

**facebook**

# Efficiency at Scale

Sanjeev Kumar  
Director of Engineering, Facebook

International Workshop on Rack-scale Computing, April 2014

# Agenda

**1** Overview

**2** Datacenter Architecture

**3** Case Study: Optimizing BLOB Storage system

**4** Questions

# Facebook Stats

- 1.15 billion users [ 6/2013 ]
- ~700 million people use facebook daily
  
- 350+ million photos added per day [ 1/2013 ]
- 240+ billion photos
  
- 4.5 billion likes, posts and comments per day [ 5/2013 ]

 Facebook Datacenter Regions



A large and growing server footprint

# Services Provided by Facebook

The screenshot shows a Facebook news feed with several services highlighted by red boxes:

- Search Bar:** Located at the top left, containing the text "Type to find people, places and things." and a search icon.
- Profile Header:** Shows the user's name "Bill Jia" and profile picture.
- Update Status:** A text input field with the placeholder "What's on your mind?" and options for "Update Status" and "Add Photo / Video".
- Sponsored Post:** A post from "Samsung Mobile USA" with a logo and text: "Victor Li likes Samsung Mobile USA. Dan Lee and 18 other friends also like this." Below the post is a photo of a large conference room.
- Birthdays:** A section listing birthdays with options to "Give him a gift".
- Reminders:** A section with reminders like "Create Event", "2 Diamond Dash invites", and "1 other app request outstanding".
- Ads:** A sponsored ad for "Facebook Custom Audiences" with the text "Ads target your customers" and "Latin Dance @ MPK 19".
- Right Sidebar:** A list of recent activity, including comments and likes, such as "Oscar Quinonez commented on his own status" and "Tim Hughes is listening to Bitch Bad by Lupe Fiasco on Spotify".
- Bottom Bar:** Contains navigation icons for home, search, and a notification bell showing "129 ms".

# Salient Points

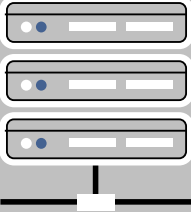
- Efficiency matters
- Complex Software Stack
  - 1000+ specialized services to run
    - A few large services + Long tail
- Custom hardware: cost of designing, validating, fixing
- Number of machines a service needs can change quickly

Many sources of complexity  
Simplify as much as possible

# Agenda

- 1** Overview
- 2** Datacenter Architecture
- 3** Case Study: Optimizing BLOB Storage system
- 4** Questions

## Front-End Cluster



**Web**  
250 racks

---

**Cache (~144TB)**



**Ads**  
30 racks



**Multifeed**  
9 racks



**Other small services**

## Service Cluster

Search   Photos   Msg   Others

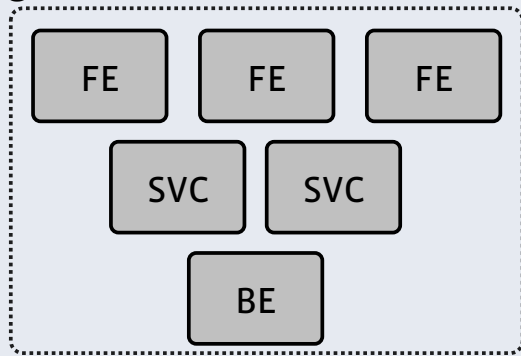
## Back-End Cluster

UDB   ADS-DB   Tao Leader

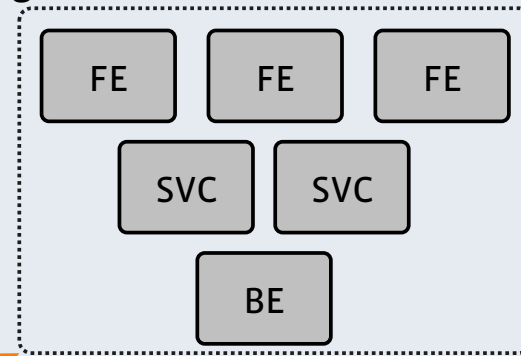


# Infrastructure Redundancy

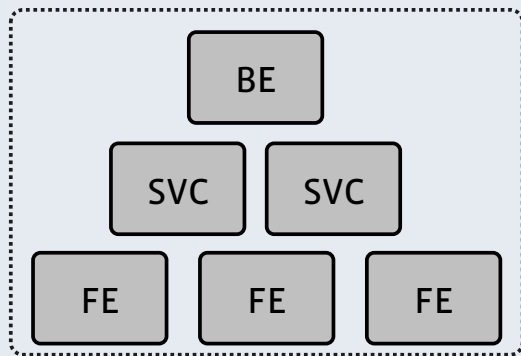
Regional Datacenter 1



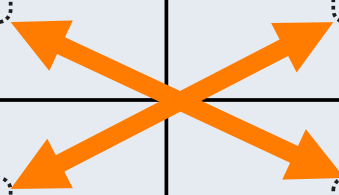
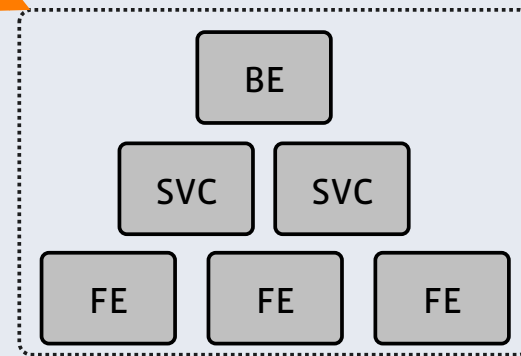
Regional Datacenter 2



Regional Datacenter 3



Regional Datacenter 4



Standard Systems	Type I	Type II	Type III	Type IV	Type V	Type VI
CPU	High 2 x EN2670	Low 1 x 6128HE (AMD)	Medium 2 x X5650	Medium 2 x X5650	Low 1 x L5630	High 2 x EN2660
Memory	Low 16GB	High 144GB	High 144GB	Medium 48GB	Low 18GB	High 144GB
Disk	Low 250GB	Low 250GB	High IOP 6 x 600GB SAS +2x1.3TB Flash	High 12 x 3TB SATA	High 12 x 3TB SATA	Medium 1TB SATA
Services	Web, Chat, Ads	Memcache, Ads	Database	Hadoop	Photos, Video	Multifeed, Search

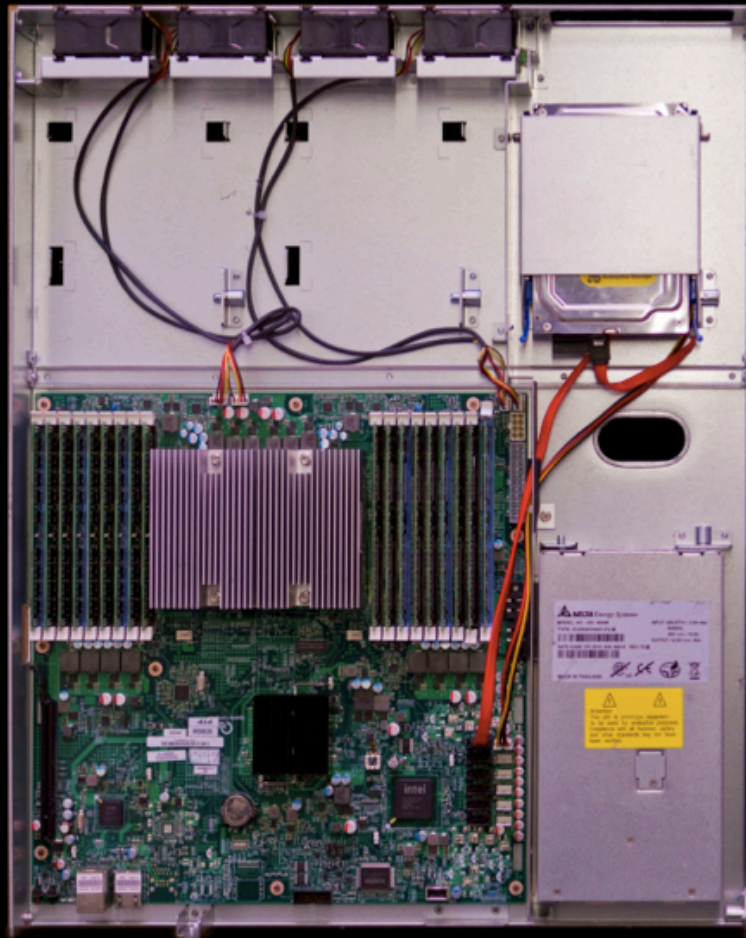
Standard Systems	Type I	Type II	Type III	Type IV	Type V	Type VI
CPU	High	Low	Medium	Medium	Low	High
Memory	Low	High	High	Medium	Low	High
Disk	Low	Low	High IOPs	High	High	Medium
Services	Web, Chat, Ads	Memcache, Ads	Database	Hadoop	Photos, Video	Multifeed, Search

# Server Generations

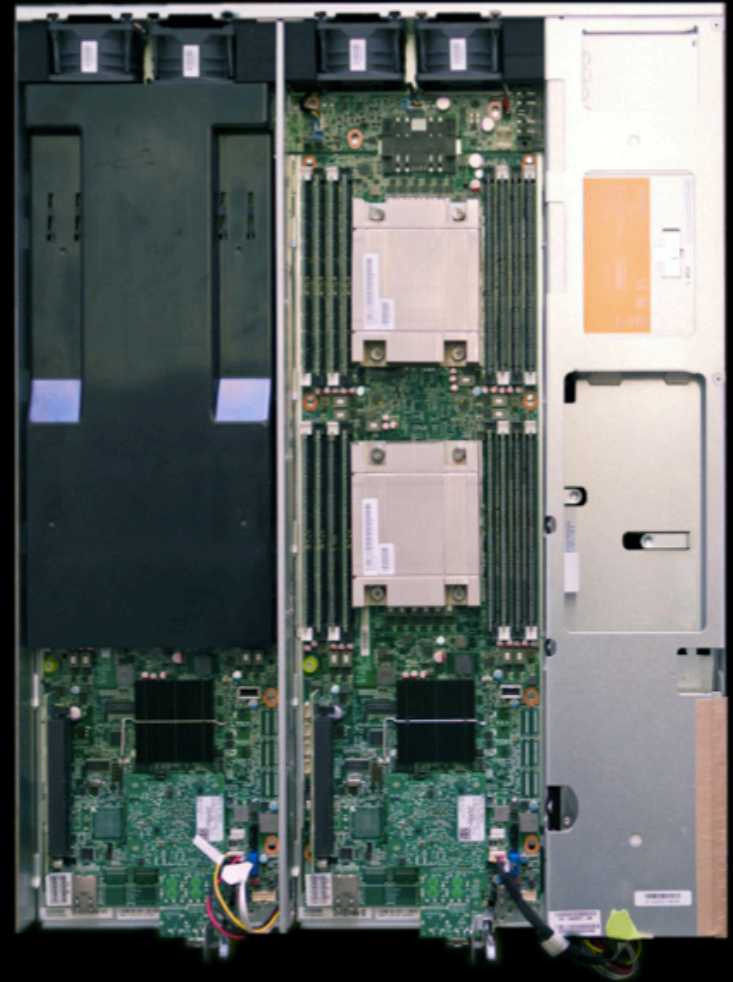
Web Servers	2008	2009	2010	2011	2012
Rack Composition	L5420 (SC)	L5520 (NHM)	L5639 (WSM)	X5650 (XWSM)	EN2670 (SND)
Cores / Speed	8 real cores 2.50 GHz	16 logical CPUs (HT) 2.27 GHz	24 logical CPUs (HT) 2.13 GHz	24 logical CPUs (HT) 2.67 GHz	32 Logical CPUs (HT) 2.33GHz
RCUs	0.6	1	1.4	1.75	2.41



# Web v1



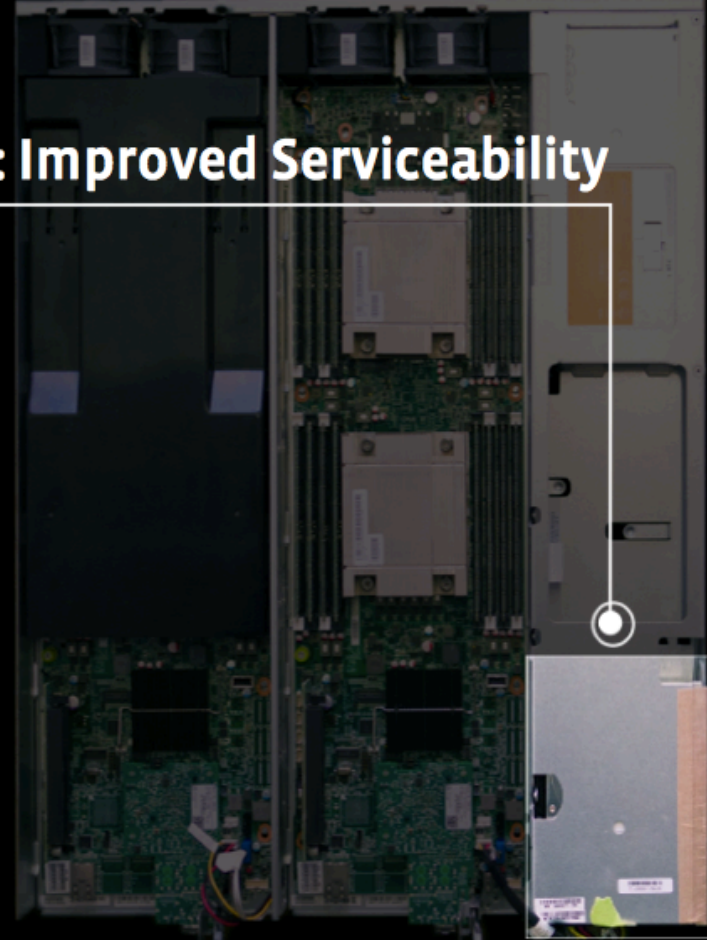
# Web v2



# Web v1



# Web v2



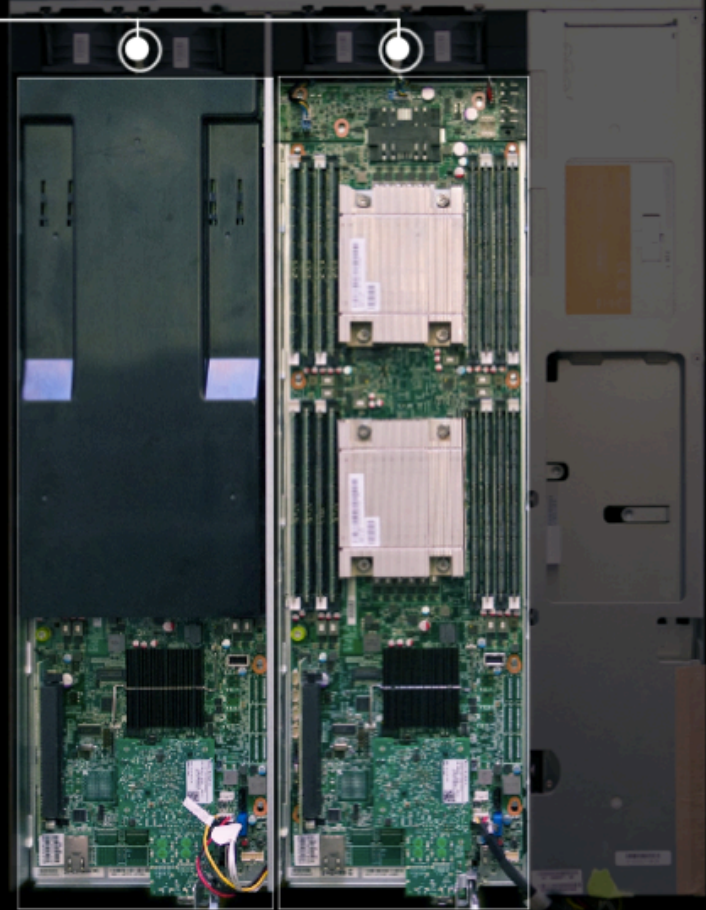
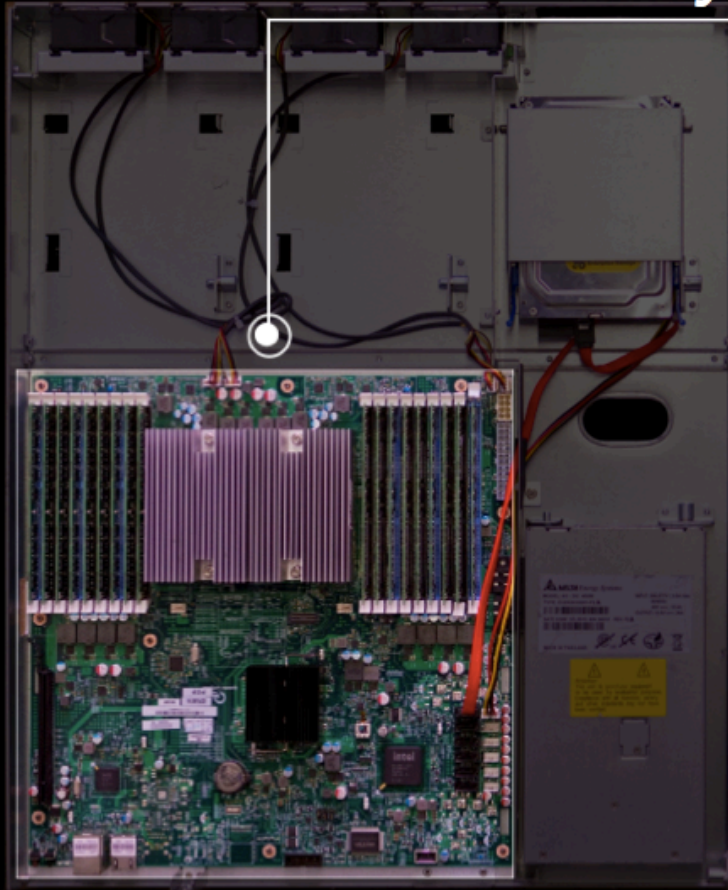
Drive: Improved Serviceability



# Web v1

# Web v2

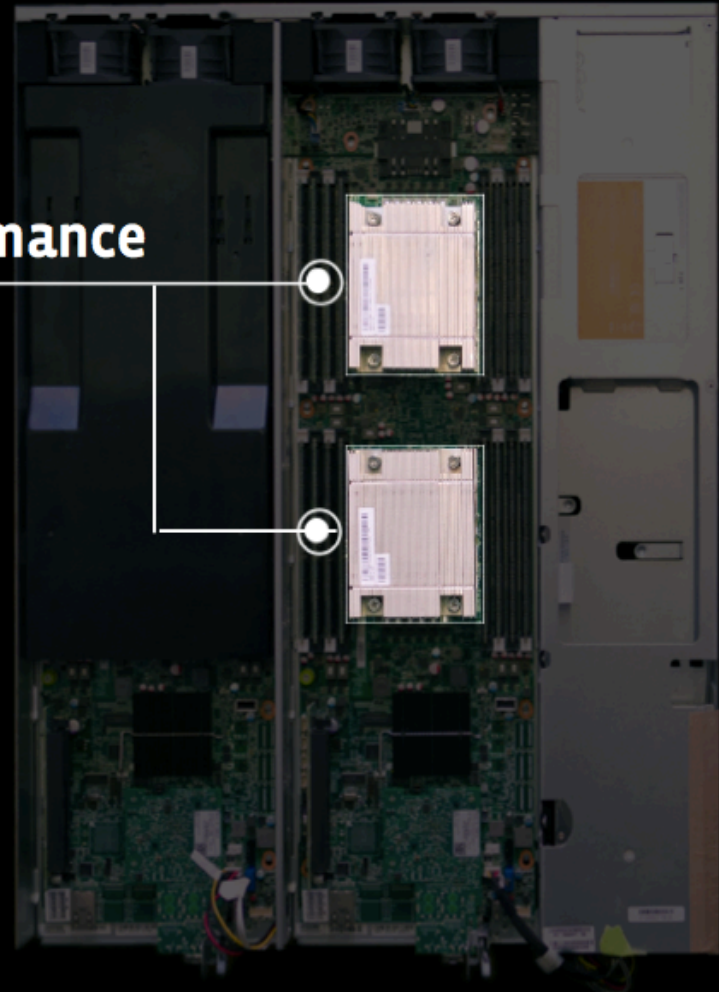
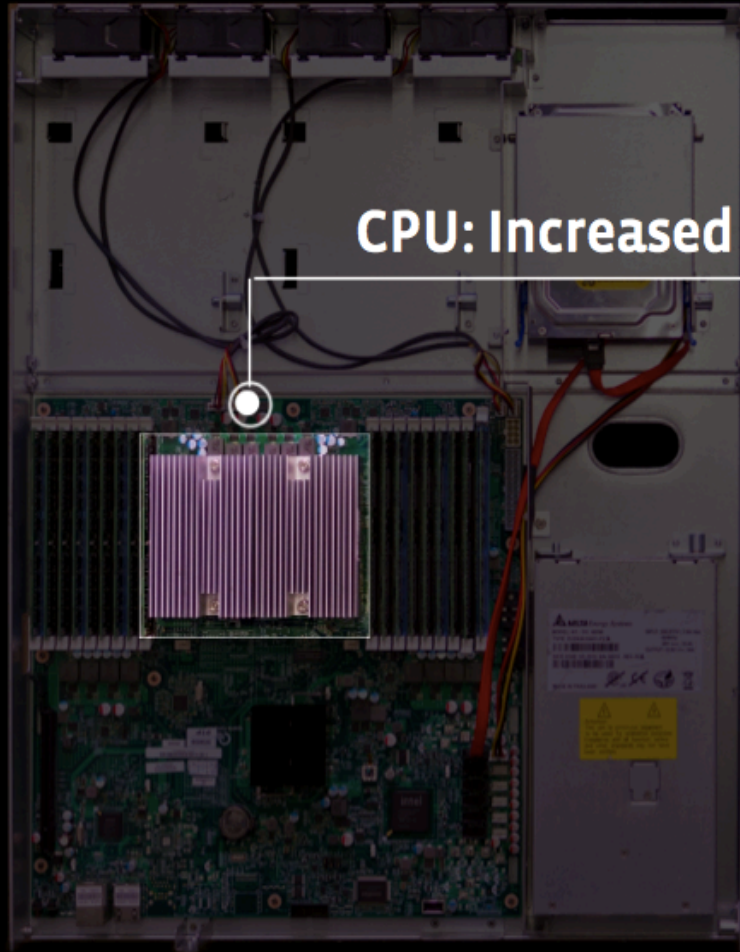
Density: 2 Motherboards





# Web v1

CPU: Increased Performance



# Agenda

- 1** Overview
- 2** Datacenter Architecture
- 3** Case Study: Optimizing BLOB Storage system
- 4** Questions

# Storage Systems

	Total Size	Storage Technology	Bottlenecks
Social Graph	Single-digit petabytes	MySQL & Alternatives	Random read IOPS
Messages & Time Series Data	10s of petabytes	HBase and HDFS	Write IOPS & Storage capacity
Photos/Videos/BLOBs	100s of petabytes	Haystack	Storage capacity
Data Warehouse	100s of petabytes	Hive, HDFS, and Hadoop	Storage capacity
Cold Storage	Exabytes**	Custom	Storage Capacity

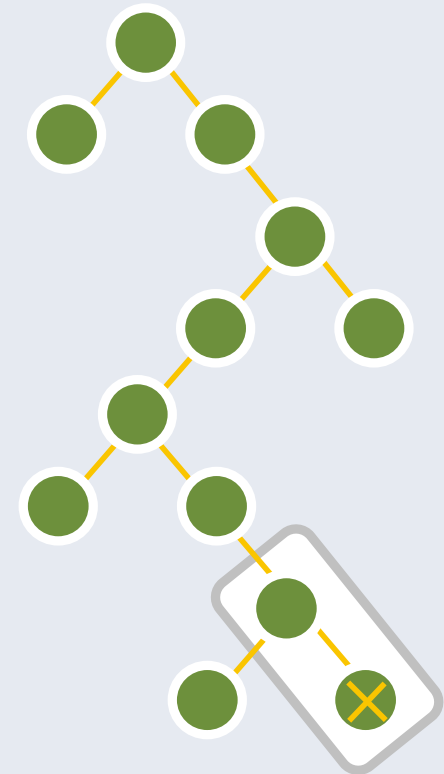
# BLOB Storage

- Storage for Photos, Videos, Attachments, etc.
- Evolved over many generations
  - Constrained resource shifts and needs to be optimized for
    - Generation 1: Time to Market
    - Generation 2 & 3: Optimize the I/O request rate (Cost)
    - Generation 4: Optimize for Storage Efficiency (Cost)

# Generation 1: Commercial Filers

- New Photos Product
- First build it the easy way
  - Commercial Storage Tier + HTTP server
  - Each Photo is stored as a separate file
- Quickly up and running
  - Reliably Store and Serve Photos
- **But:** Inefficient
  - Limited by IO rate and not storage density
  - Average 10 IOs to serve each photo
  - Wasted IO to traverse the directory structure

NFS Storage



# Effective but inefficient

- Disks are slow: 100 reads per disk per second
  - 1 photo read → 10 disk reads
  - Each disk can serve 10 photos per second

A copy of each photo



A copy of each photo



A copy of each photo



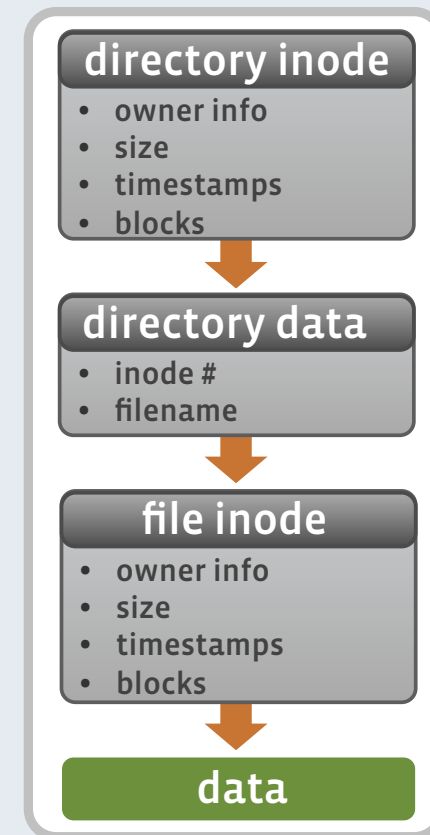
A copy of each photo



# Generation 2: Gen 1 Optimized

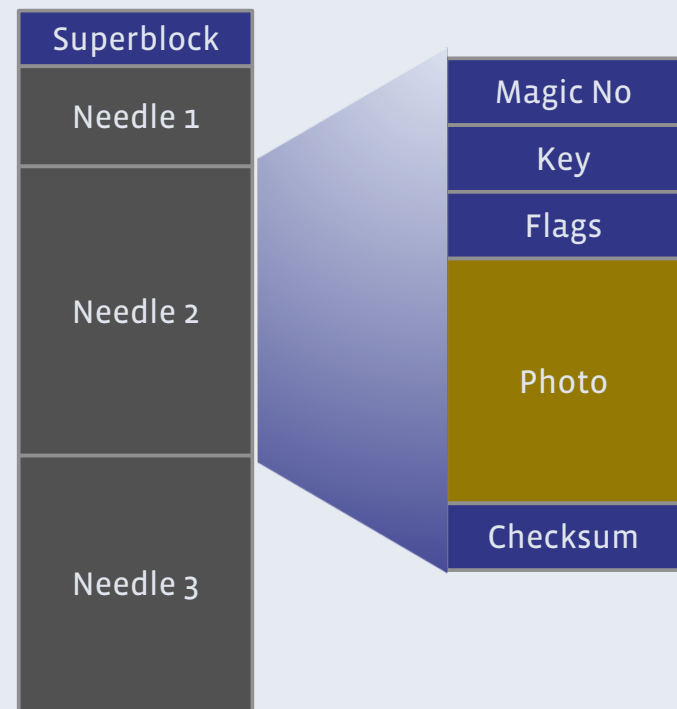
- Optimization Example:
  - Cache NFS handles to reduce wasted IO operations
- Reduce the number of IO operations per photo by 3X
- **But:**
  - **Still expensive:** High end storage boxes
  - **Still inefficient:** Still IO bound and wasting IOs

## NFS Storage Optimized



# Generation 3: Haystack [OSDI'10]

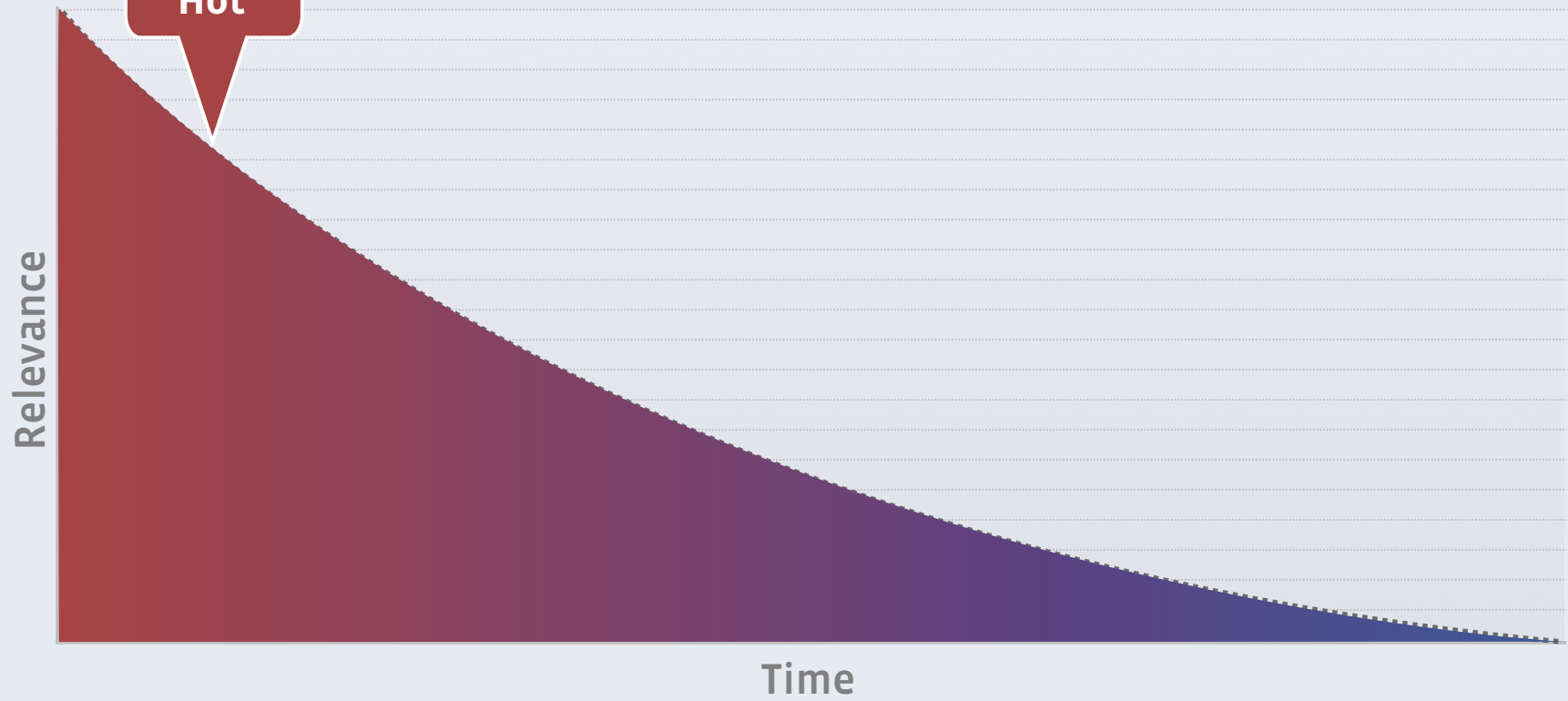
- Custom Solution
  - Commodity Storage Hardware
  - Optimized for 1 IO operation per request
    - **File system on top of a file system**
    - Compact Index in memory
    - Metadata and data laid out contiguously
- Efficient from IO perspective
- **But:**
  - Problem has changed now



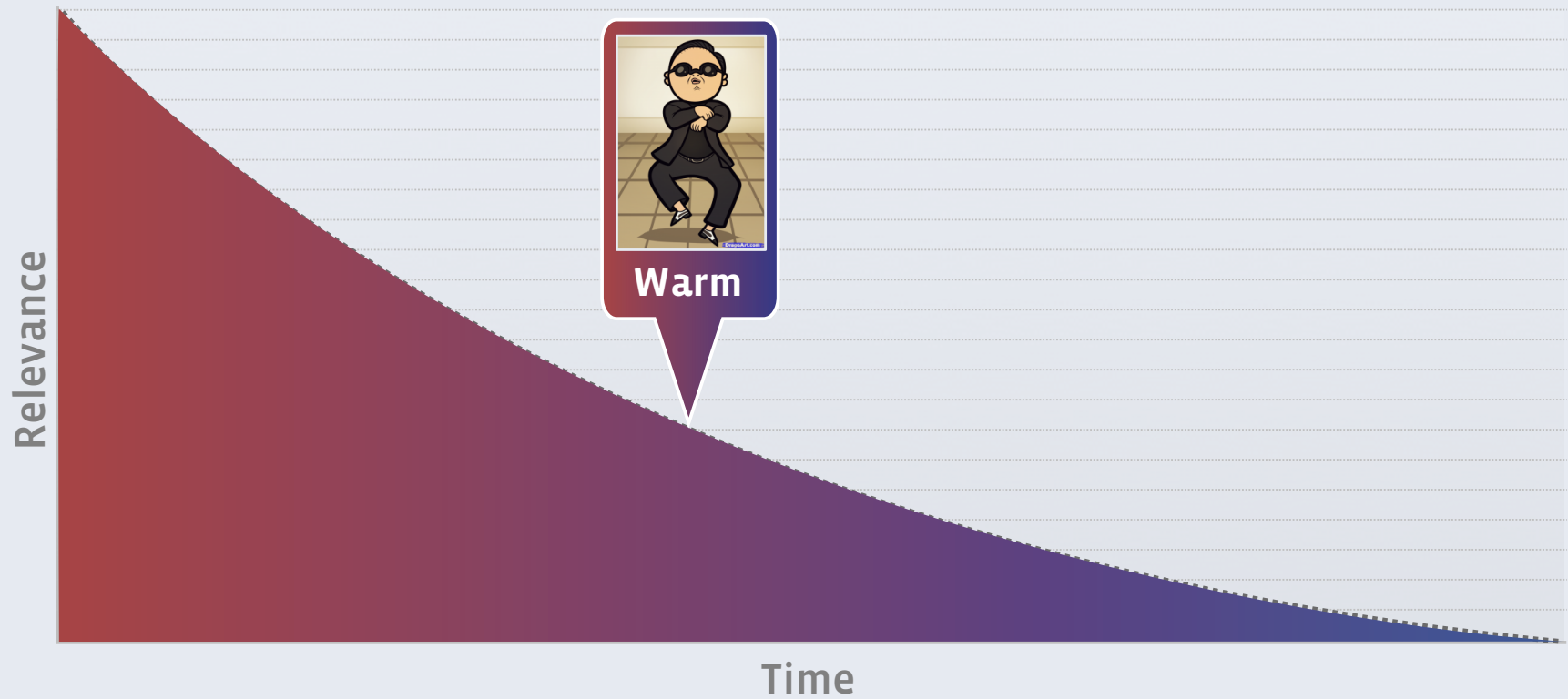
Single Disk IO to read/write a photo



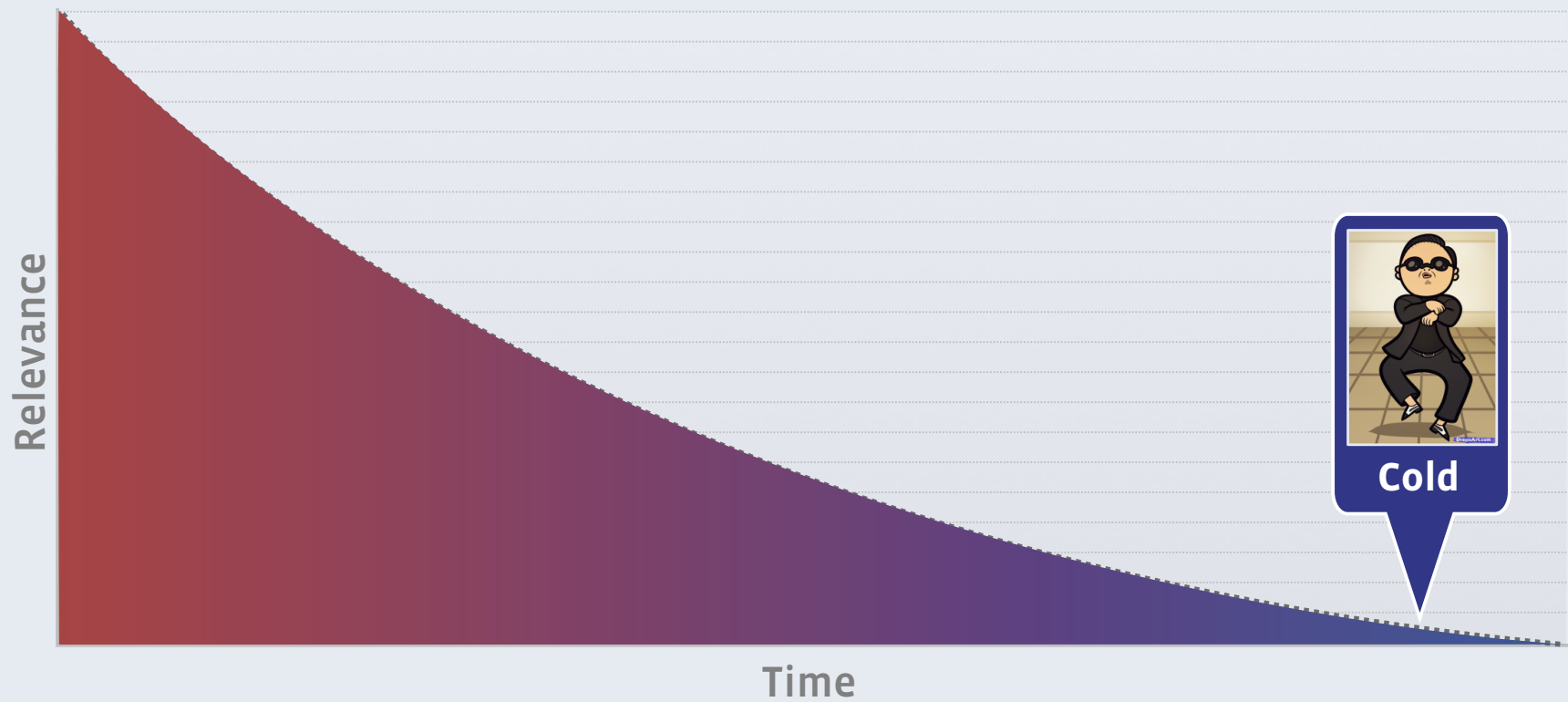
# Photo lifecycle



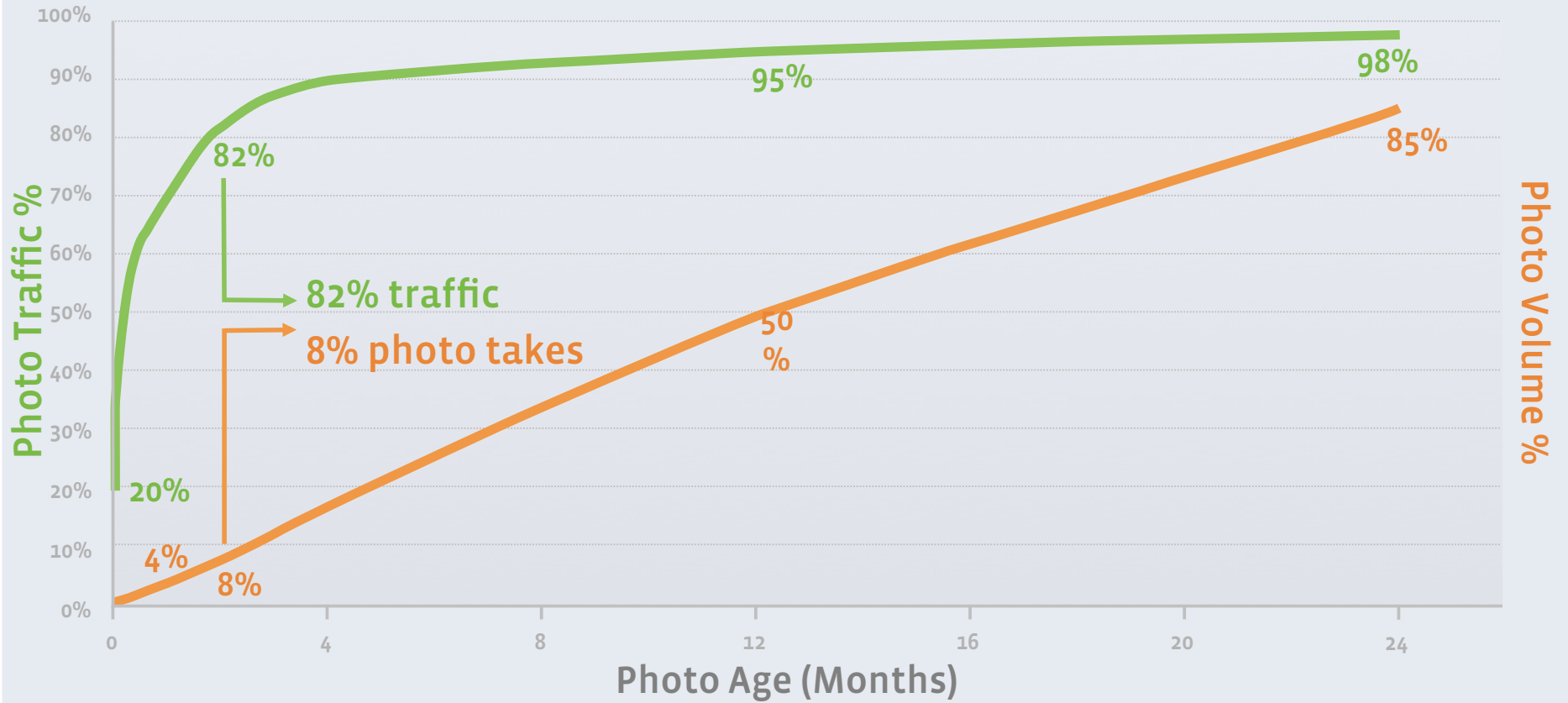
# Photo lifecycle



# Photo lifecycle

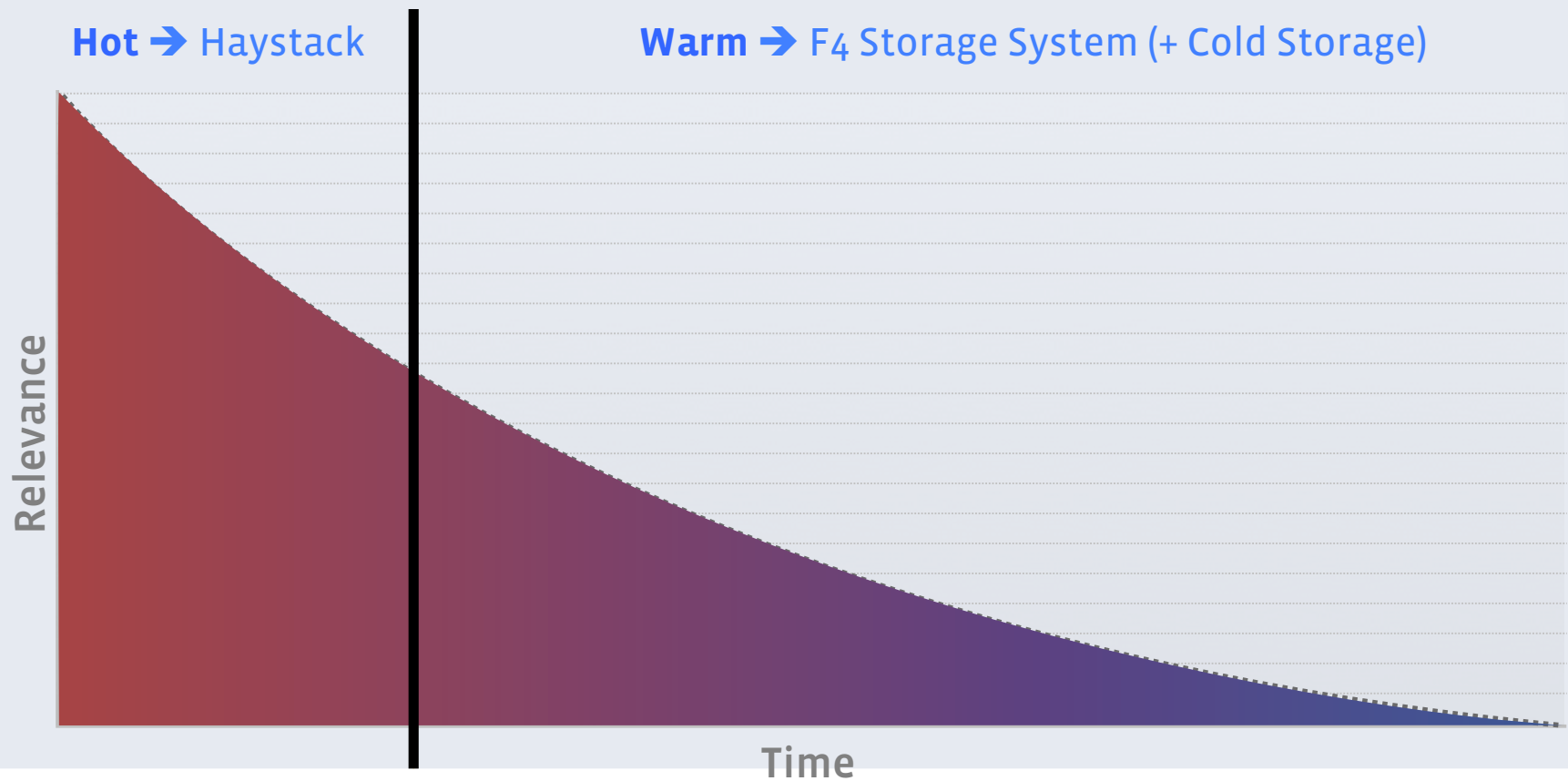


# Access patterns



# Generation 4: Tiered Storage

- Different storage solutions for hot, warm, cold photos



# Agenda

- 1** Overview
- 2** Datacenter Architecture
- 3** Case Study: Optimizing BLOB Storage system
- 4** Questions

**facebook**

(c) 2009 Facebook, Inc. or its licensors. "Facebook" is a registered trademark of Facebook, Inc.. All rights reserved. 1.0