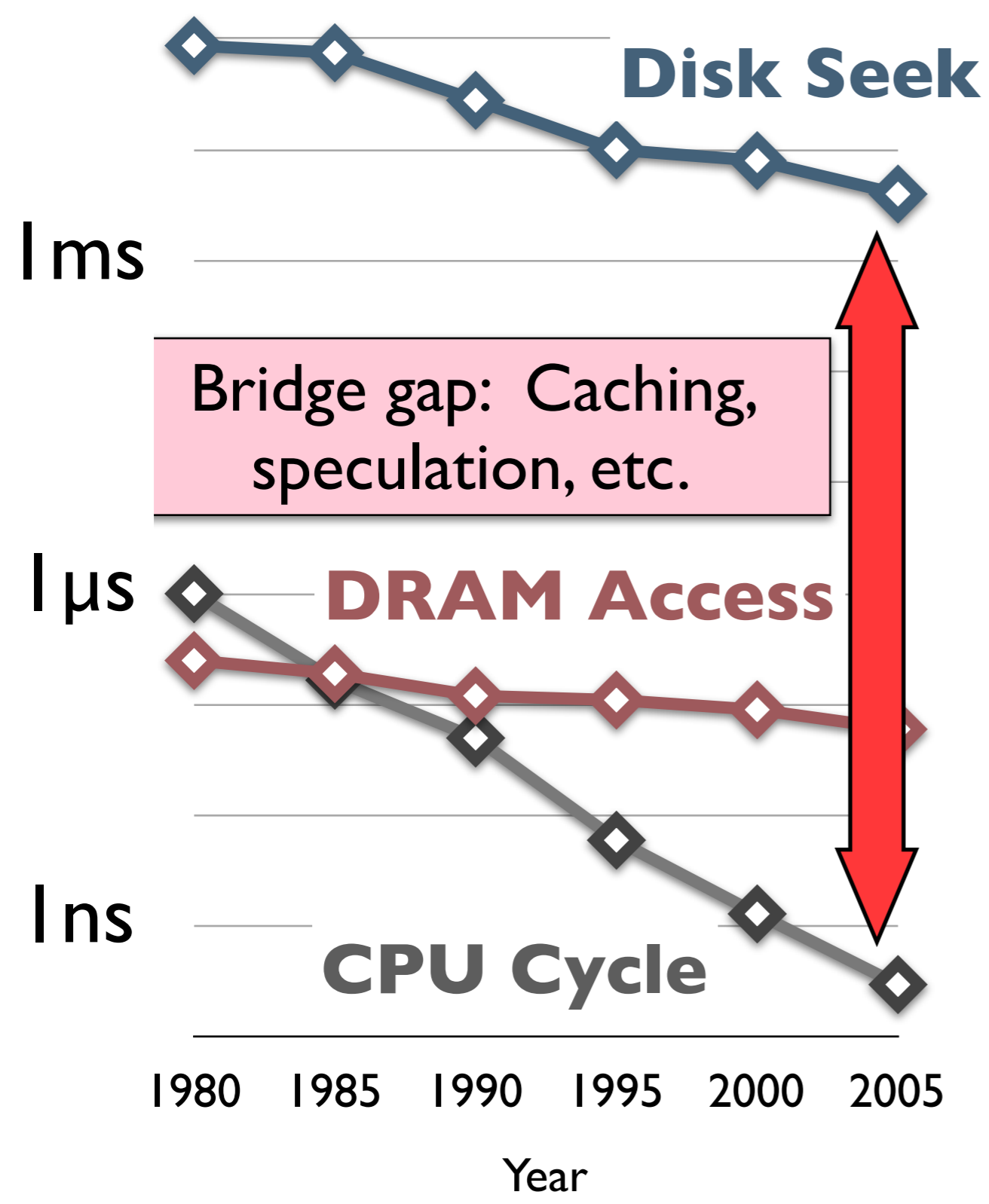
Speed and power calculated from specification sheets  
Power includes "system overhead" (e.g., Ethernet)



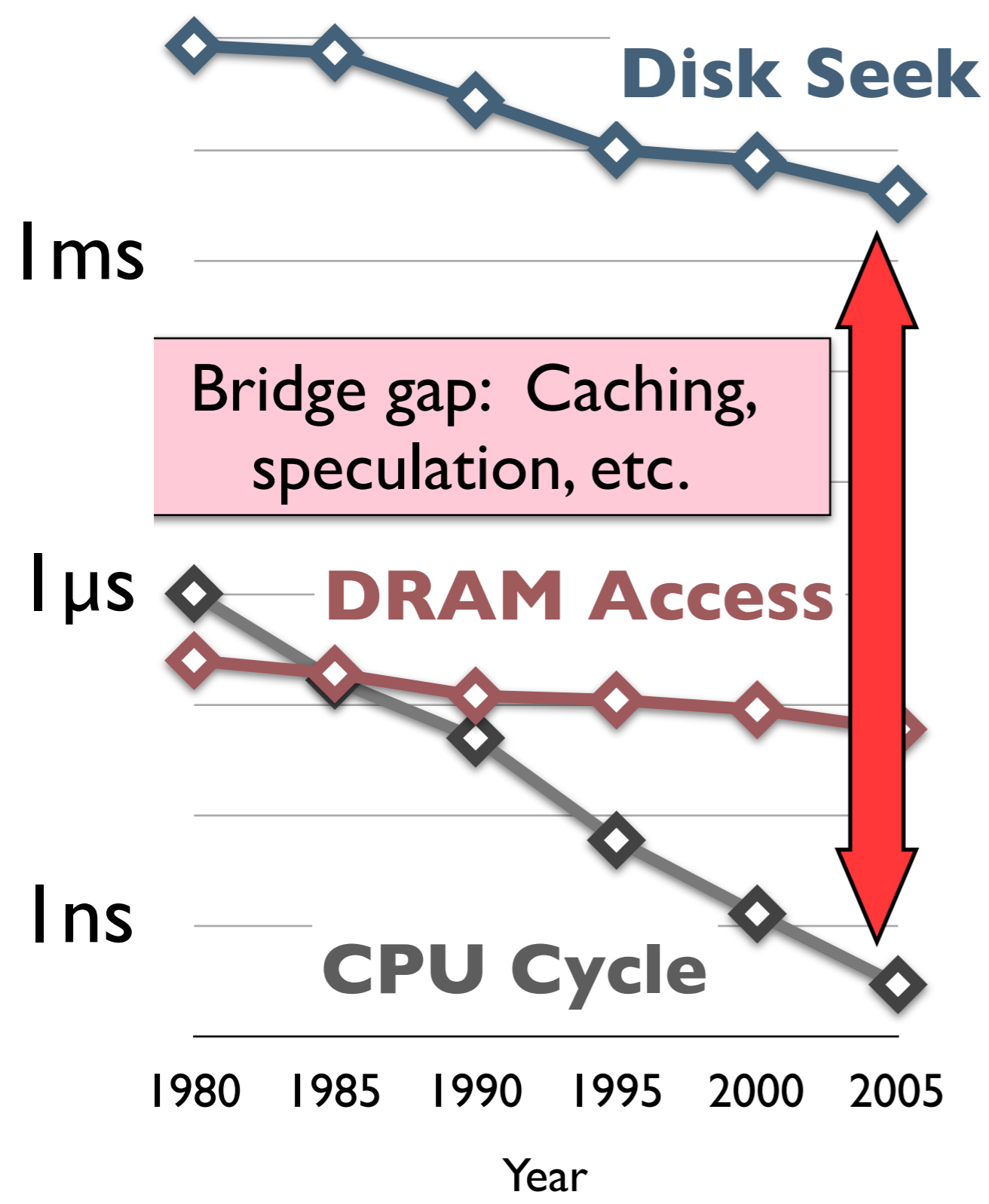
# The Memory Wall



# The Memory Wall

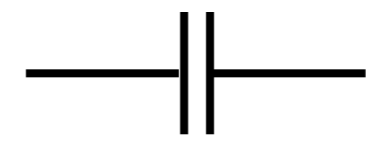


# The Memory Wall

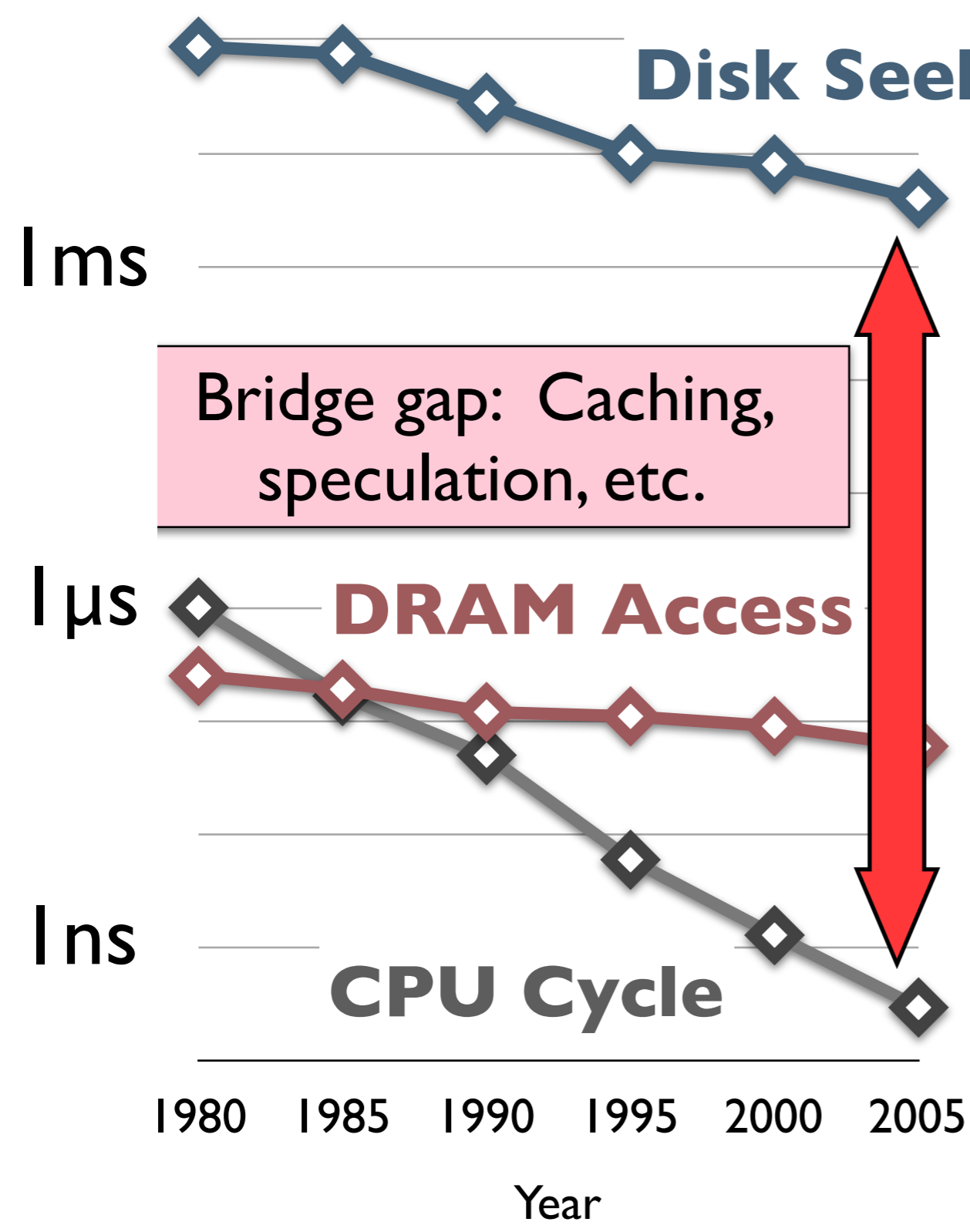


# Transistors

Have the soul of a capacitor

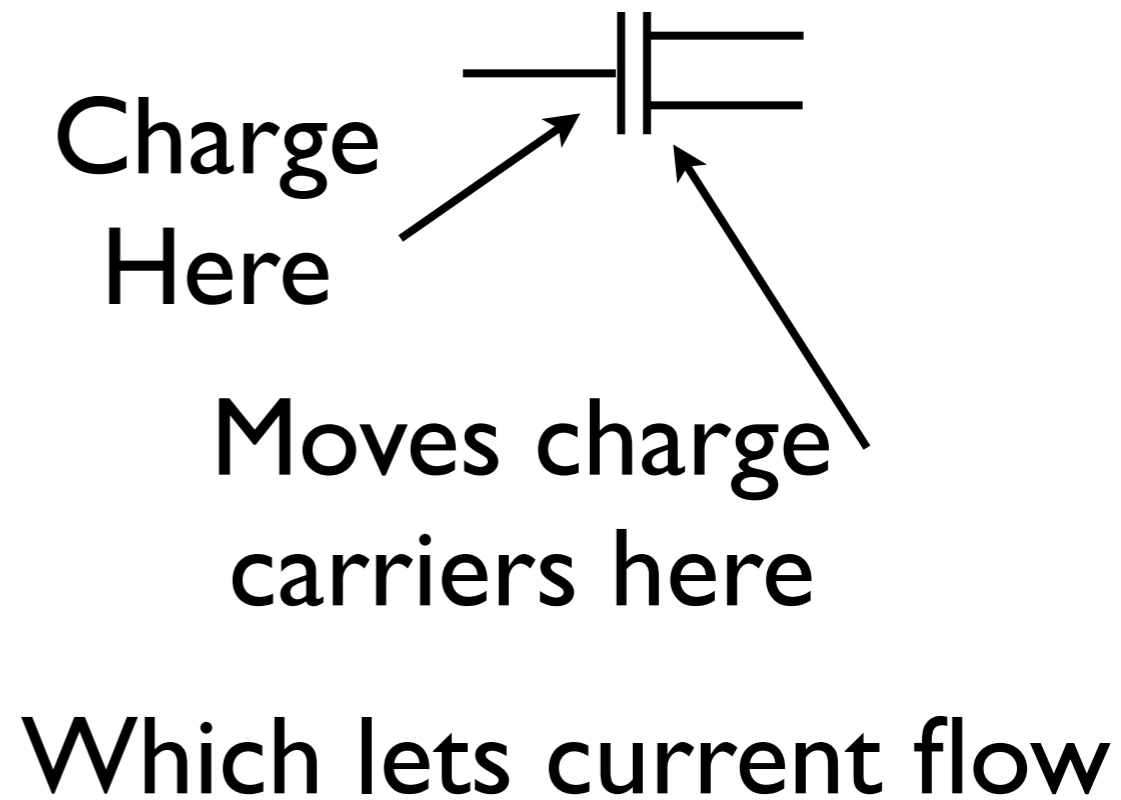


# The Memory Wall



# Transistors

Have the soul of a capacitor



**Gigahertz hurts**

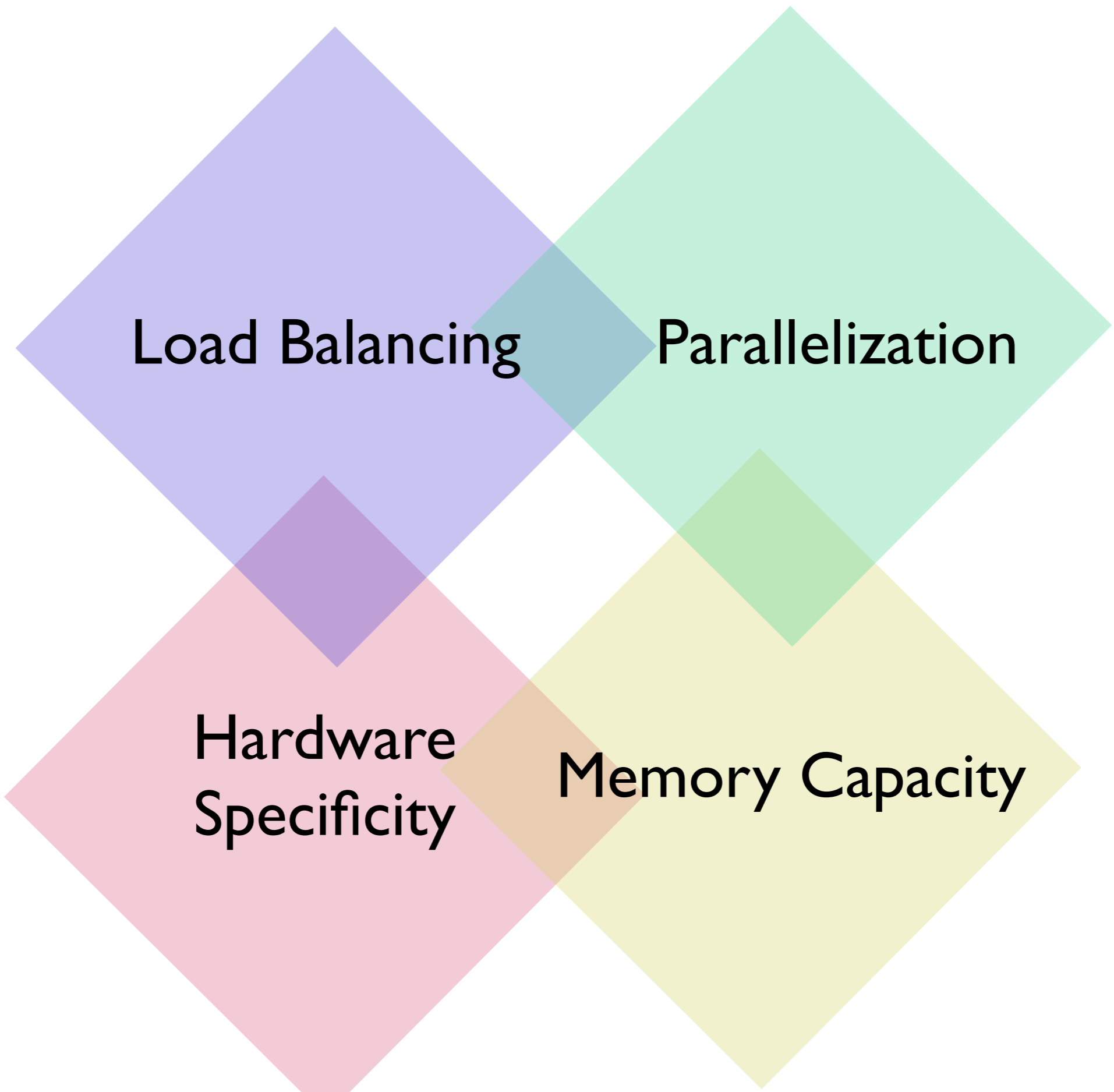
**Remember:  
Memory capacity costs you**

# “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**

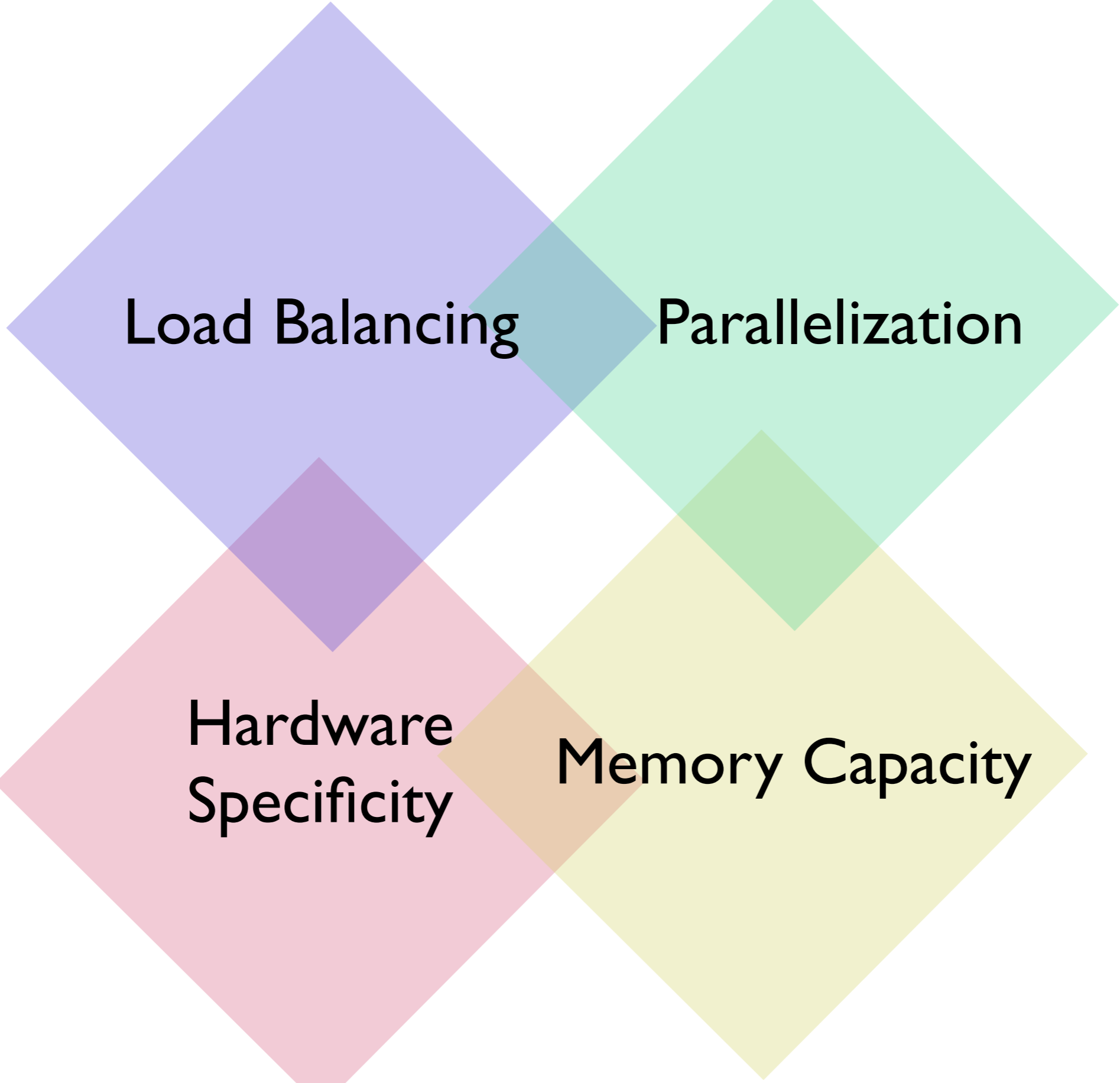
**Hardware  
Specificity**

**Memory Capacity**

Bigger Clusters

Wimpy Nodes

# The FAWN Quad of Pain



**Load Balancing**

**Parallelization**

**Hardware  
Specificity**

**Memory Capacity**

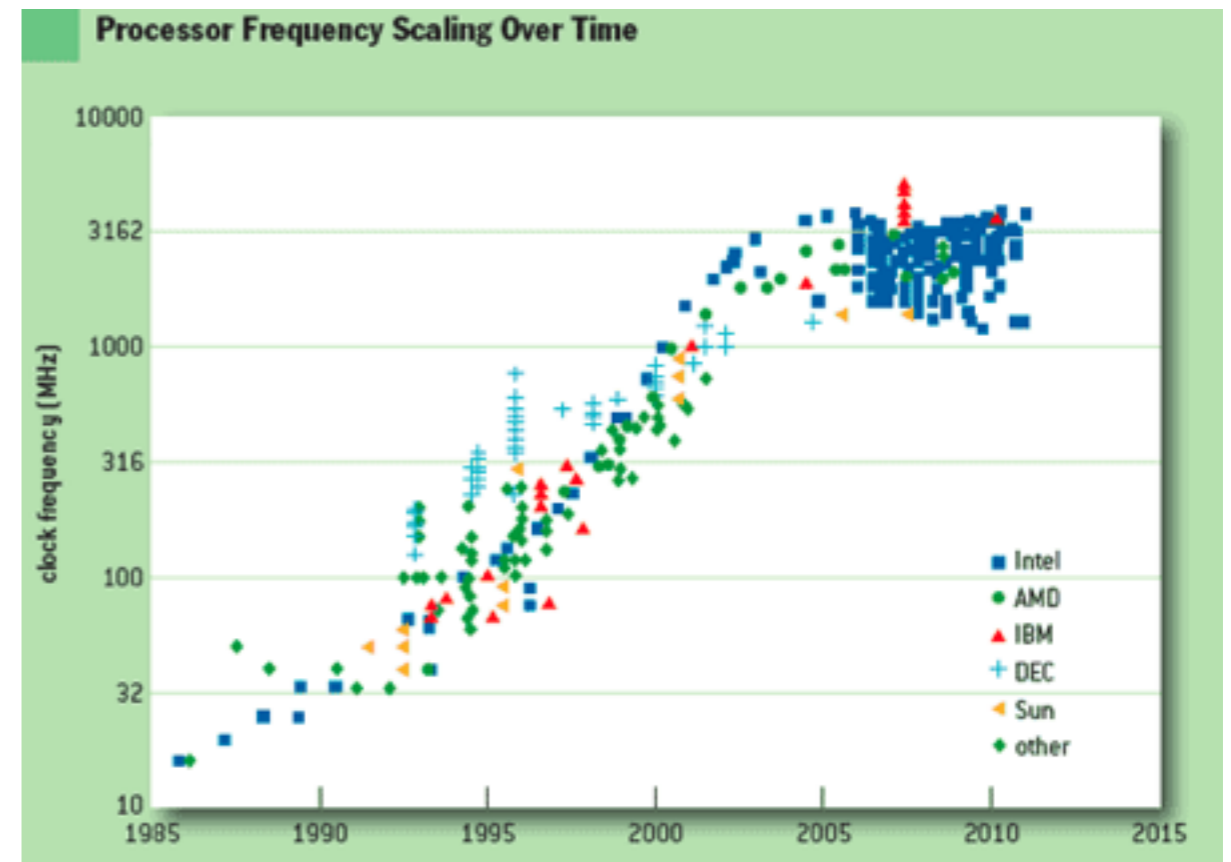
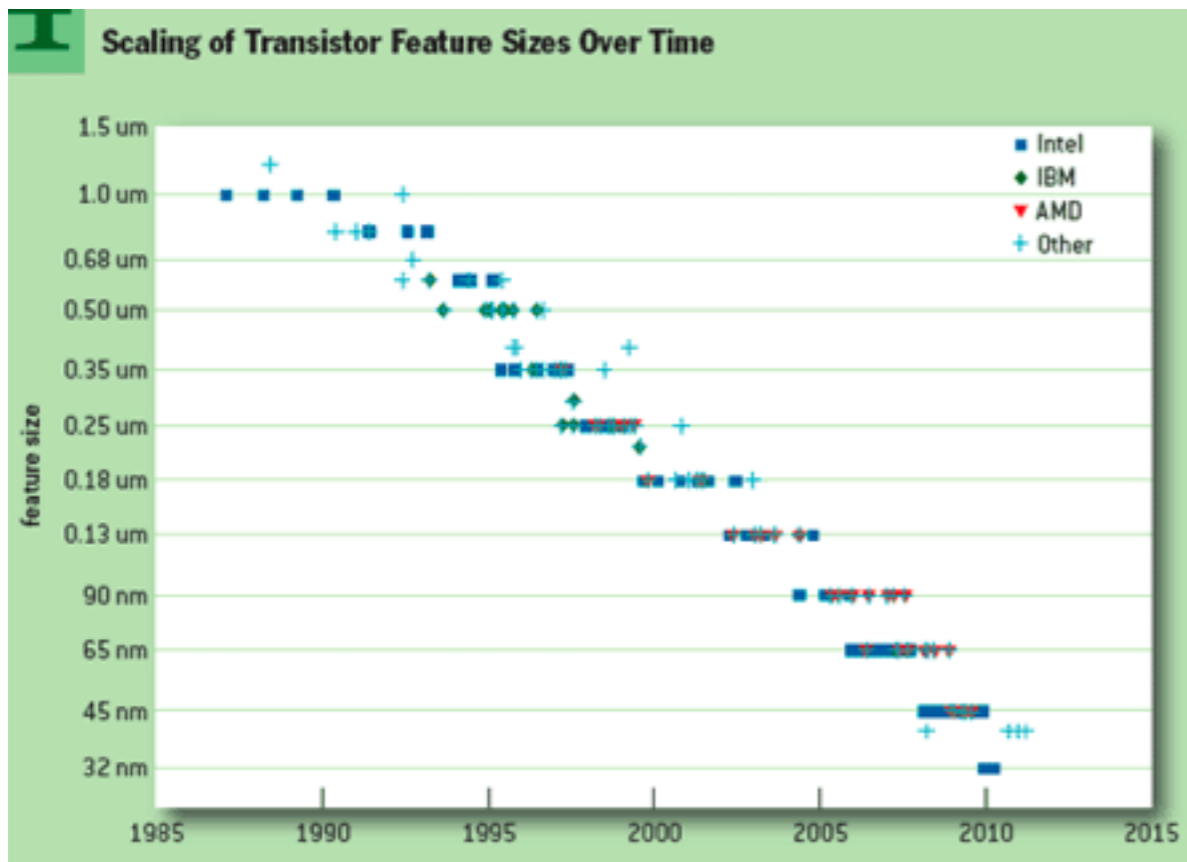
Bigger Clusters

Wimpy Nodes

# It's not just masochism

Moore

Dennard



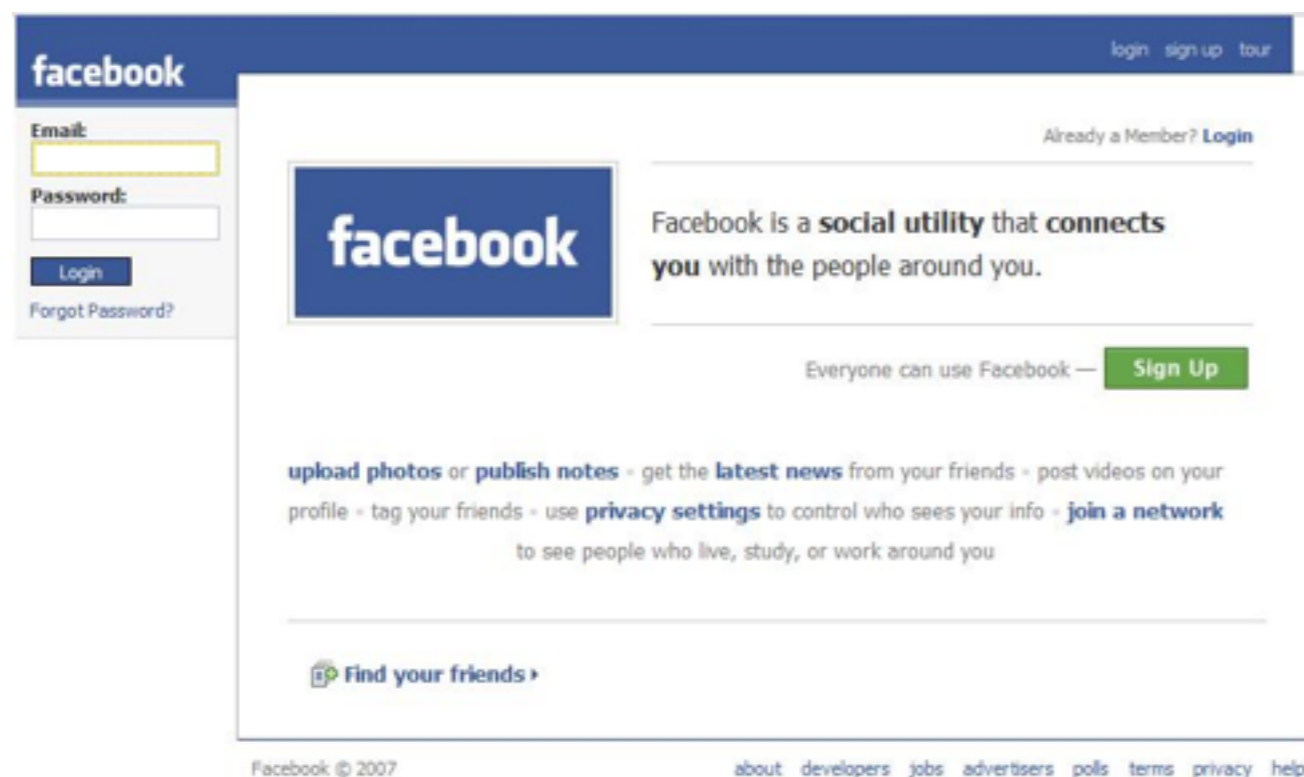
(Figures from Danowitz, Kelley, Mao, Stevenson, and Horowitz: CPU DB)

*All systems will face this challenge over time*

**FAWN:**  
It started  
with a key-value store

# Key-value storage systems

- Critical infrastructure service
- Performance-conscious
- Random-access, read-mostly, hard to cache



The screenshot shows the Facebook login page. On the left, there is a login form with fields for "Email" and "Password", a "Login" button, and a "Forgot Password?" link. The main content area features the Facebook logo, the text "Facebook is a social utility that connects you with the people around you.", and a "Sign Up" button. Below this, there are links for "upload photos or publish notes", "get the latest news", "post videos", "tag your friends", "privacy settings", and "join a network". At the bottom, there is a "Find your friends" link.



The screenshot shows the Twitter homepage. At the top, there is a search bar with the text "Search for a keyword or phrase" and a "Search" button. Below the search bar, there is a navigation bar with links for "Home", "Direct Messages", "Profile", "Settings", and "Sign Out". The main content area is divided into several sections: "See who's here" with a grid of user avatars, "Top tweets" with a list of tweets, and "New to Twitter?" with a "Let me in" button. At the bottom, there is a footer with copyright information and links for "About Us", "Contact", "Blog", "Status", "Cookies", "API", "Business", "Help", "Jobs", "Terms", "Privacy", and "Language: English".

# 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  
happy birthday dude, its been awhile!  
[Wall-to-Wall - Write on Vish's Wall](#)

---

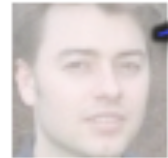
Sep 19

 **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;
```

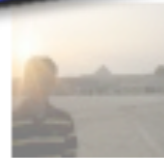
99 friends



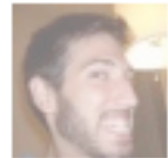
Carsten  
Varming



Timor  
Tsentsiper



Arvind  
Chari



Corey  
Iyican



John  
Bethencourt



Ram  
Ravichandran



**Dan Wendlandt wrote** at 6:47pm

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

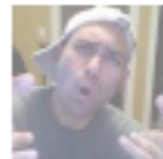
Wall-to-Wall - Write on Dan's Wall



**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

happy 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

Create a Profile Badge

# Small record, random access

The image shows a screenshot of a Facebook profile page. On the left, there is a list of 99 friends with their names and profile pictures. The main content area shows a timeline of posts from September 21st and 19th. A callout box with a blue border and a tail pointing to the top of the page contains the SQL query: `Select name, photo from users where uid=818503;`. The posts include birthday wishes from Patrick Gage Kelley, Jagan Seshadri, Vish Subramanian, and Bobby Gregg.

99 friends    See All    Sep 21

Carsten Varming    Timor Tsentsiper    Arvind Chari

Corey Iyican    John Bethencourt    Ram Ravichandran

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  
happy birthday dude, its been awhile!  
Wall-to-Wall - Write on Vish's Wall

Sep 19

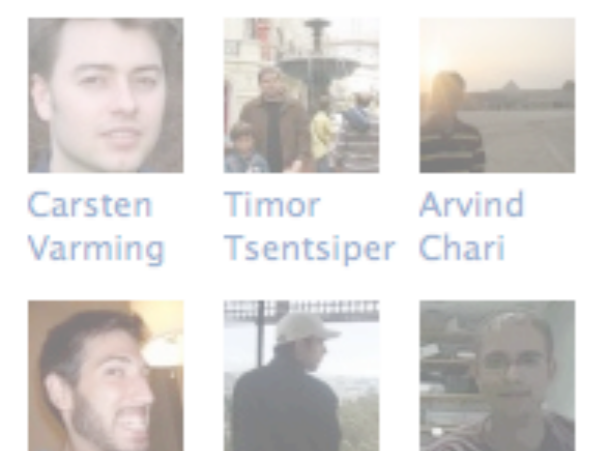
Bobby Gregg wrote at 2:22pm  
hi vijay! i'm super early but i'm bad about checking facebook regularly nowadays so i wanted to say happy birthday. let's catch up about our respective grad school woes.  
Wall-to-Wall - Write on Bobby's Wall

Create a Profile Badge



# Small record, random access

99 friends [See All](#)



Carsten Varming   Timor Tsentsiper   Arvind Chari  
Corey Iyica   John Bethenco   Ram Ravichandran

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](#)

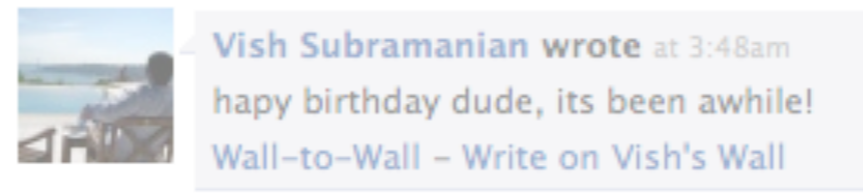
---



**Jagan Seshadri wrote** at 1:50pm  
...e's so much more...  
[Wall](#)

Select name, photo from users where uid=468883;

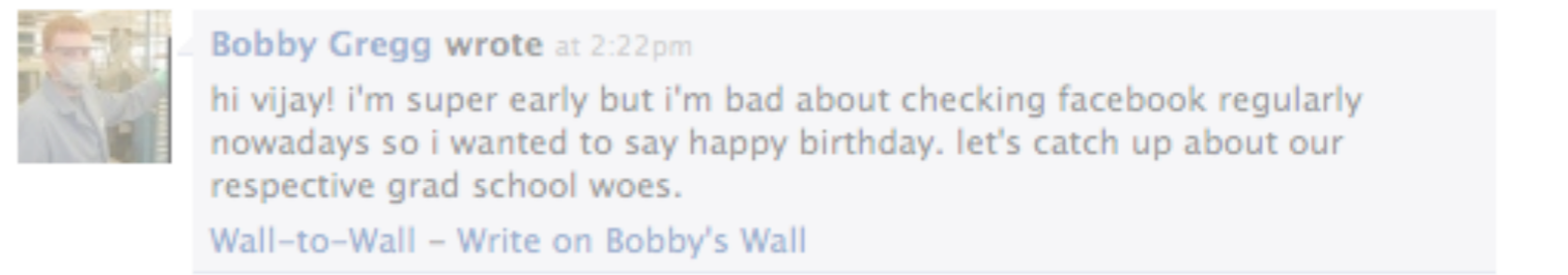
[Create a Profile Badge](#)



**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

The image shows a screenshot of a Facebook profile page. On the left, there is a '99 friends' section with a grid of profile pictures and names: Carsten Varming, Timor Tsentsiper, Arvind Chari, Corey Iyican, John Bethencourt, and Rajendra. The main content area shows a feed of posts from September 21st and 19th. Two callout boxes with SQL queries are overlaid on the page:

- A callout box pointing to a post by Dan Wendlandt: `Select wallpost from posts where pid=13821828188;`
- A callout box pointing to a post by Patrick Gage: `Select name, photo from users where uid=124111;`

Other visible posts include: 'Oh! birthday!', 'Happy birthday Vij! 24 and there's so much more...', 'hapy birthday dude, its been awhile!', and '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.'

# Small record, random access



99 friends See All

Carsten Arvind

Corey

Create a Profile Badge

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

Wall-to-Wall - Write on Iagan's Wall

Wall-to-Wall - Write on Vish's Wall

Wall - Write on Bobby's Wall

Oh! birthday!

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

hapy birthday, dude, its been awhile!

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

```
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=124566;
```

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

```
Select name,photo from users where uid=12223;
```

```
Select wallpost from posts where pid=12314144887;
```

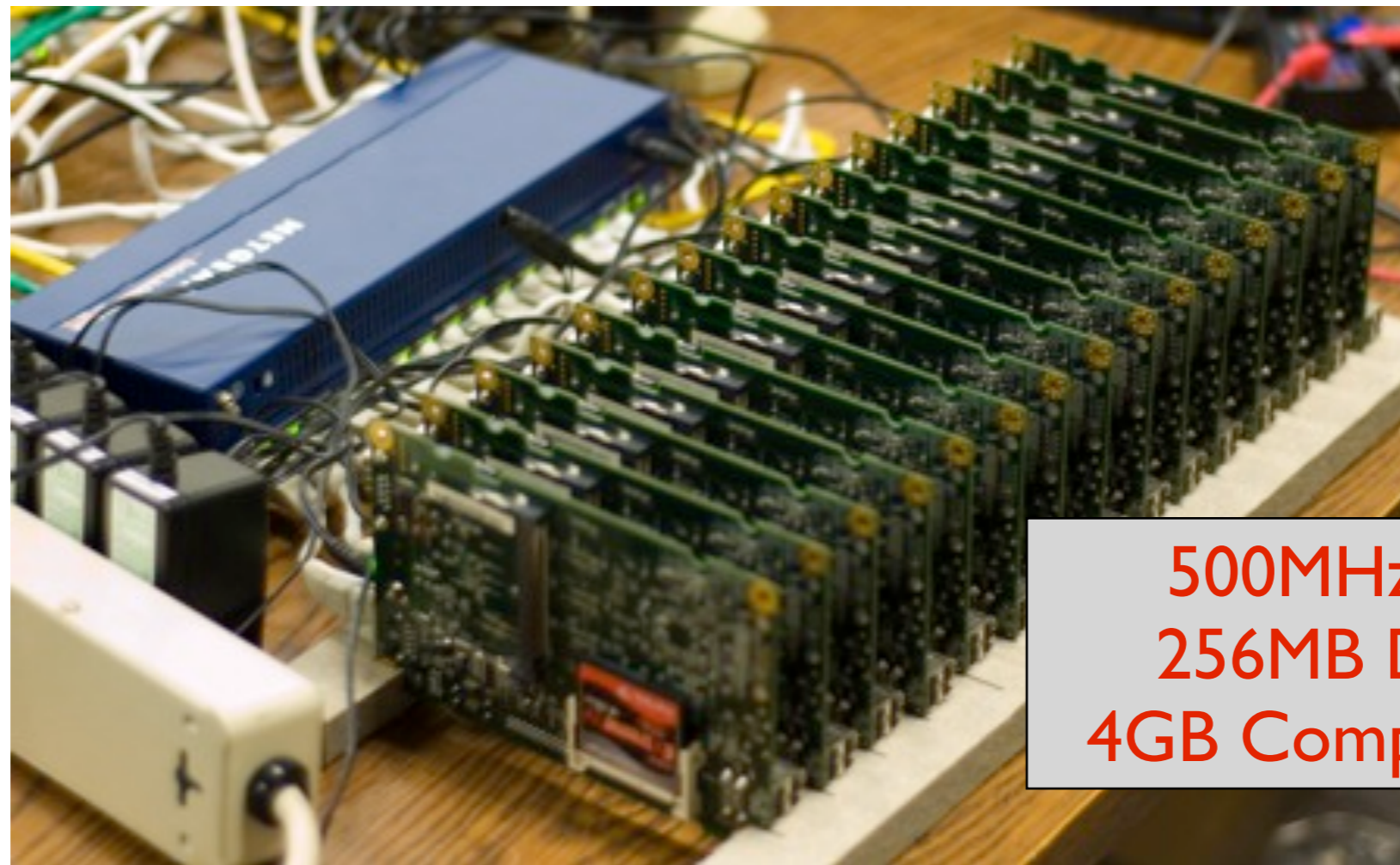
```
Select name,photo from users where uid=007788;
```

```
Select wallpost from posts where pid=738838402;
```

```
Select name,photo from users where uid=357845;
```

# FAWN-DS and -KV: Key-value Storage System

Goal: improve **Queries/Joule**



500MHz CPU  
256MB DRAM  
4GB CompactFlash

# FAWN-DS and -KV: Key-value Storage System

Goal: improve **Queries/Joule**

## Unique Challenges:

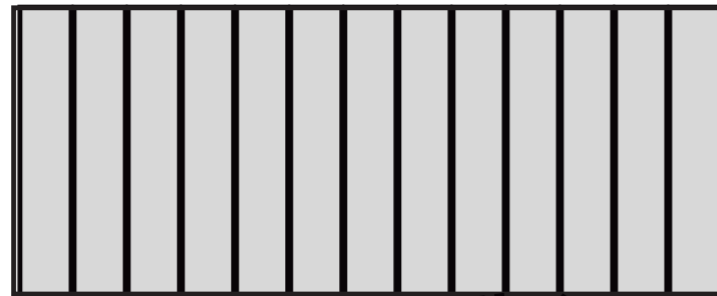
- Wimpy CPUs, limited DRAM
- Flash poor at small random writes
- Sustain performance during membership changes



256MB DRAM  
4GB CompactFlash

# Avoiding random writes

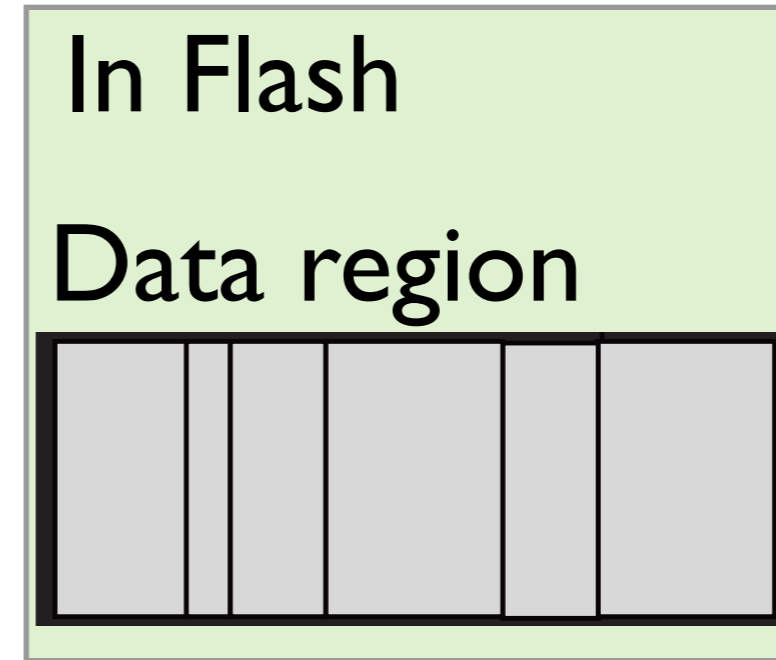
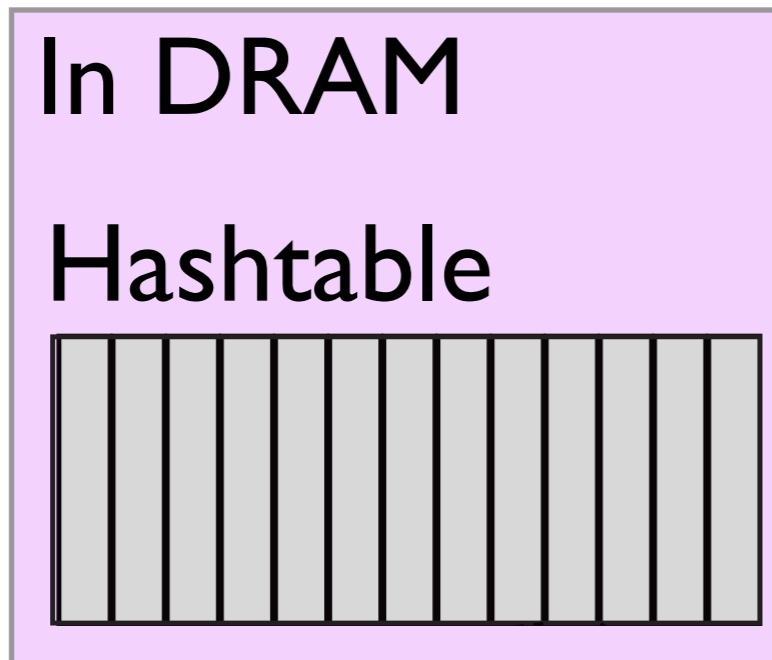
Hashtable



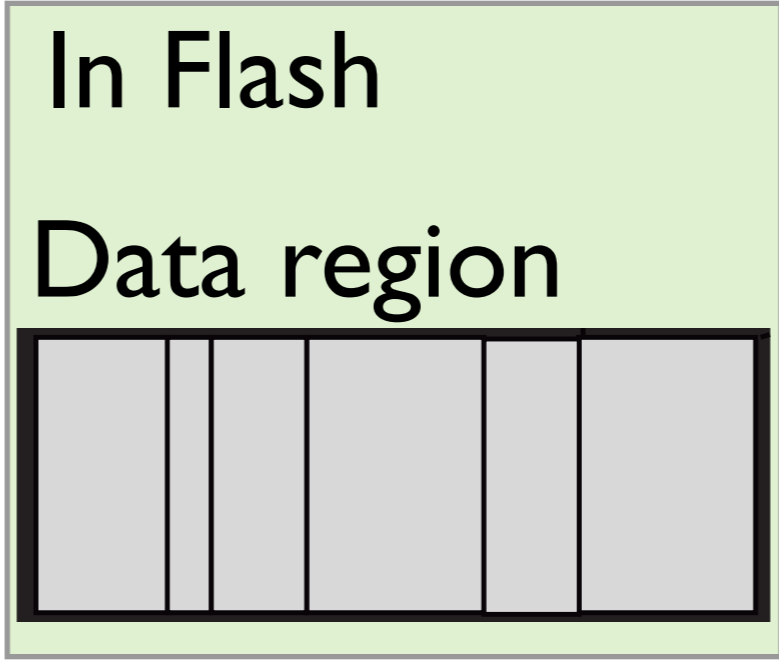
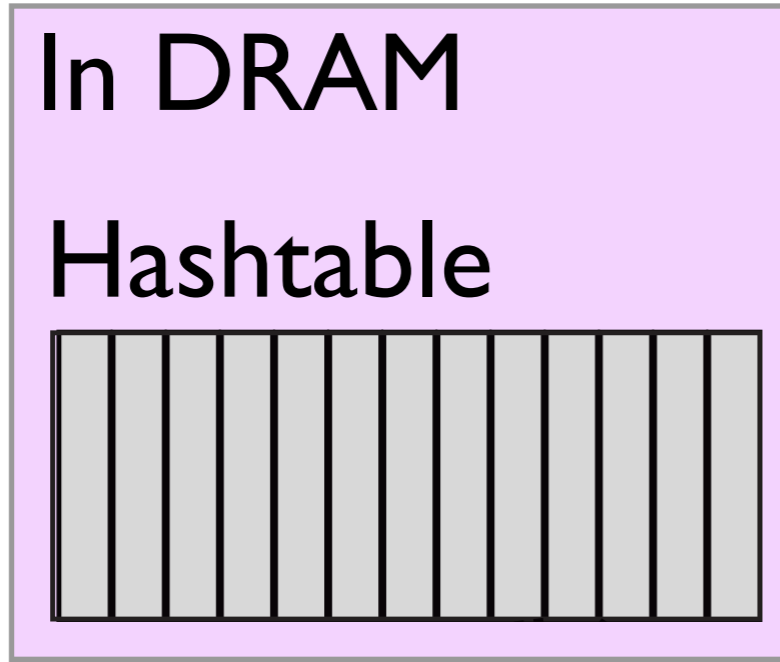
Data region



# Avoiding random writes



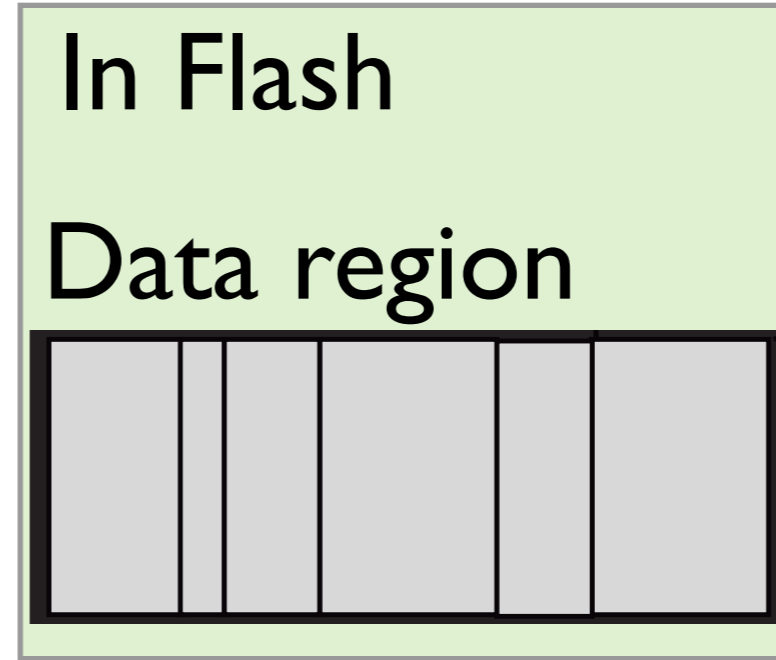
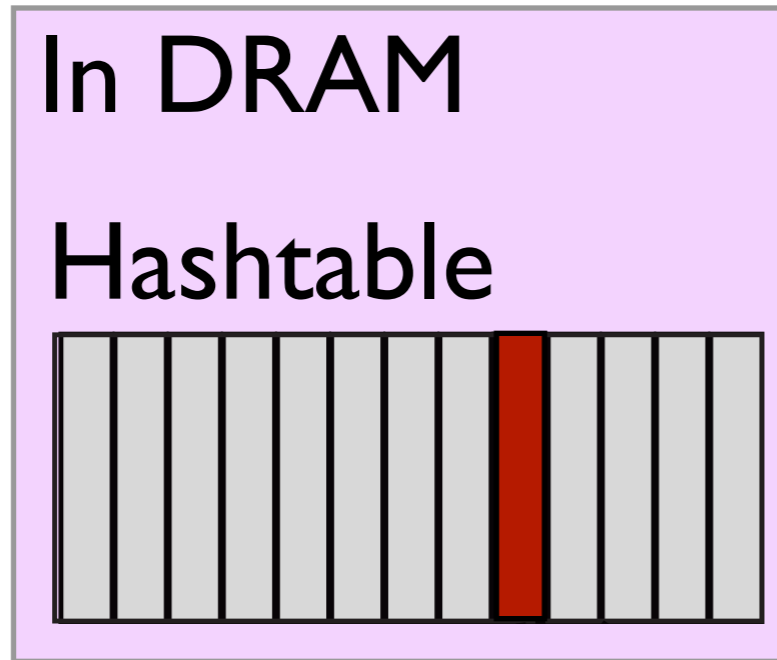
# Avoiding random writes



Put  $K_i, V$

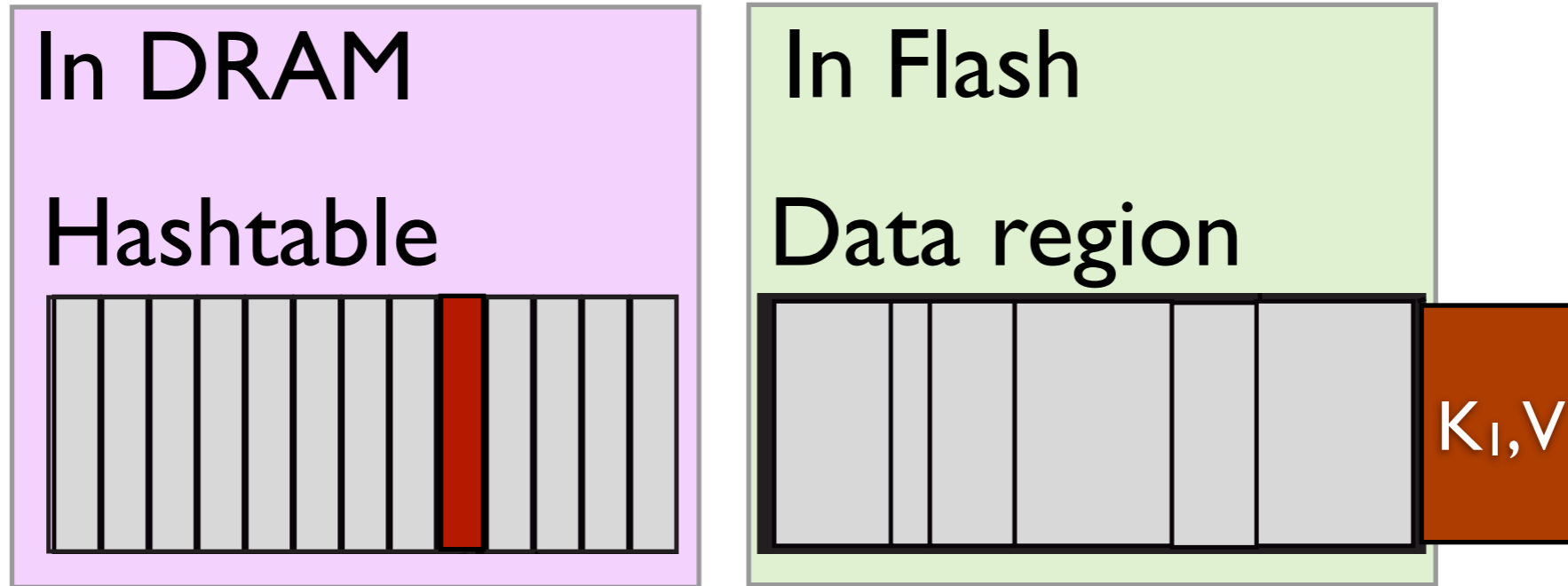


# Avoiding random writes



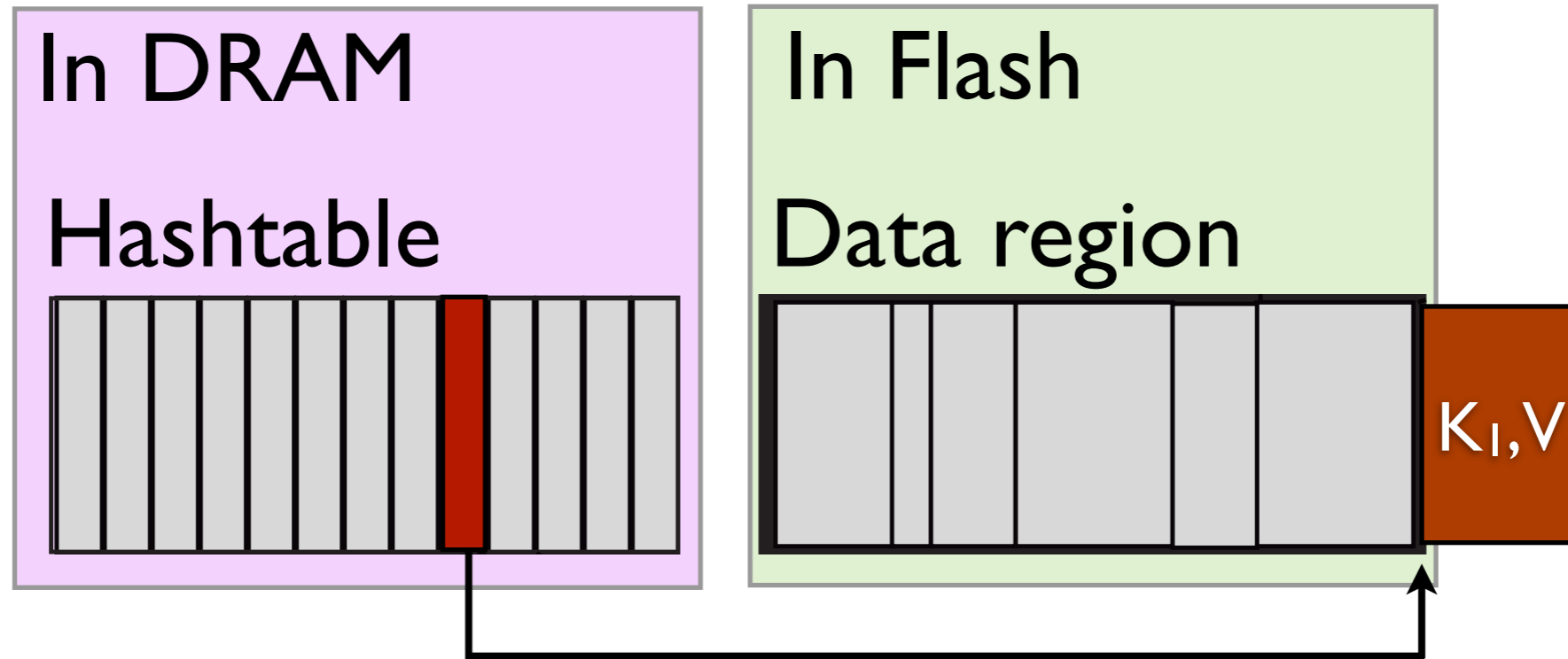
Put  $K_i, V$

# Avoiding random writes



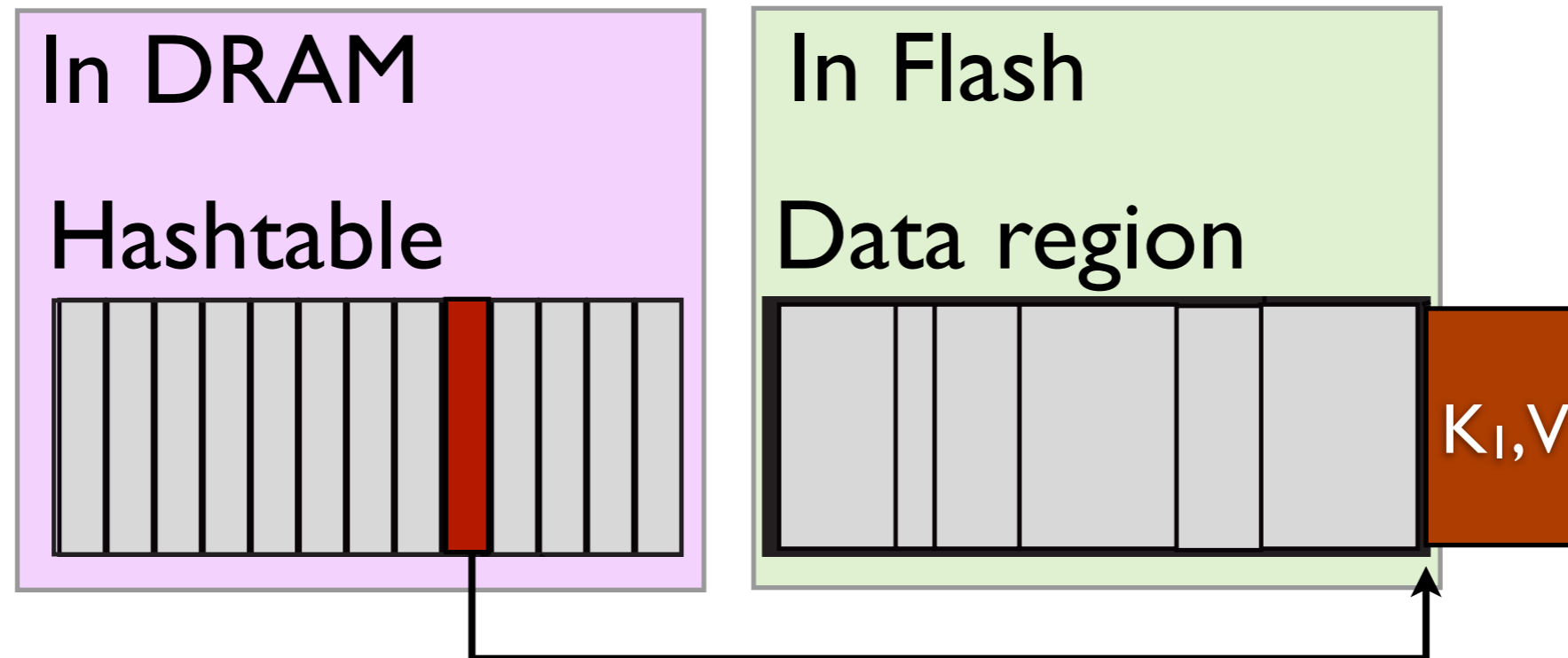
Put

# Avoiding random writes



Put

# Avoiding random writes



Put

**All writes to Flash are sequential**

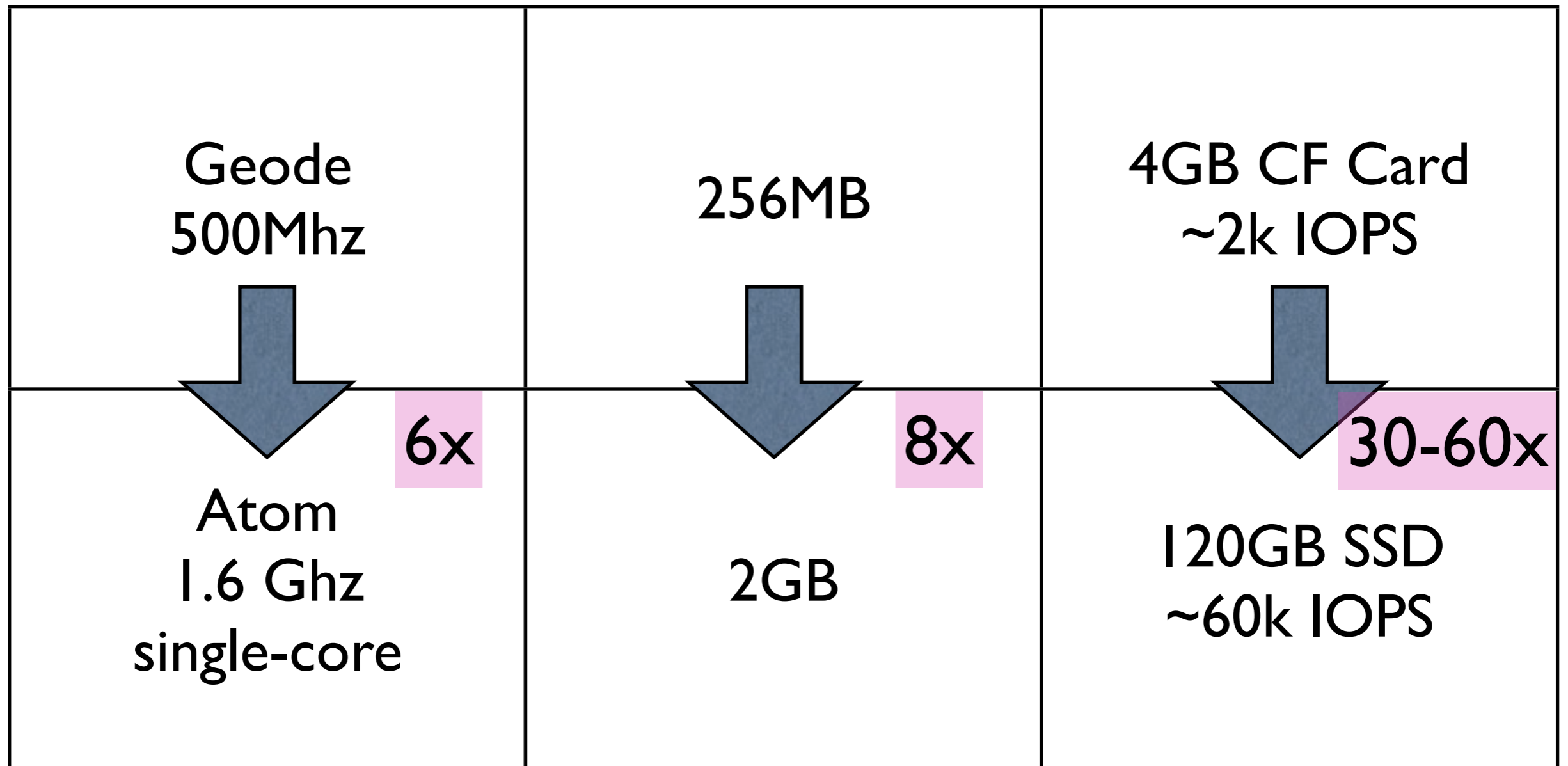
# Research Example

- Developed DRAM-efficient system to find location on flash
  - (“Partial-key hashing”) 2008-9
- We’ve continued this since then:
  - Partial-key cuckoo hashing 2011
  - Optimistic concurrent cuckoo hashing 2012

# Evaluation Takeaways

- 2008: FAWN-based system 6x more efficient than traditional systems
- Partial-key hashing enabled memory-efficient DRAM index for flash-resident data
- Can create high-performance, predictable storage service for small key-value pairs

# And then we moved to Atom + SSD



FAWN-DS

FAWN-KV

Small Cache

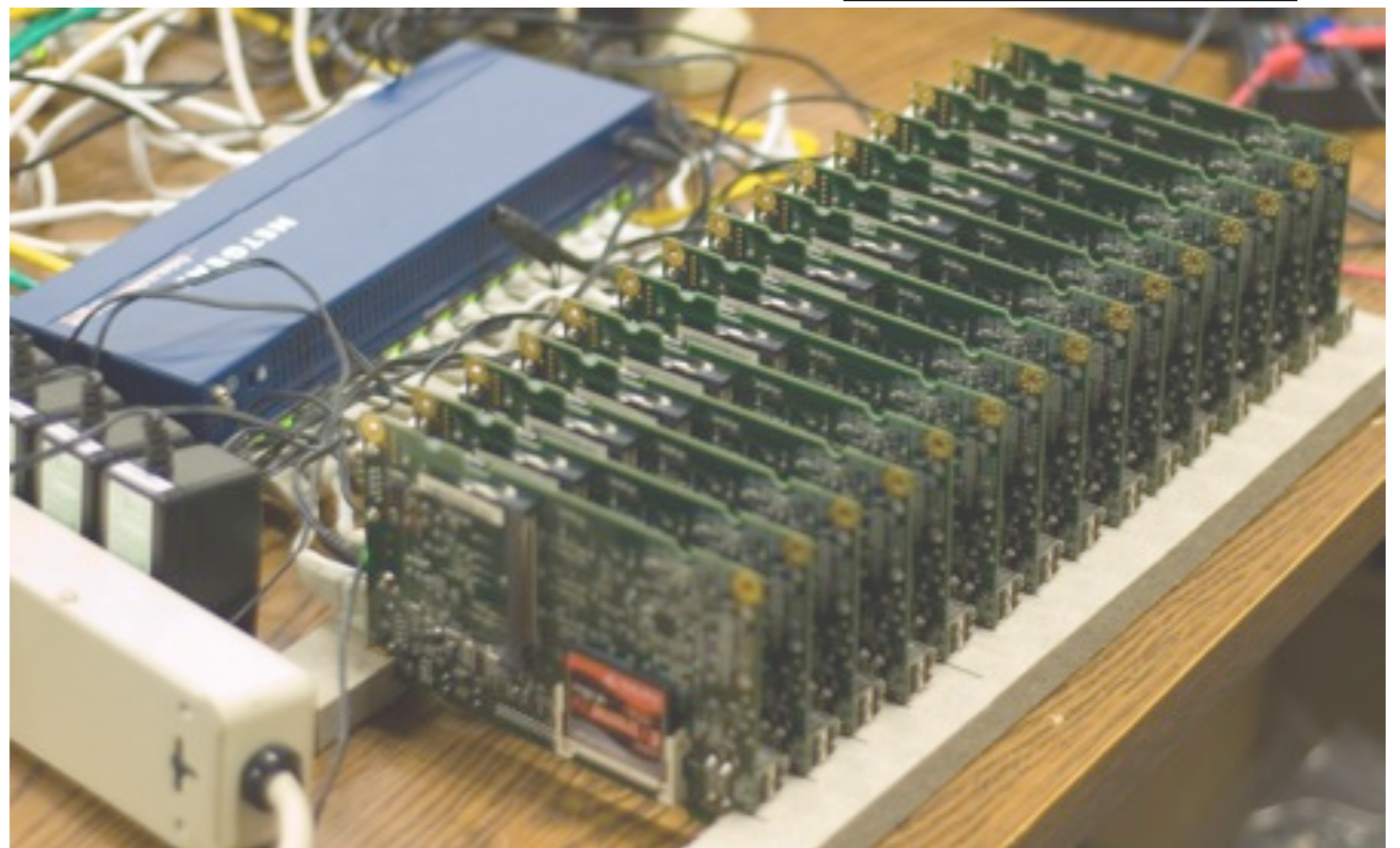
Cuckoo

Fawn-KV

Fawn-DS

Fawn-DS

Fawn-DS





FAWN-DS

FAWN-KV

SILT

Small Cache

Cuckoo

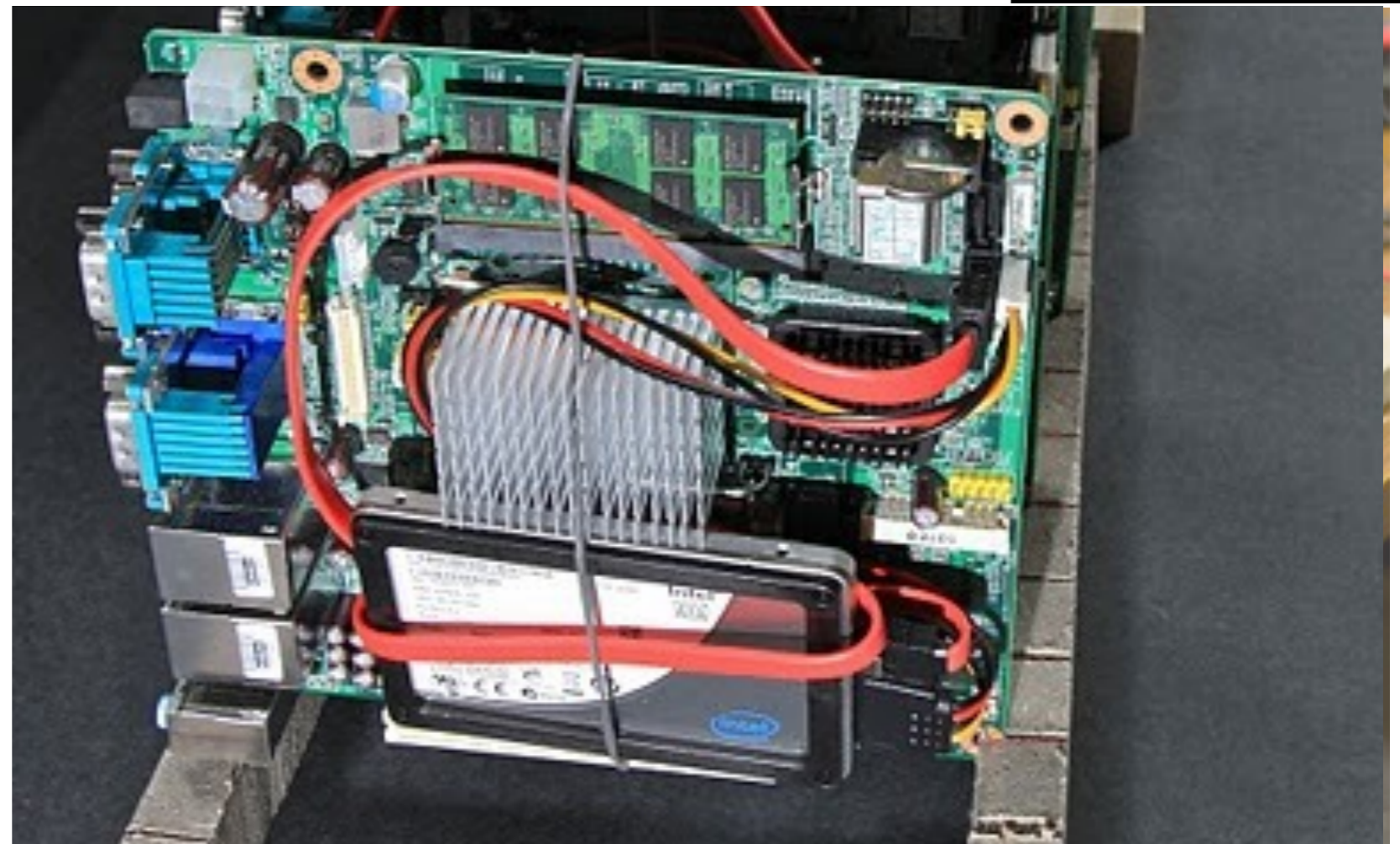
backend store  
hyper-optimized  
for low DRAM  
and large flash

Fawn-KV

SILT

SILT

SILT



Systems begat  
algorithms:

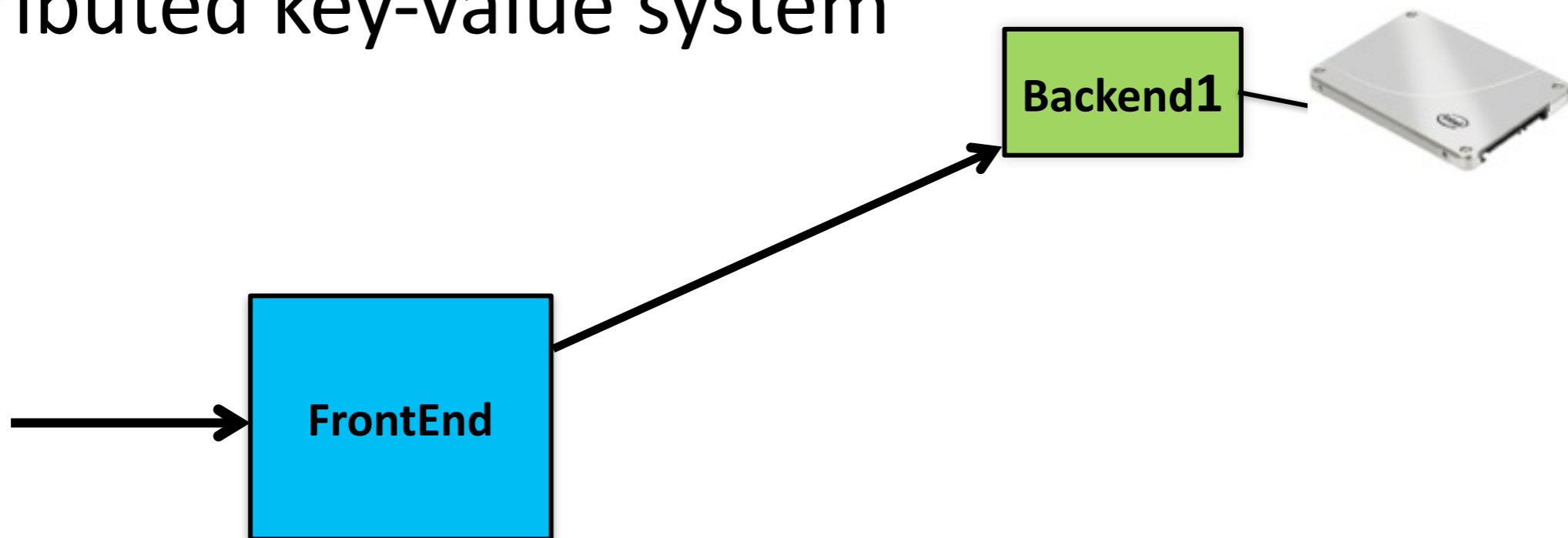
“Practical Batch-Updatable  
External Hashing with Sorting”

H. Lim et al., **ALENEX** 2012

(Recently heard that Bing uses  
several state-of-the-art,  
memory-efficient indexes)

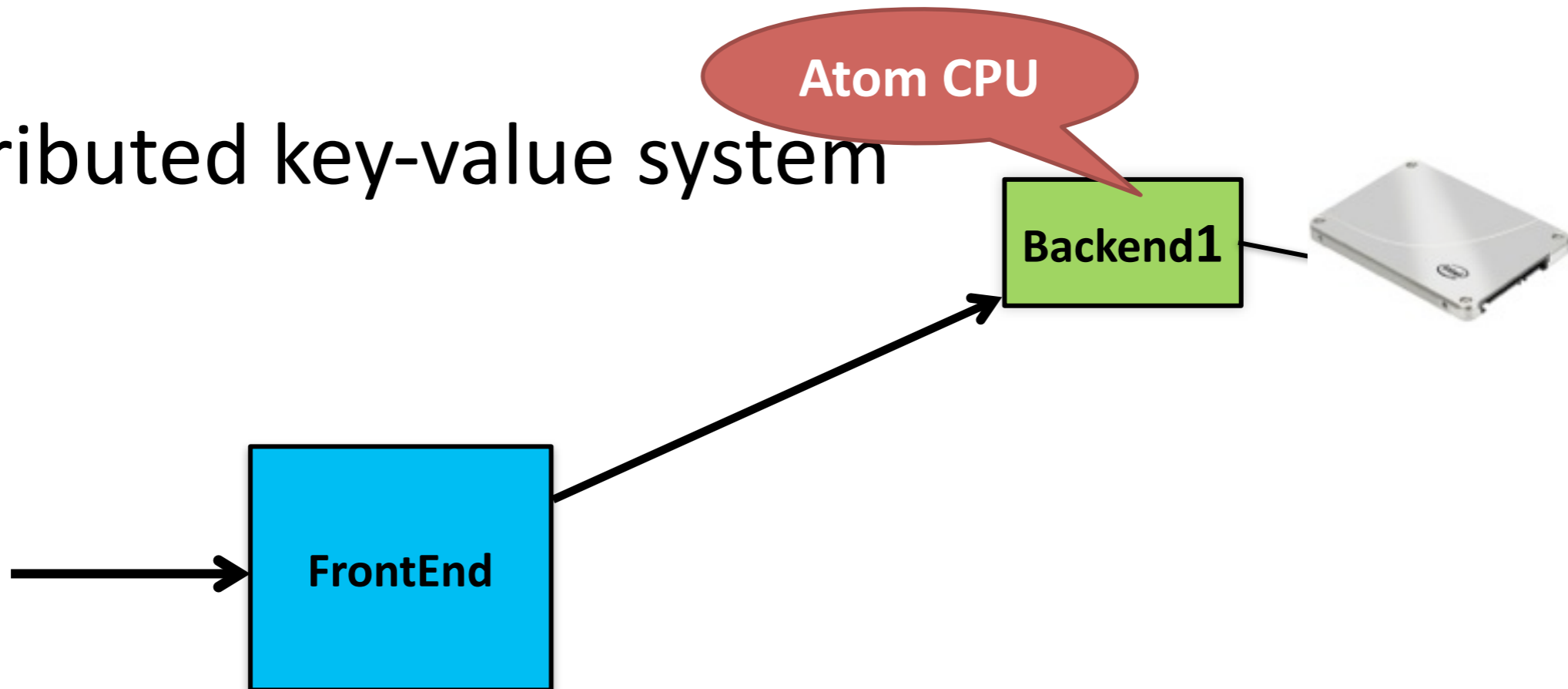
# And now... Load imbalance

- Distributed key-value system



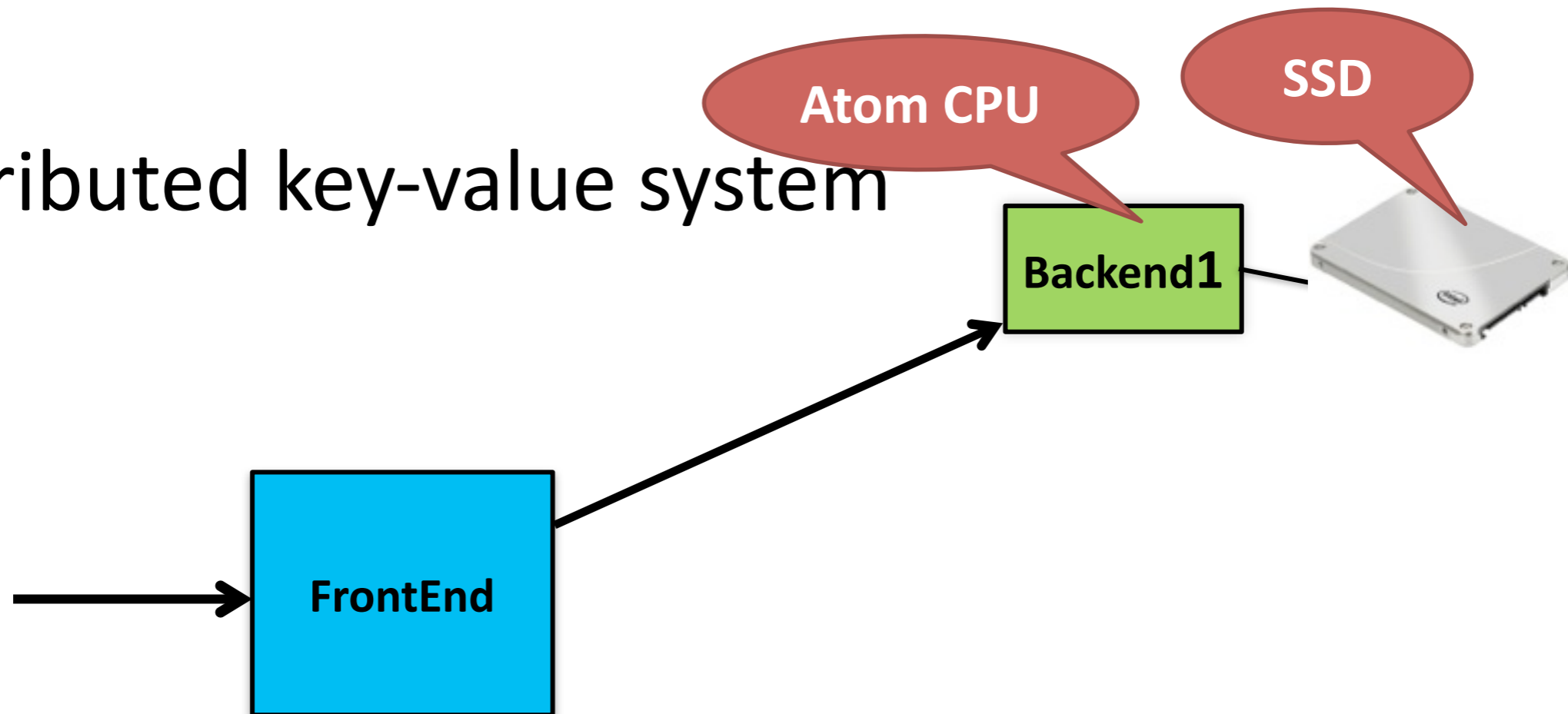
# And now... Load imbalance

- Distributed key-value system



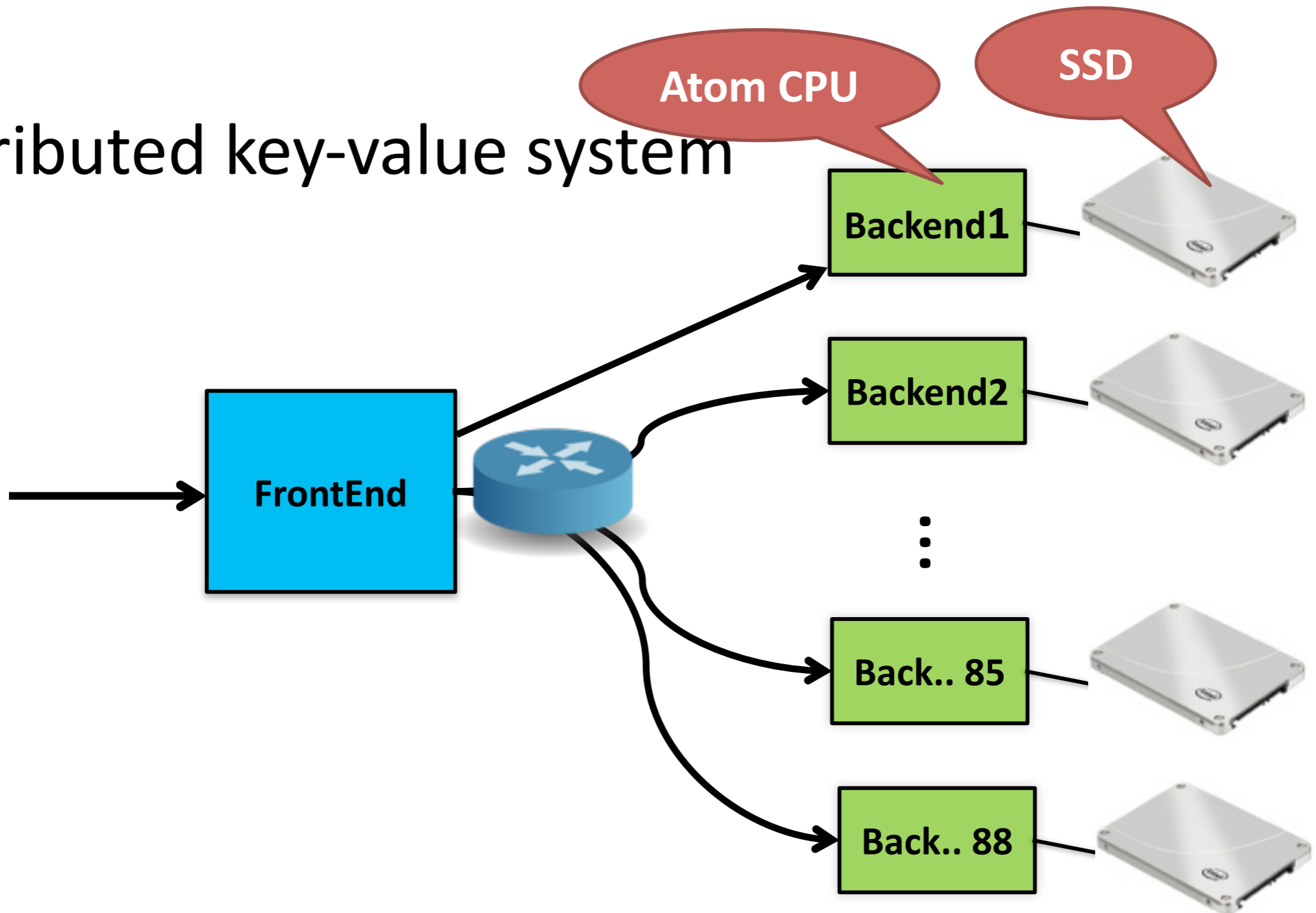
# And now... Load imbalance

- Distributed key-value system



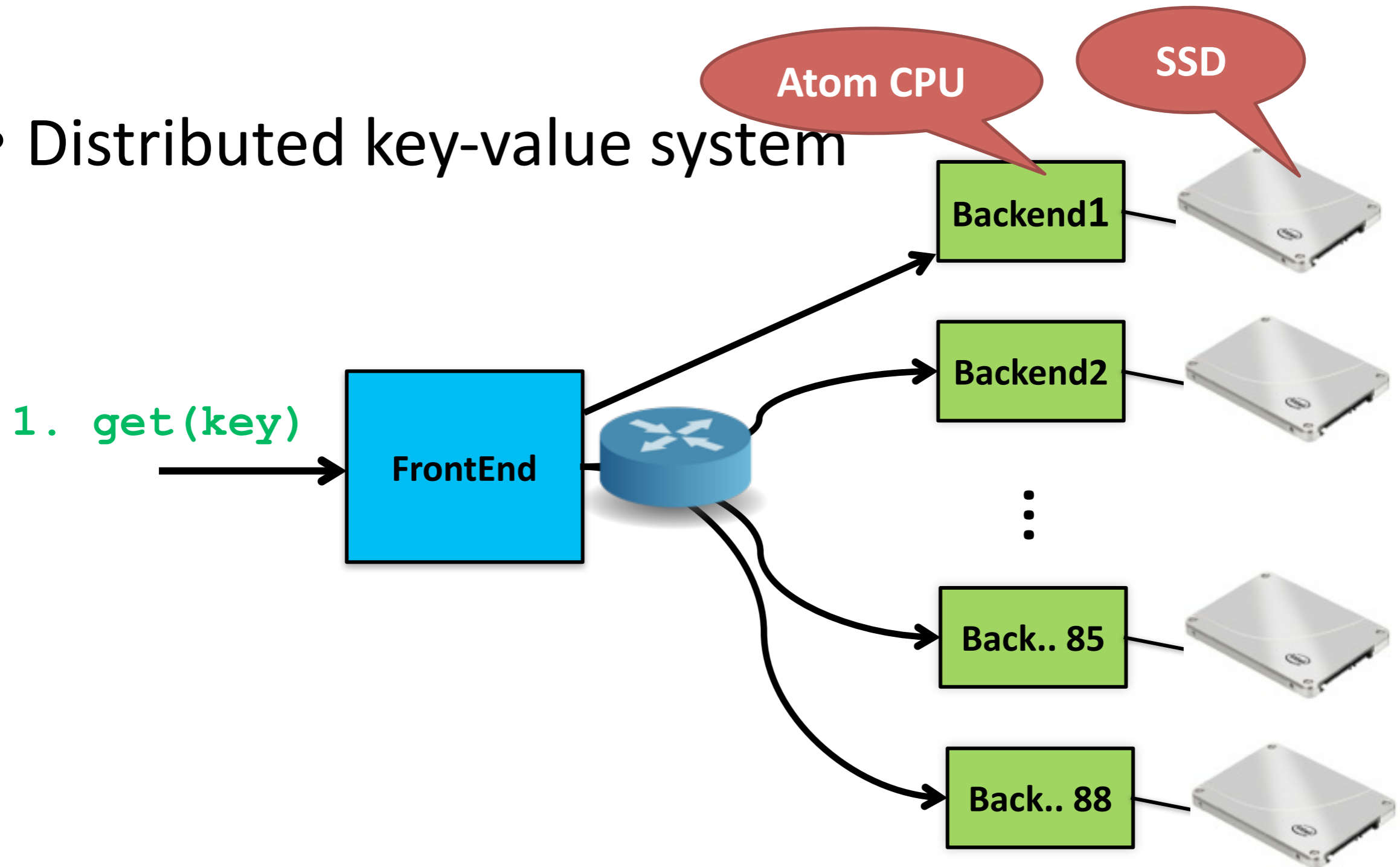
# And now... Load imbalance

- Distributed key-value system



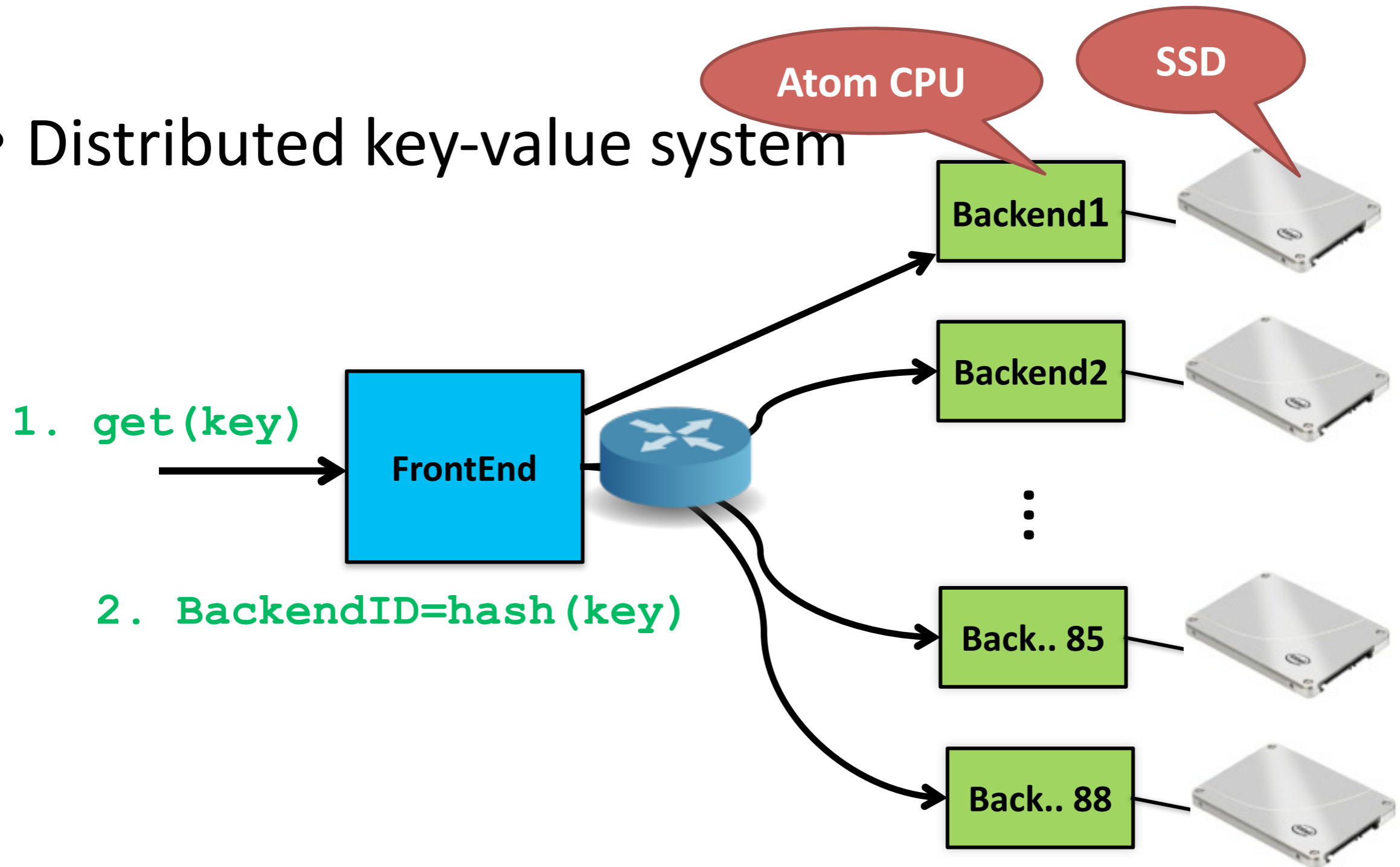
# And now... Load imbalance

- Distributed key-value system



# And now... Load imbalance

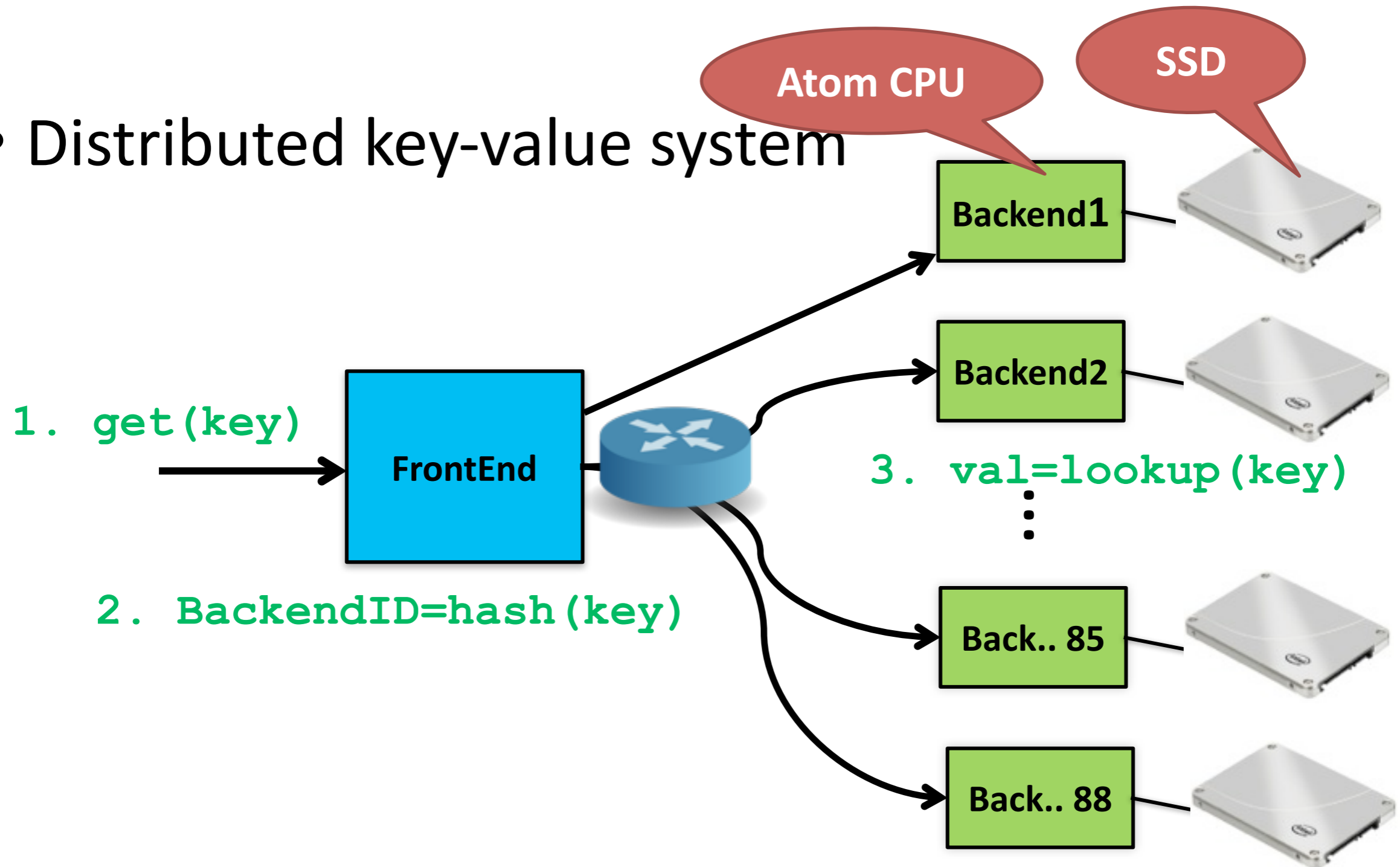
- Distributed key-value system





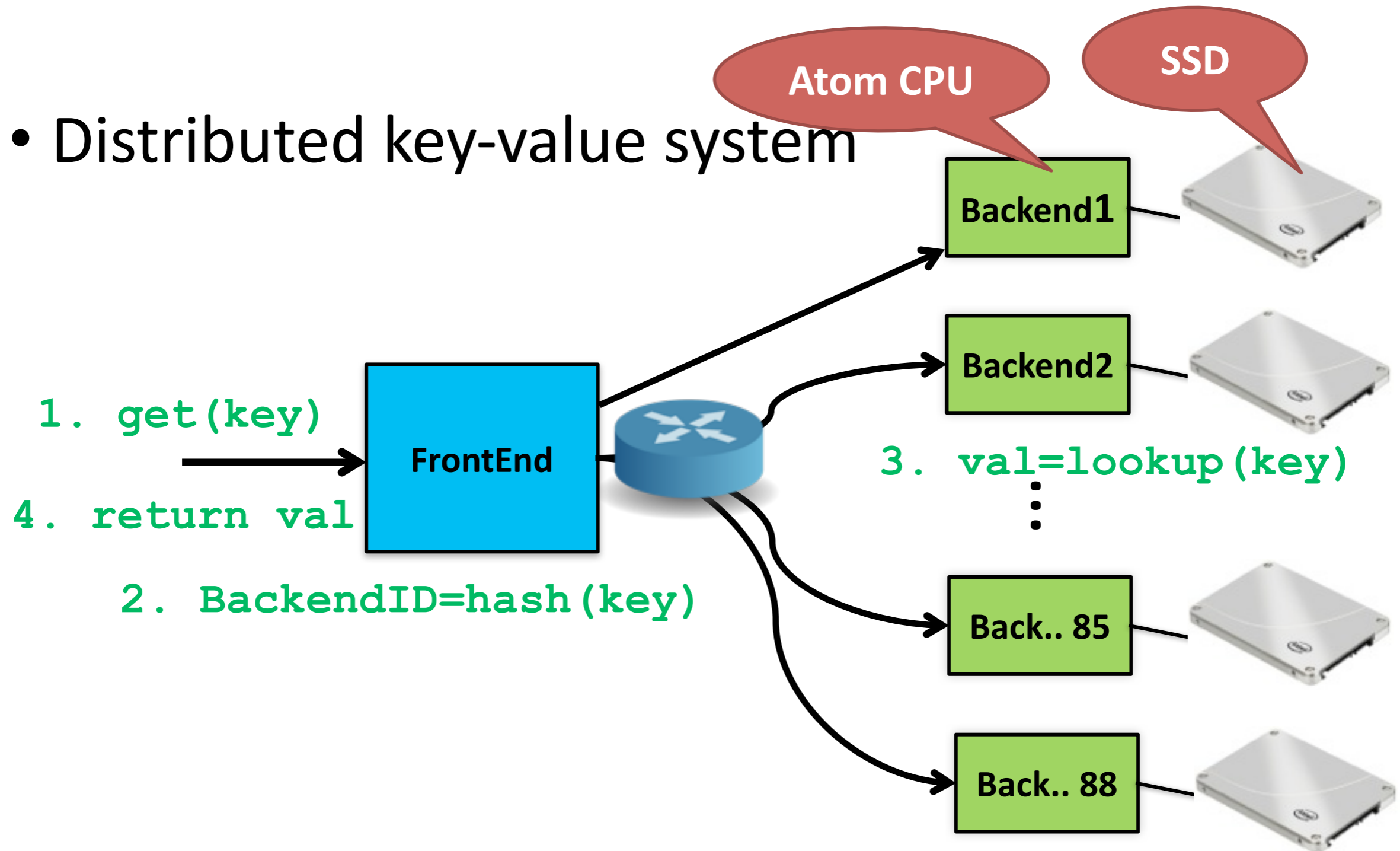
# And now... Load imbalance

- Distributed key-value system



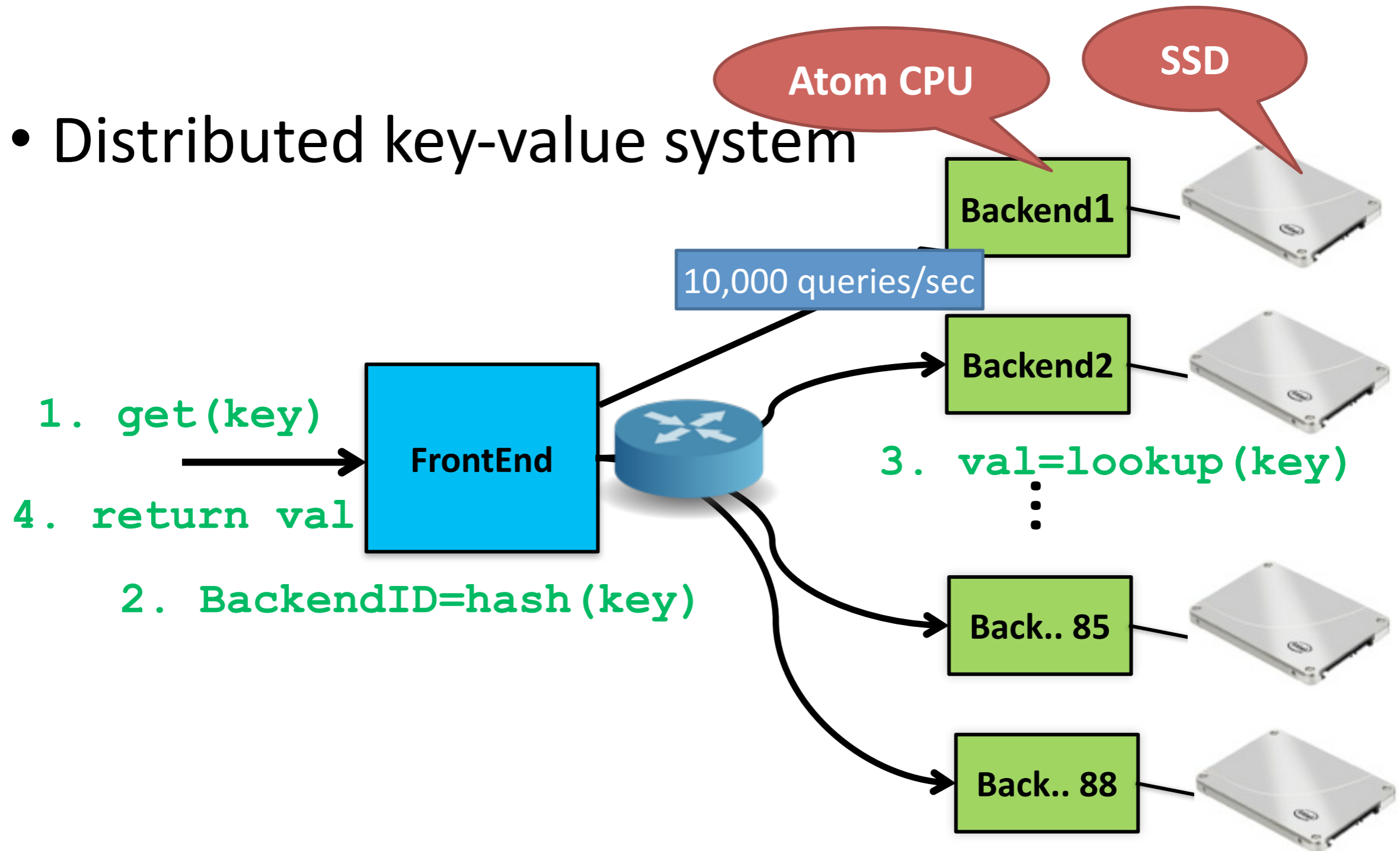
# And now... Load imbalance

- Distributed key-value system



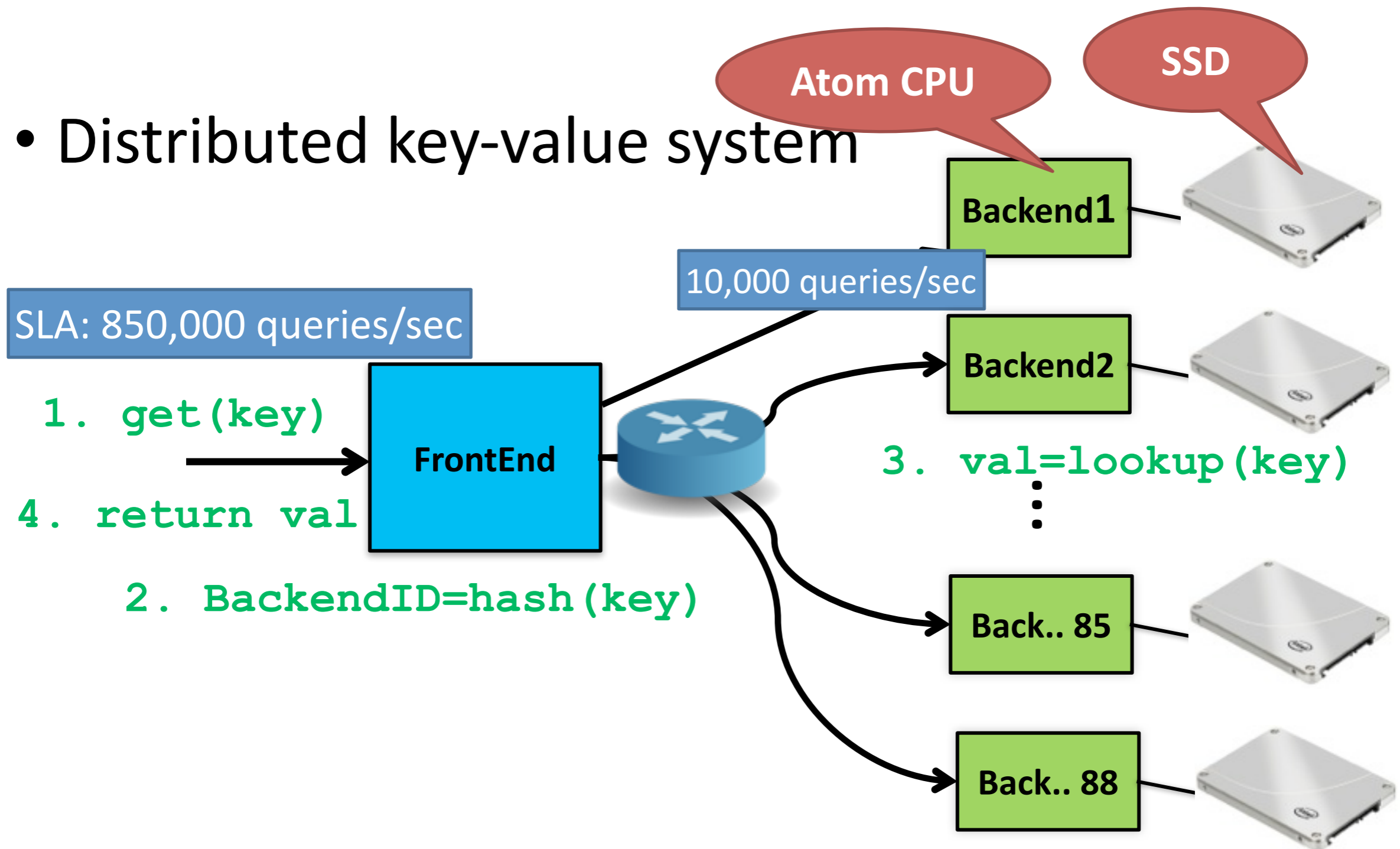
# And now... Load imbalance

- Distributed key-value system

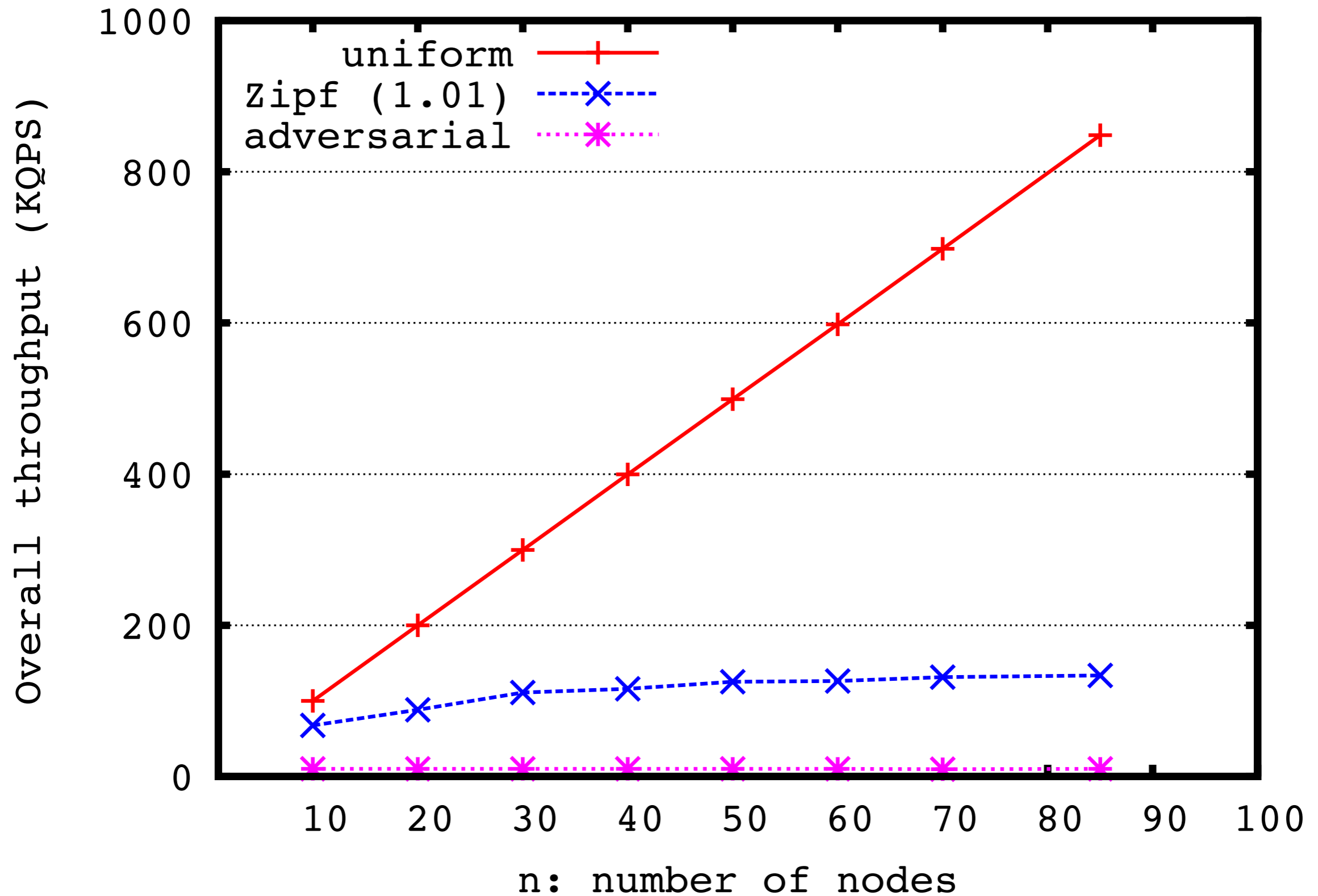


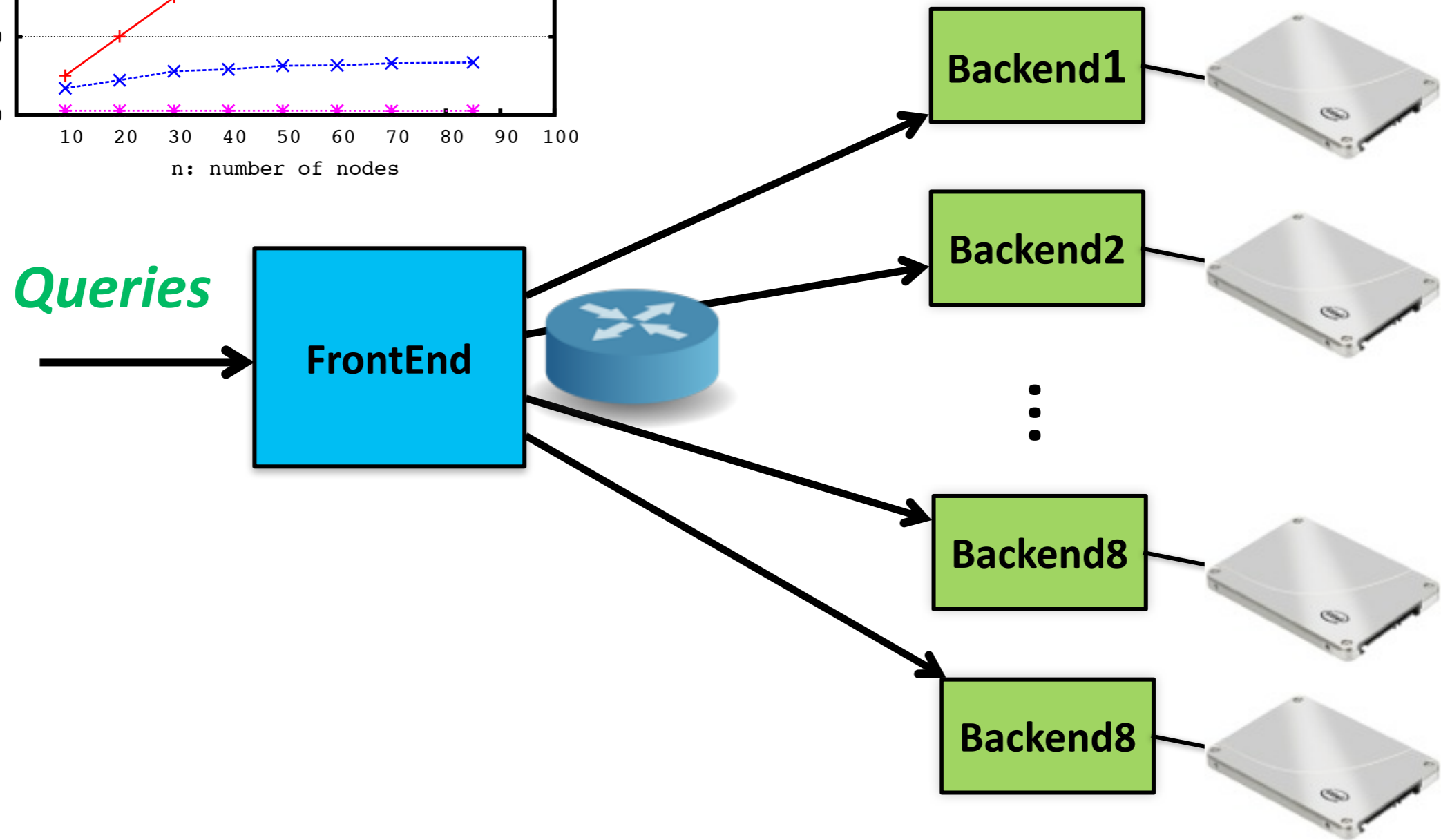
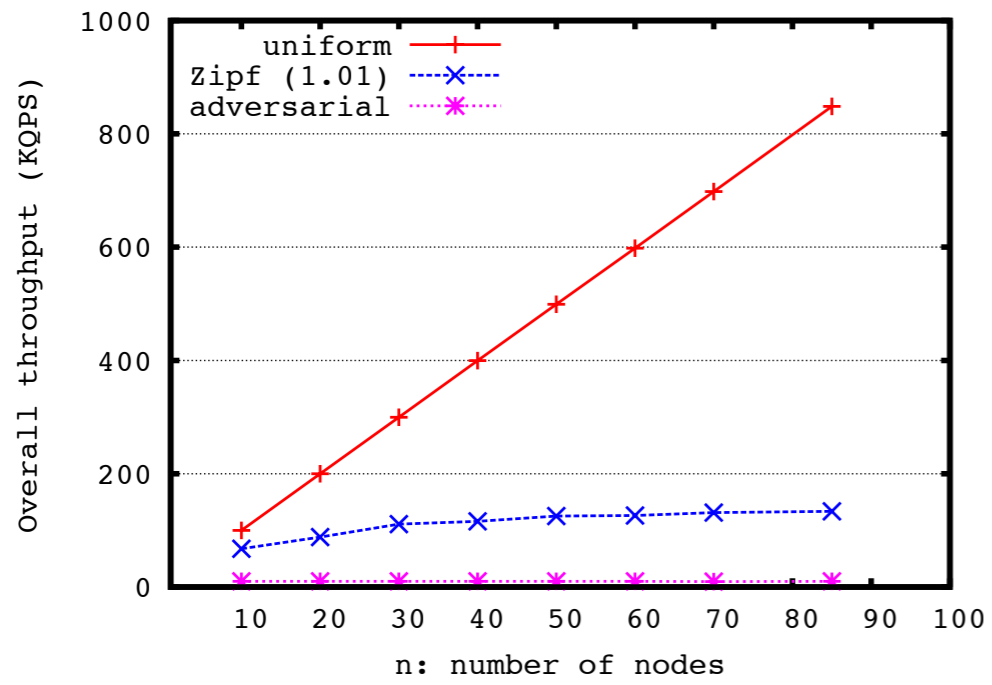
# And now... Load imbalance

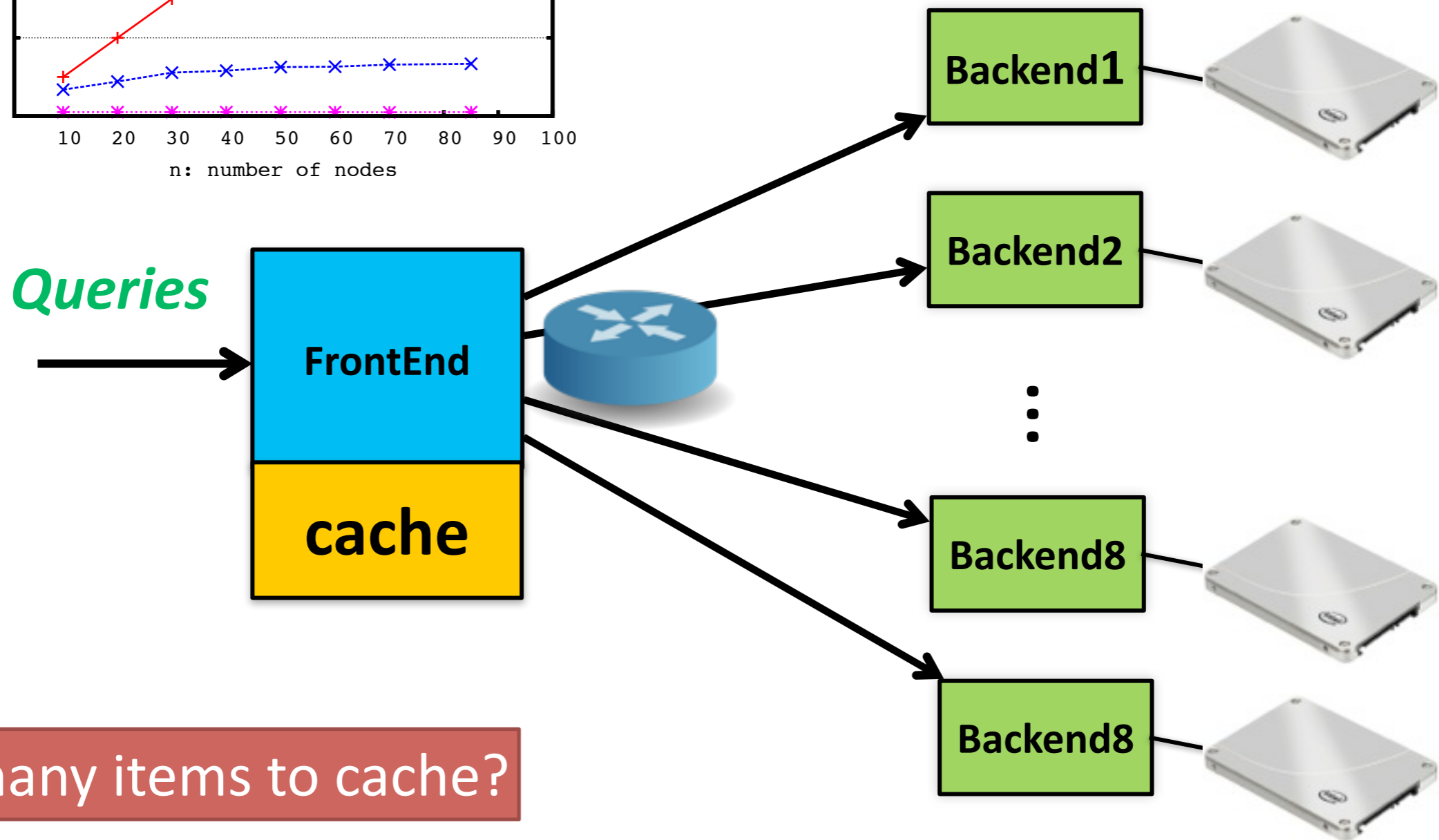
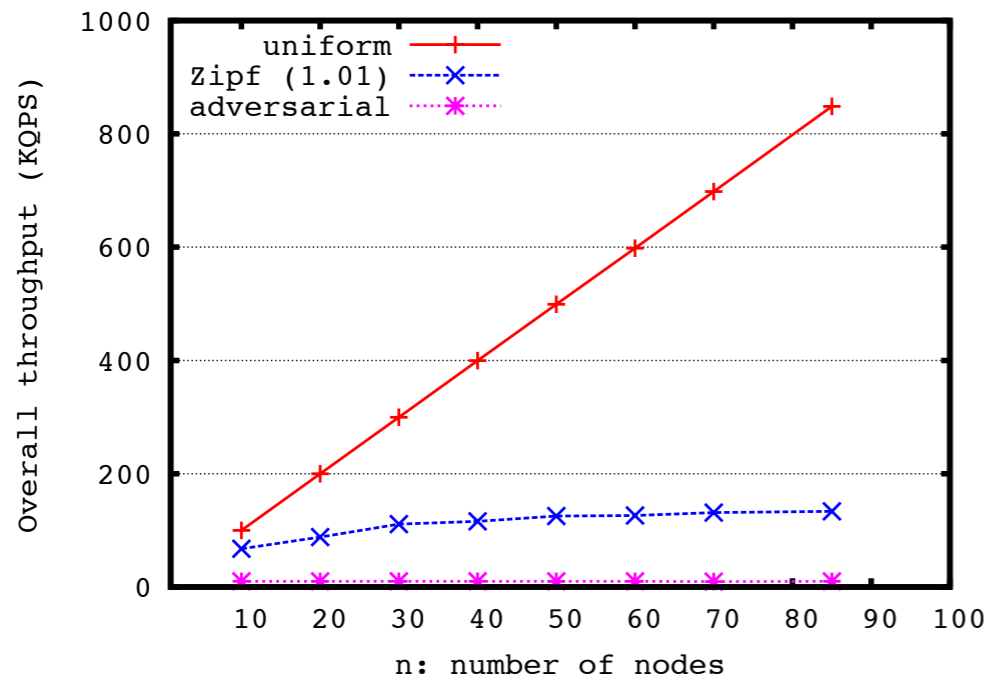
- Distributed key-value system



# Measured tput on FAWN testbed

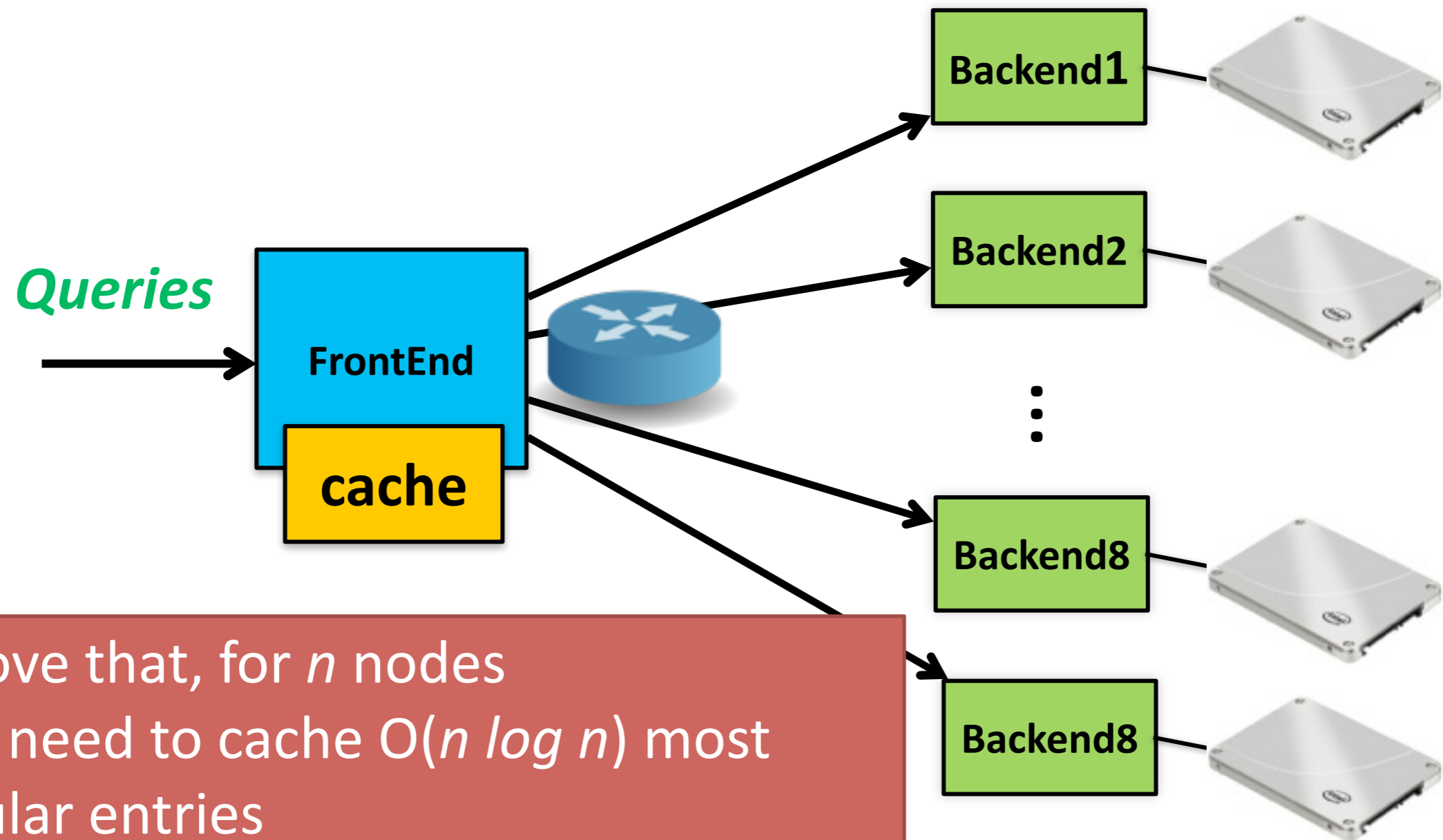






How many items to cache?

# small/fast cache is enough!

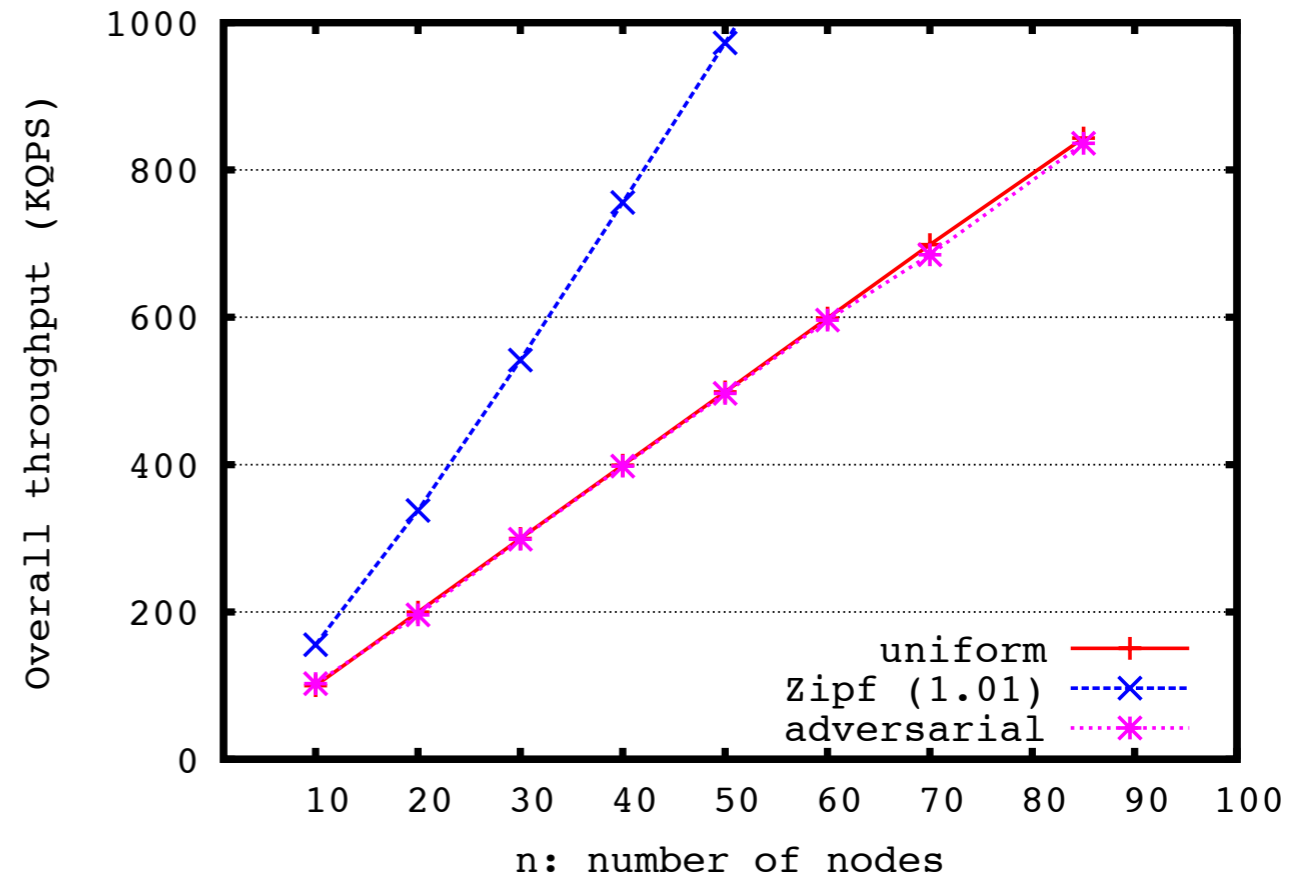
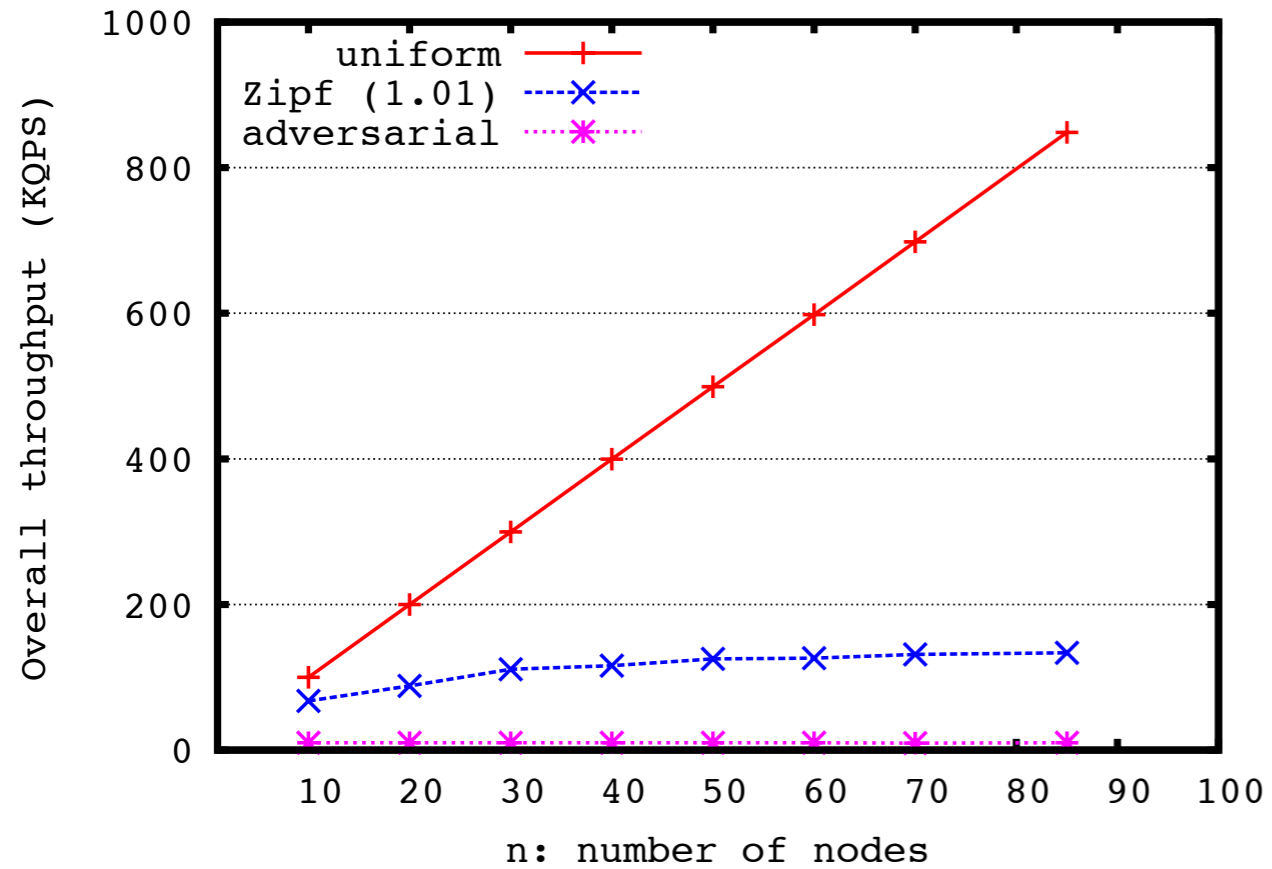


We prove that, for  $n$  nodes

- Only need to cache  $O(n \log n)$  most popular entries
- With 100 backend nodes, need only about 4,000 items in the cache. Tiny!



# Worst case? Now best case



**Thus...**

FAWN-DS

FAWN-KV

SILT

Small Cache

Cuckoo

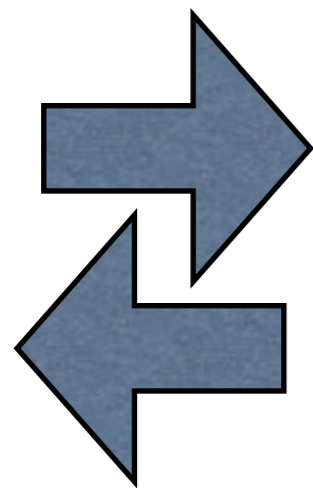
“Wimpy” servers

[FAWN, SOSP 2009]



[SILT, SOSP 2011]

“Brawny” server



Insanely  
Fast Cache

$O(N \log N)$

[“small cache” socc 2011]

Multi-reader  
parallel cuckoo  
hashing

[“MemC3” - NSDI 2013]

Entropy-coded tries

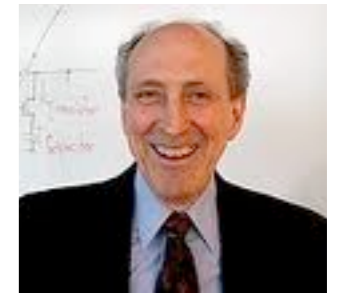
[SOSP + ALENEX]

Partial-key cuckoo hashing

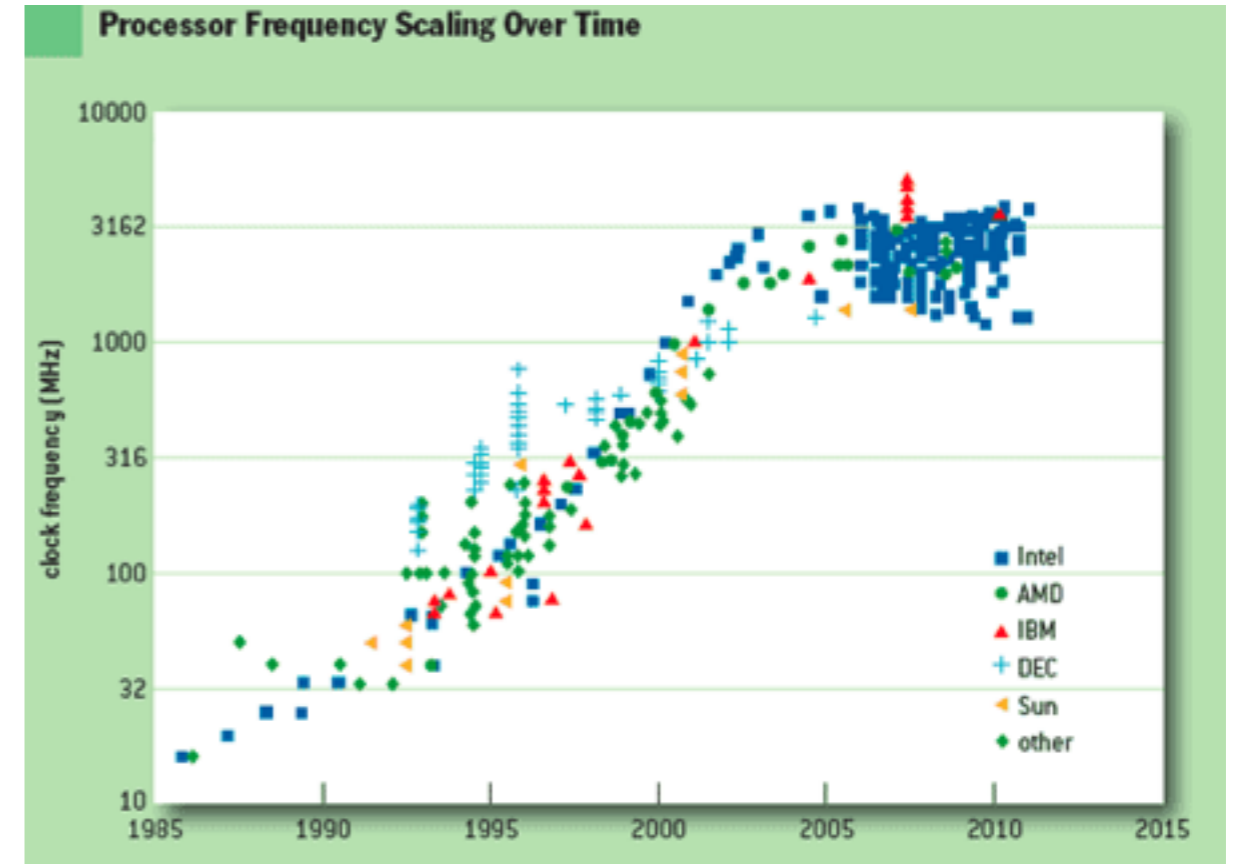
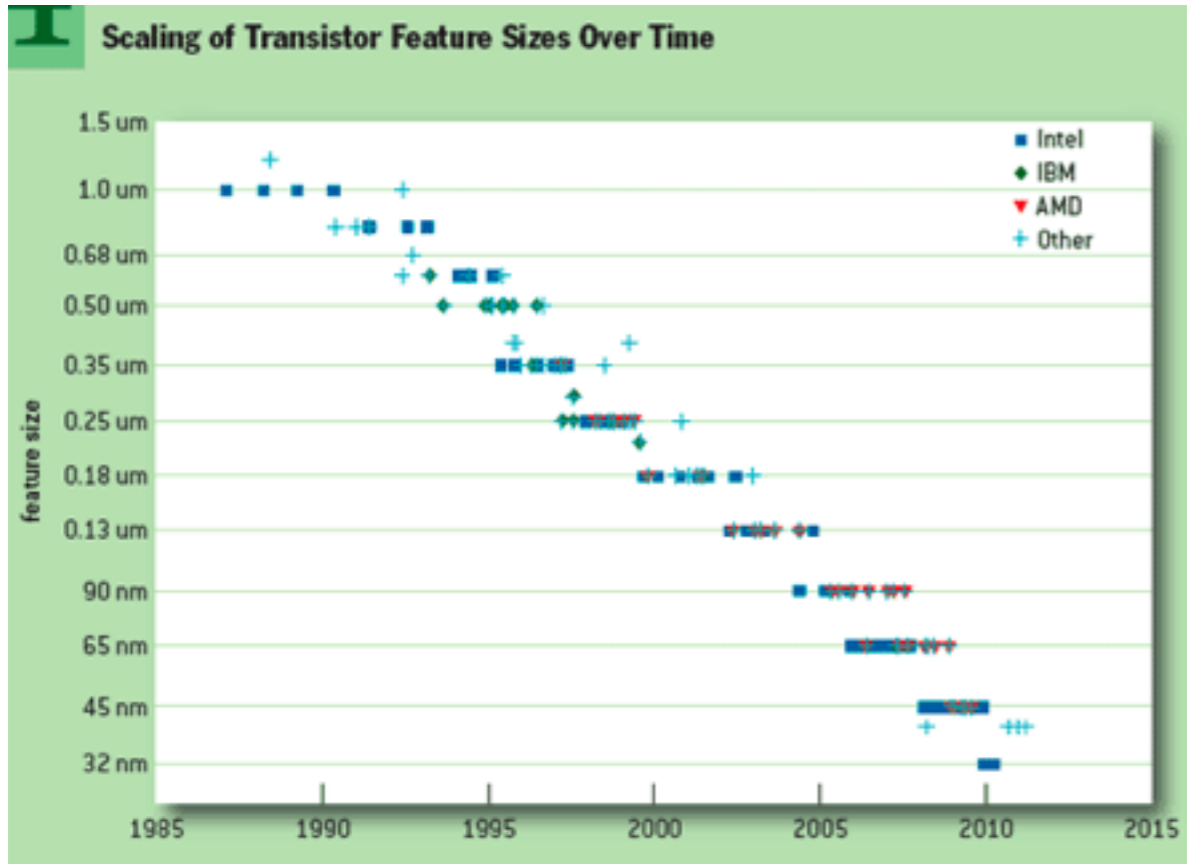
Cuckoo filter



Moore



Dennard



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

*Architectures, algorithms, and programming*