

# 对象存储系统专题

施展 2019

# 课程信息

## 专题内容

### ➤ 关于对象存储系统的研究

- 一些基础

- <https://github.com/cs-course/iot-storage-experiment> 物联网3年级下学期
- <https://github.com/cs-course/obs-tutorial>

- 需要研究的问题

### ➤ 关于研究的研究

- 研究的基本要求和方法学习

# 课程信息



计算机系统设计2019秋  
扫一扫二维码，加入该群。

- ◆ 选课同学请实名入群
- ◆ 完成一个小投票
  - 本科学习过
    - 操作系统
    - 系统结构
    - 物联网数据存储实验
    - 均未学过



公告



文件



相册



群日历



群投票



群链接



群活动

# 授课目标

- ◆ 研究对象
  - 对象存储系统
    - 理解经典存储系统的设计
  
- ◆ 研究者
  - 基础研究能力
    - 明白未来研究的主要任务

# 新手上路

## ♦ 内心里

➤ 兴(MENG)奋(QUAN)的研究生生活开始了

## ♦ 然后

➤ 我是什么人，我要做什么？

➤ 研究的初心

# 新手上路

## ◆ 研究的初心

### 2.1 Motivation and Goals

- Why are you doing this? You must have a very good answer to this before you jump in to do a Ph.D. Don't do it if your main reason is "because I couldn't get a job" or "because there wasn't anything else I was particularly interested in doing at the moment..."
- It is very important to keep your goals and motivations in mind when in graduate school. Visualize what and where you would like to be in 5 years, and figure out what you need to do to get there.

[https://www.ece.rutgers.edu/~pompili/index\\_file/extra/HowToDoResearch\\_ANRG\\_WP02001.pdf](https://www.ece.rutgers.edu/~pompili/index_file/extra/HowToDoResearch_ANRG_WP02001.pdf)

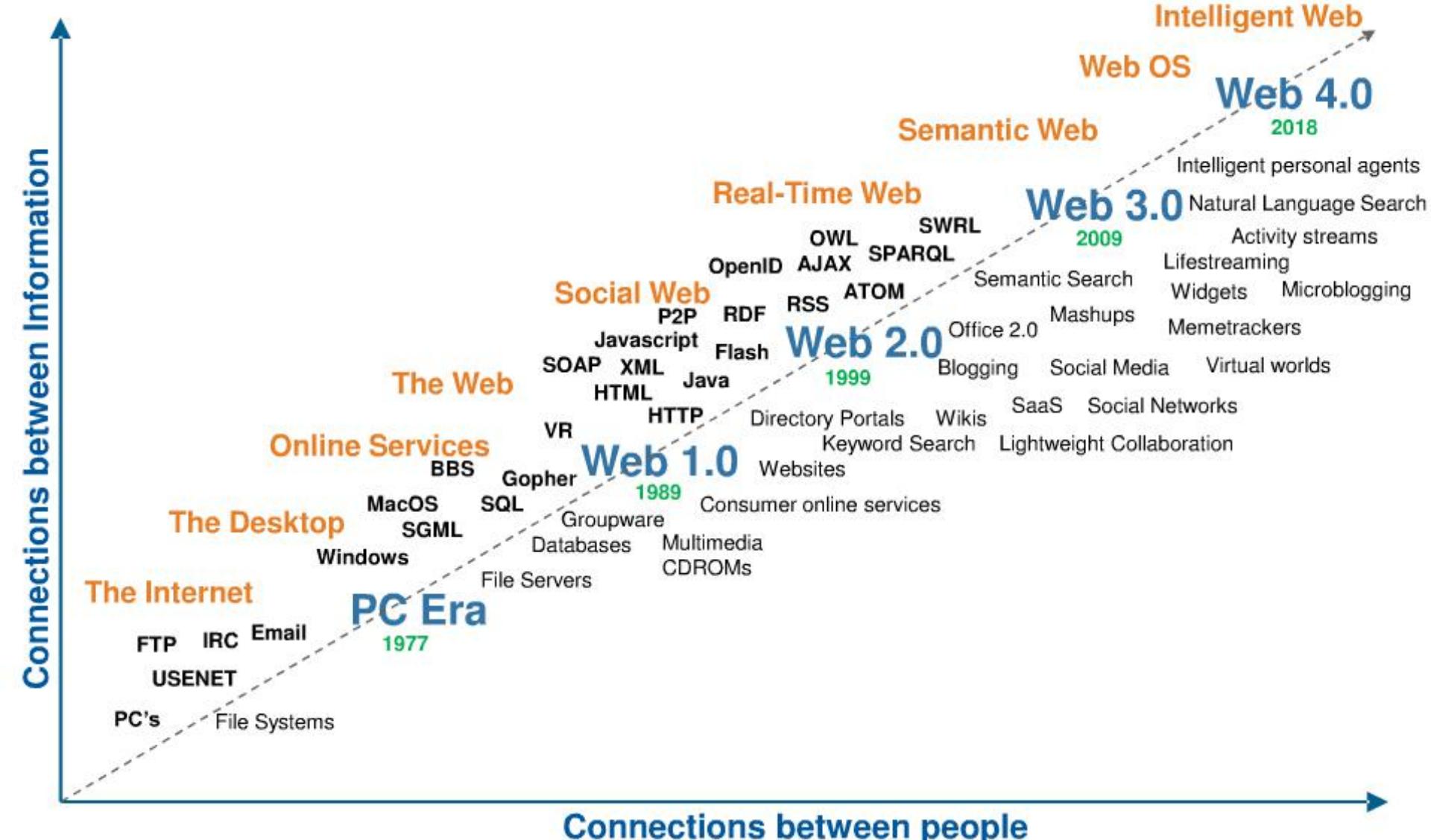
Read with a question in mind. "How can I use this?" "Does this really do what the author claims?" "What if...?" Understanding what result has been presented is not the same as understanding the paper. Most of the understanding

[https://dspace.mit.edu/bitstream/handle/1721.1/41487/AI\\_WP\\_316.pdf](https://dspace.mit.edu/bitstream/handle/1721.1/41487/AI_WP_316.pdf)

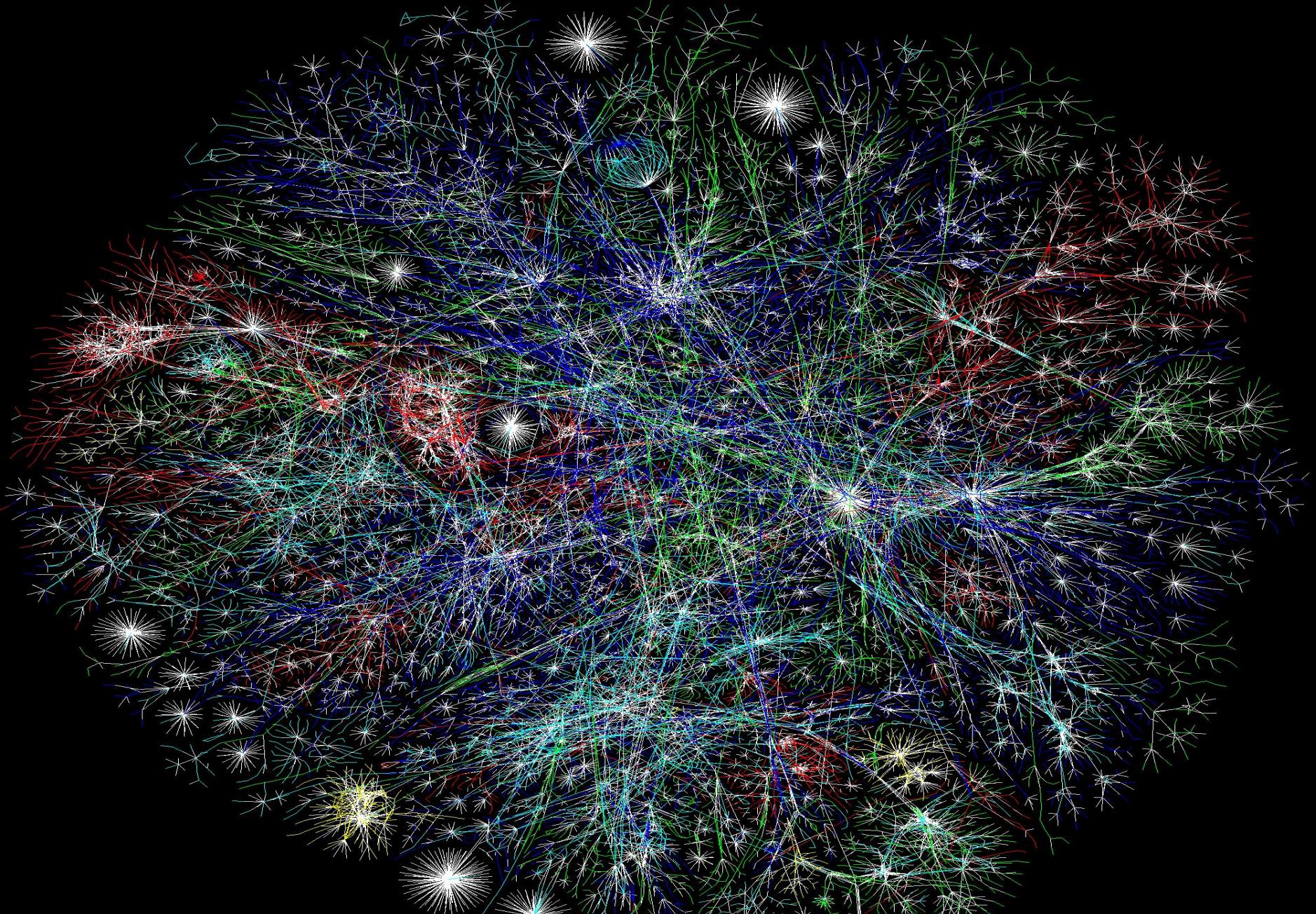
# 新手上路

- ◆ 首先
  - 从身旁熟悉的事物开始

# The Intelligence is in the Connections



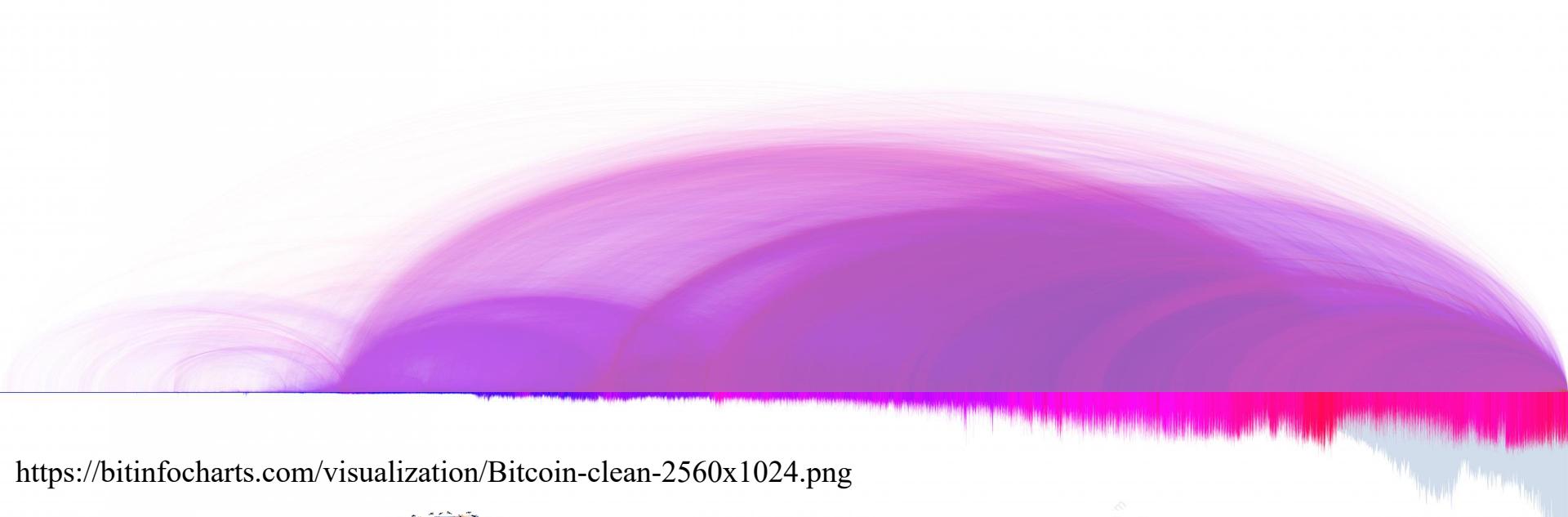
万维网持续进化



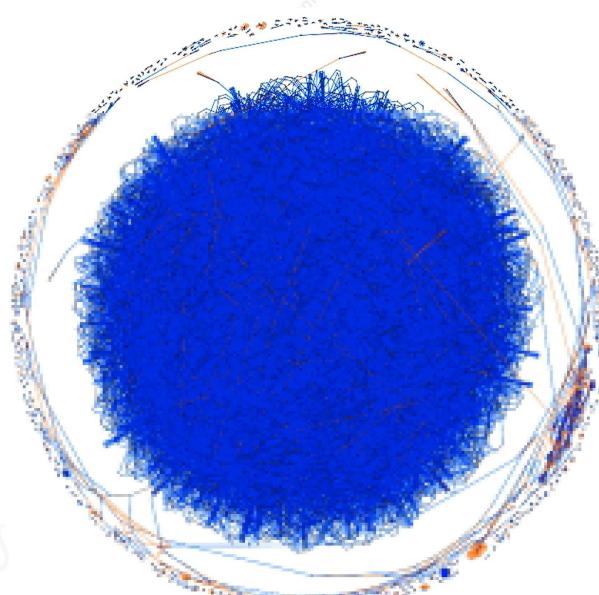
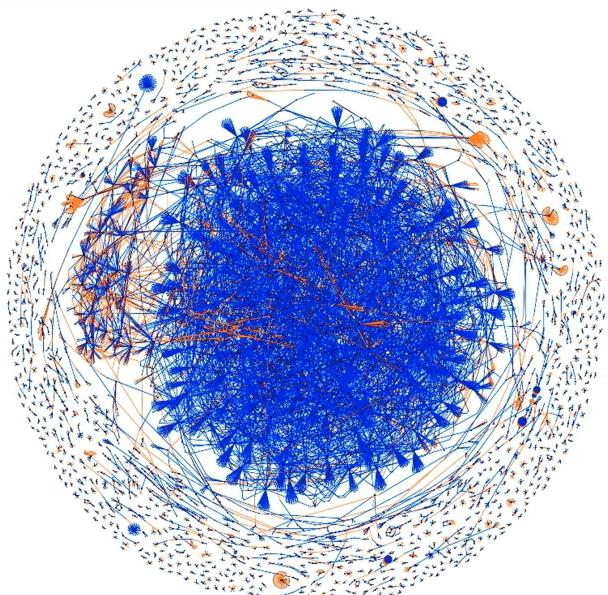
规模持续扩张



内容异常丰富



<https://bitinfocharts.com/visualization/Bitcoin-clean-2560x1024.png>



<https://datastori.es/107-visualizing-bitcoin-with-dan-mcginn/>

数据各处传播

# 身旁腾讯的现实

## 整体存储量规模趋势 (PB)



- 社交、游戏，访问密度高达**100万次/秒/100GB**量级的数据读写；
- 在线业务，应保证良好的用户体验，不论数据访问密集程度如何，均要求延迟在**100毫秒以内**



月活跃用户 **>8亿**  
同时在线用户 **>2亿**



月活跃用户 **>6亿**



QQ空间相册

图片 **4000亿张**  
存储量 **200PB**



微信朋友圈

日上传 **10亿张**  
日下载 **1200亿张**

# 整个数字世界

数据存储挑战  
庞大：规模持续扩张  
复杂：内容结构丰富

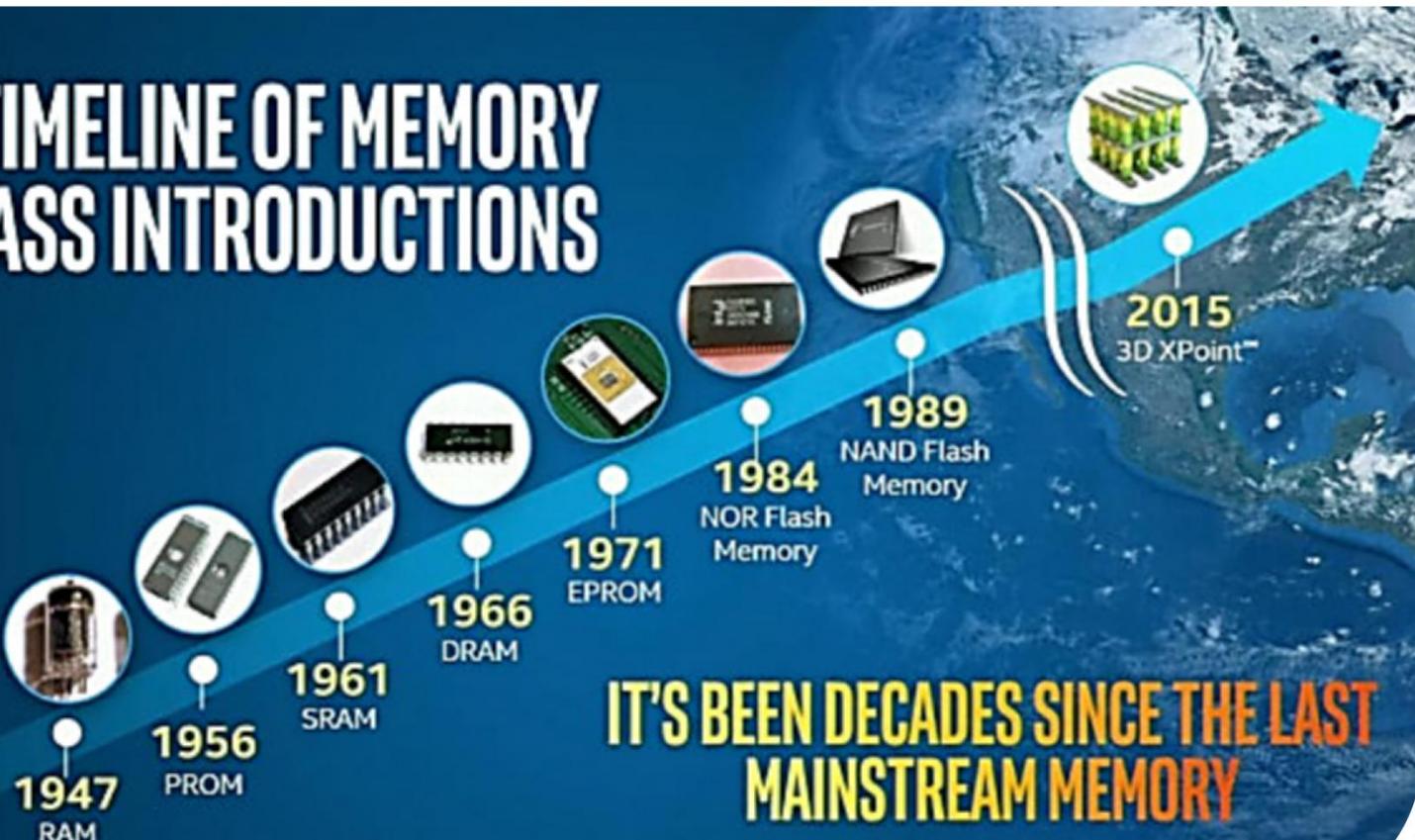


Source: Mario Morales, IDC

## 该怎么办？

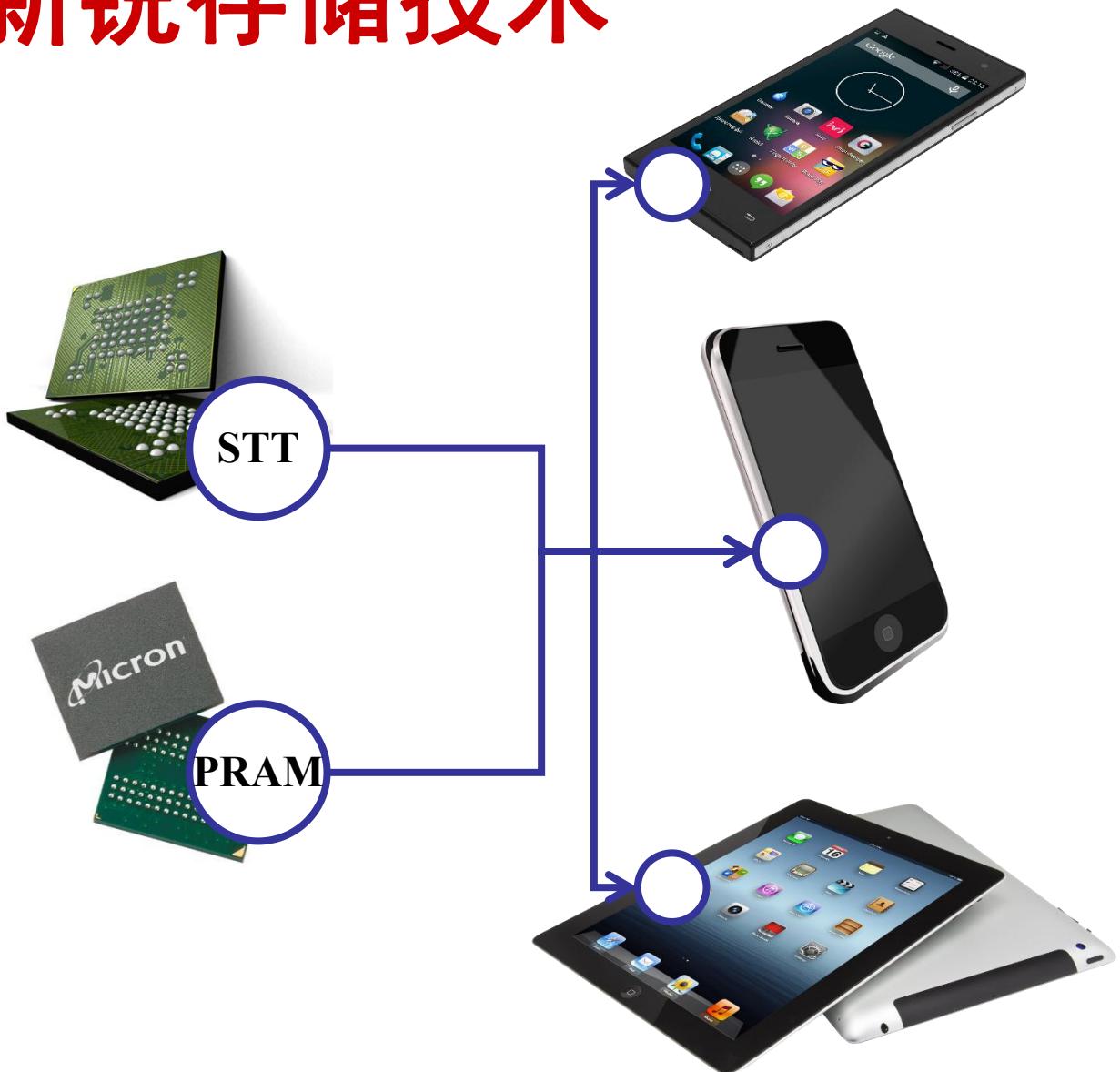
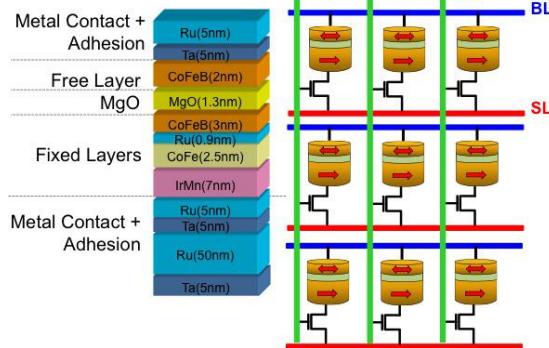
# 信息存储发展

## A TIMELINE OF MEMORY CLASS INTRODUCTIONS



# 新锐存储技术

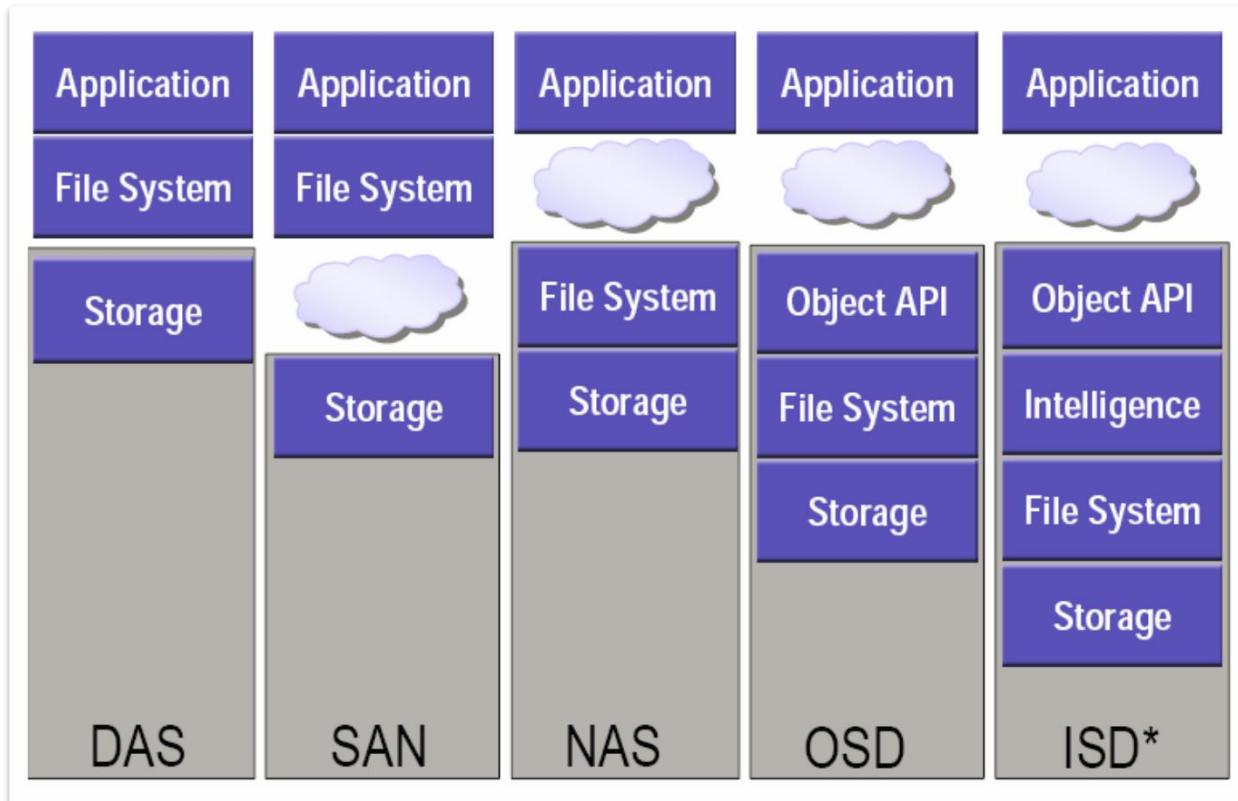
只靠这些？



还远远不够：数据规模、内容种类……

# 从战术上升至战略

## ✿ 系统层面还能提供什么干货？



# 第一部分

# 对象存储

## Abstract

- Object Storage is a generalized data container with uses in cloud storage, HPC file systems, and custom applications that provide their own indexing and metadata layers over objects. This tutorial provides a survey of these different kinds of objects, their APIs, and the applications that use them. The OSD (Object Storage Device) command set for SCSI provides a secure, general purpose mechanism used in HPC file systems via the NFSv4.1 pNFS (parallel NFS) protocol. The Amazon S3 object interface uses a web-based transport to provide cloud-based storage, and there are a number of similar interfaces in the open source community such as OpenStack Swift and Eucalyptus. The differentiating features of object storage systems include the access protocols (SCSI, RPC, REST), performance (high-speed LAN or WAN), security mechanisms, replication and reliability schemes, metadata and indexing services. As a result, different application domains have evolved their own Object Storage ecosystems.

# 历史

## Agenda

- Object storage background and history
  - ◆ Abstract data containers
  - ◆ NASD, OSD, HTTP
- Objects and File Systems
  - ◆ Lustre, PanFS, Ceph, many others
- Objects and Web Storage
  - ◆ S3, Azure, Swift, many others



# Two Paths to Objects

## 历史

### ➤ Storage Devices

- ◆ Move smarts into the device (NASD)
  - > Network Attached Secure Disk
- ◆ Raise the level of abstraction
  - > Containers
  - > Attributes
  - > Security
- ◆ SCSI Model
  - > OSD command set

### ➤ Web Services

- ◆ Add storage abstraction to a web-based system
  - > Containers
  - > Metadata
  - > Security
- ◆ REST Model
  - > HTTP protocol

# 历史

## Some History

### ➤ NASD (Network Attached Secure Disk)

- ◆ 1990's research by Dr. Garth Gibson about moving intelligence into the storage device
- ◆ Google cites NASD as inspiration for data node in its file system

### ➤ OSD (Object-based Storage Device)

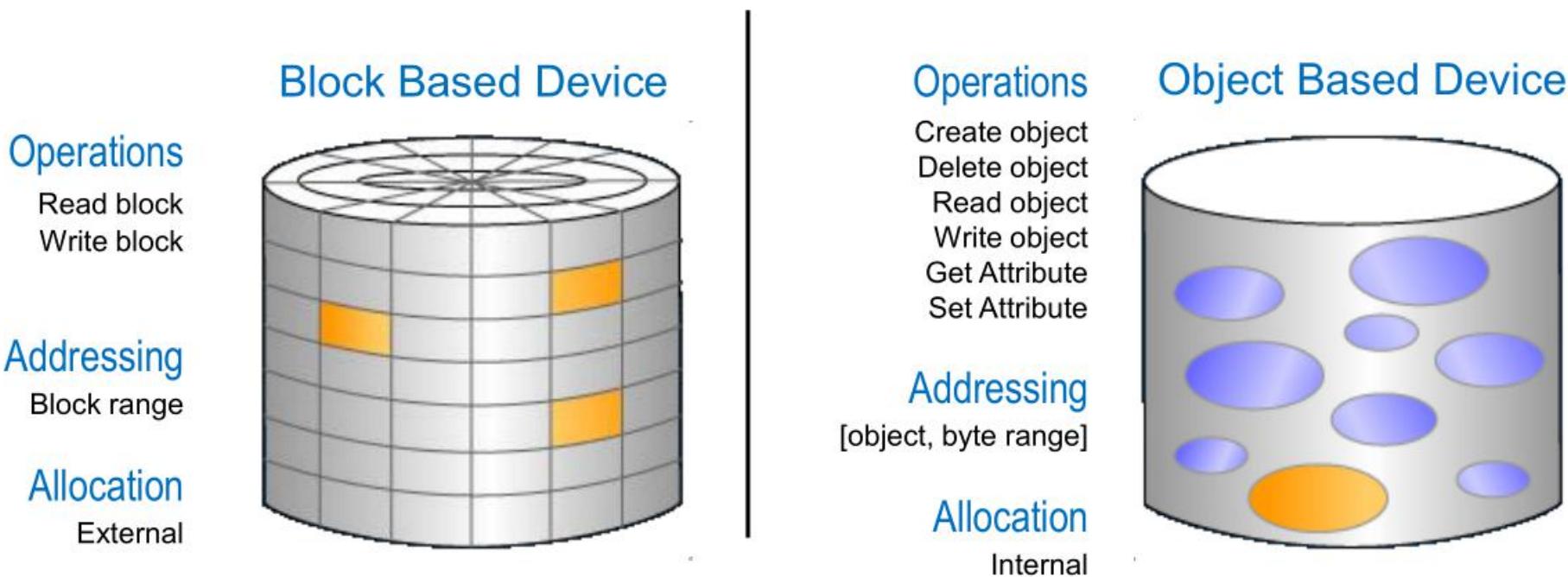
- ◆ Standards effort in the early 2000's created a SCSI command set for objects
- ◆ There is a storage device behind this interface

### ➤ HTTP (HyperText Transfer Protocol)

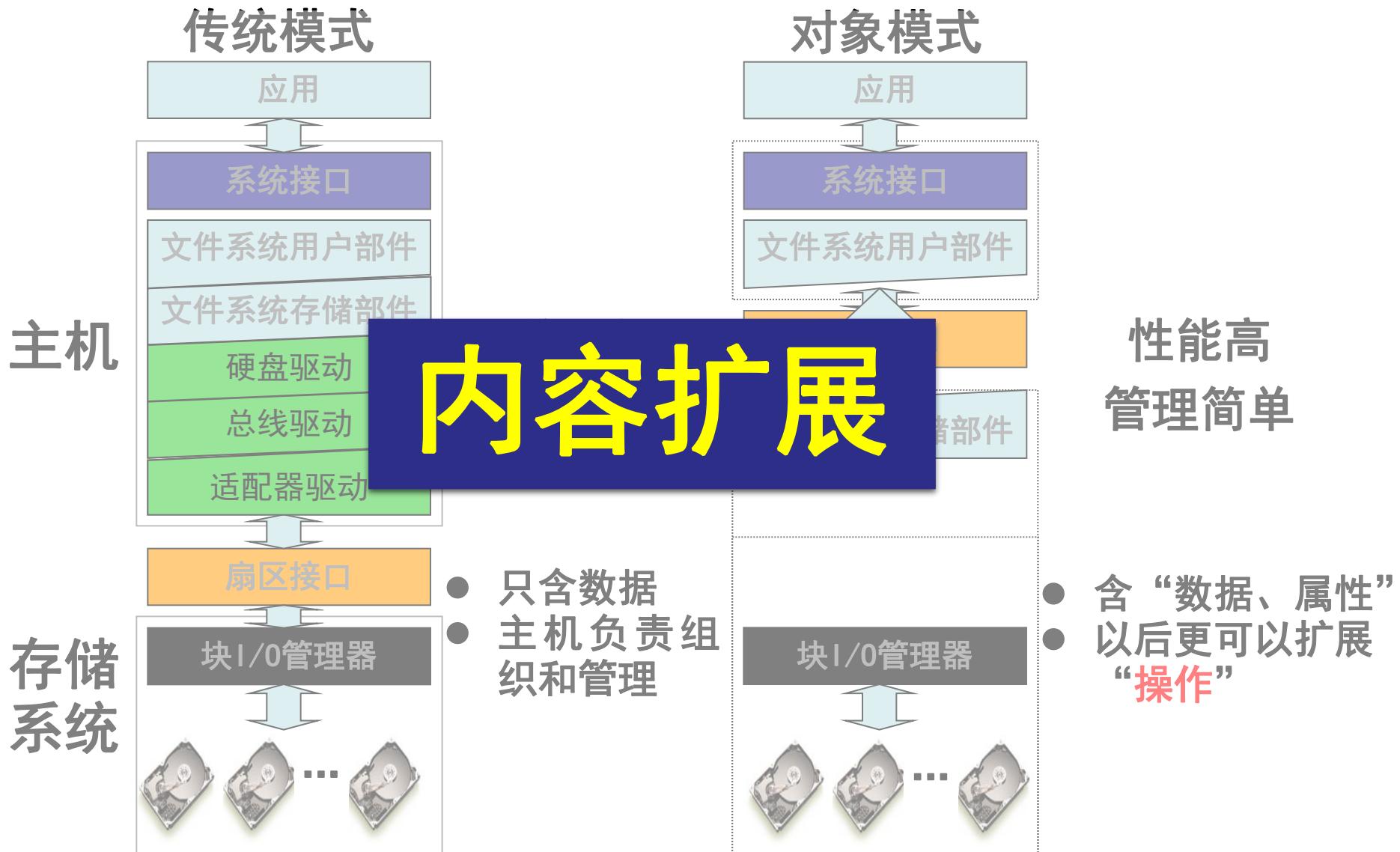
- ◆ A simple put/get protocol for the world-wide web
- ◆ There is an arbitrary service behind this interface

# 基础概念

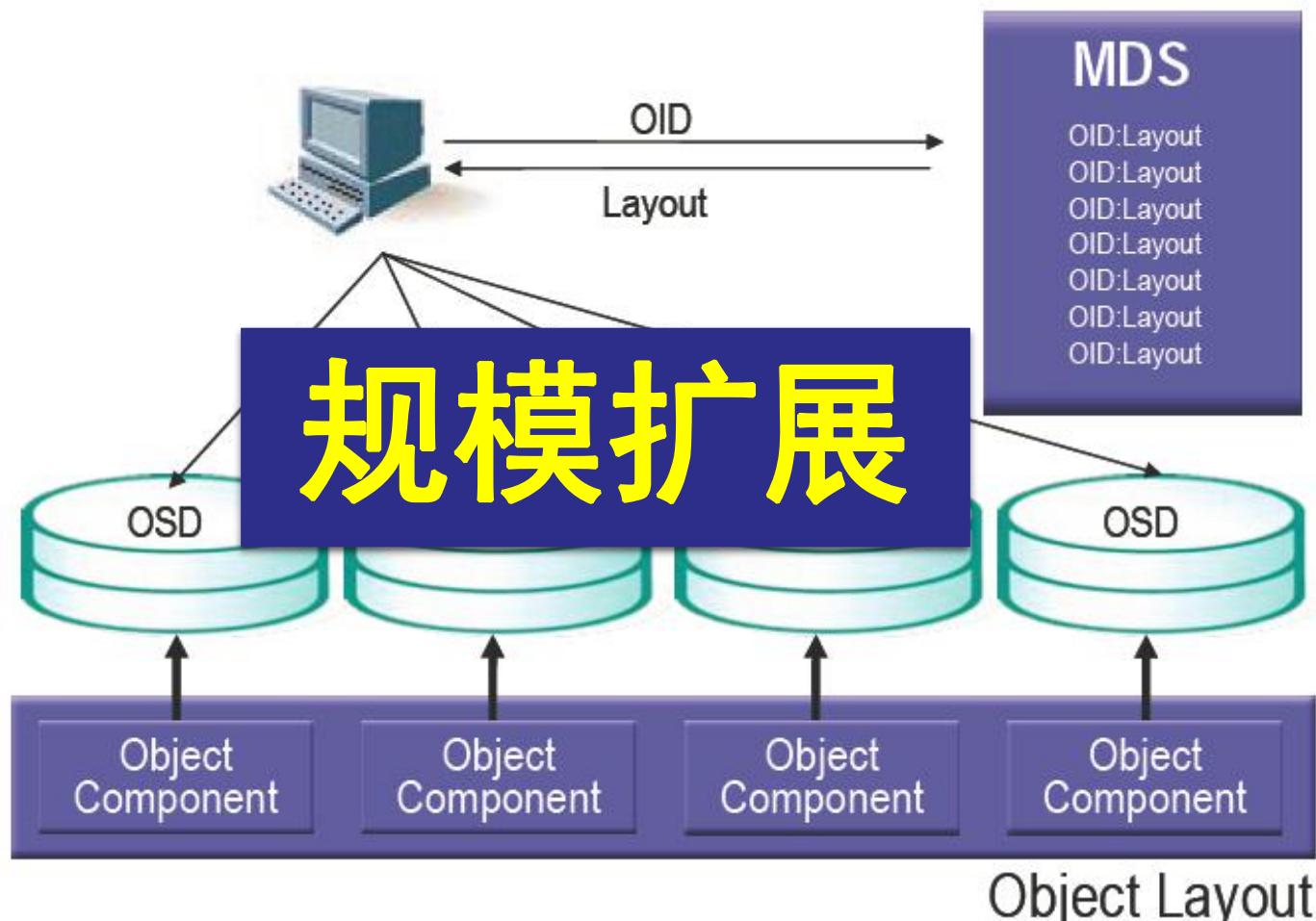
- Objects are containers for data and attributes
  - ◆ Every file system has an *inode* that is data blocks plus attributes
  - ◆ They are created, deleted, read, written, and have attributes



# 对象存储技术

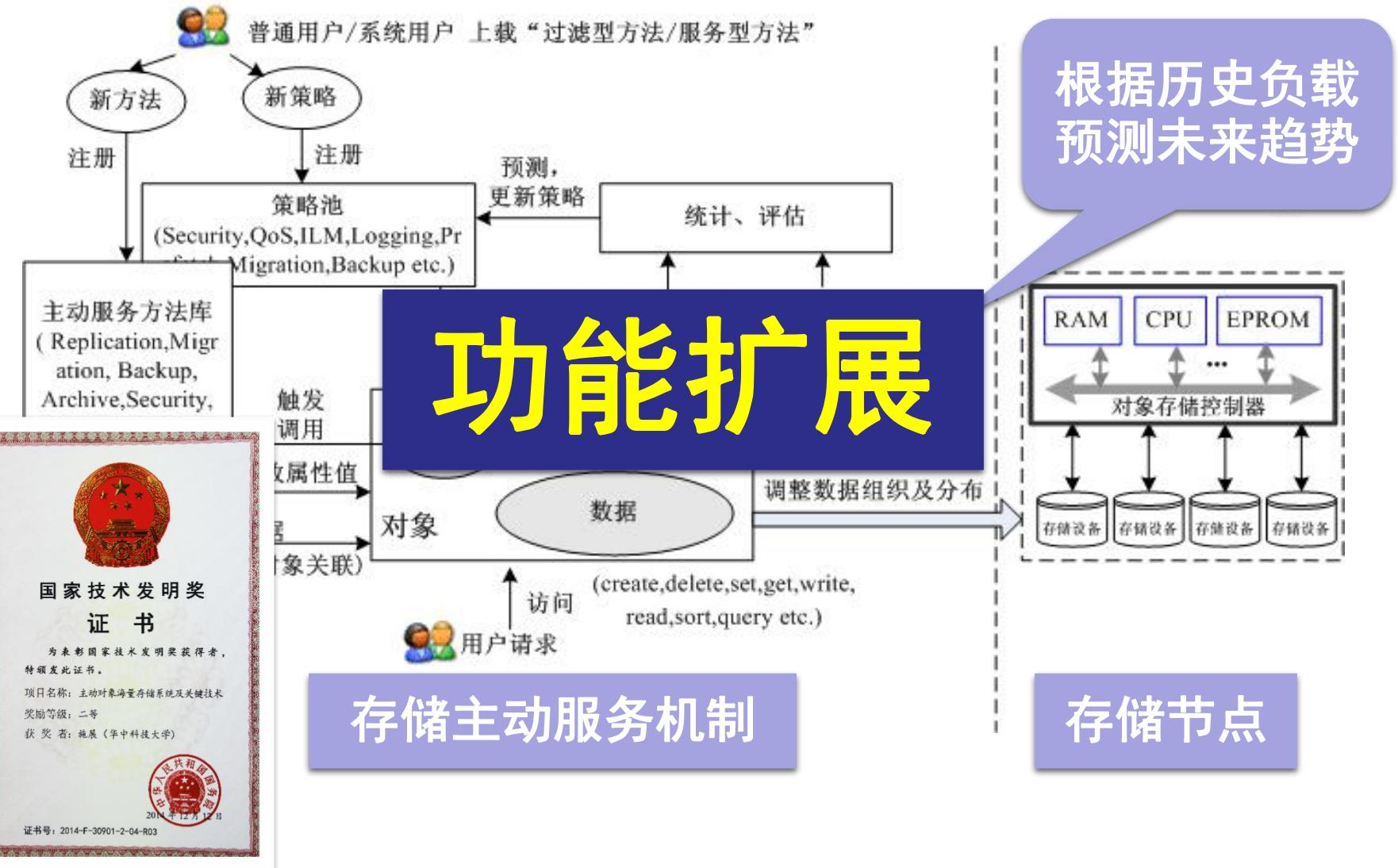


# 对象存储系统

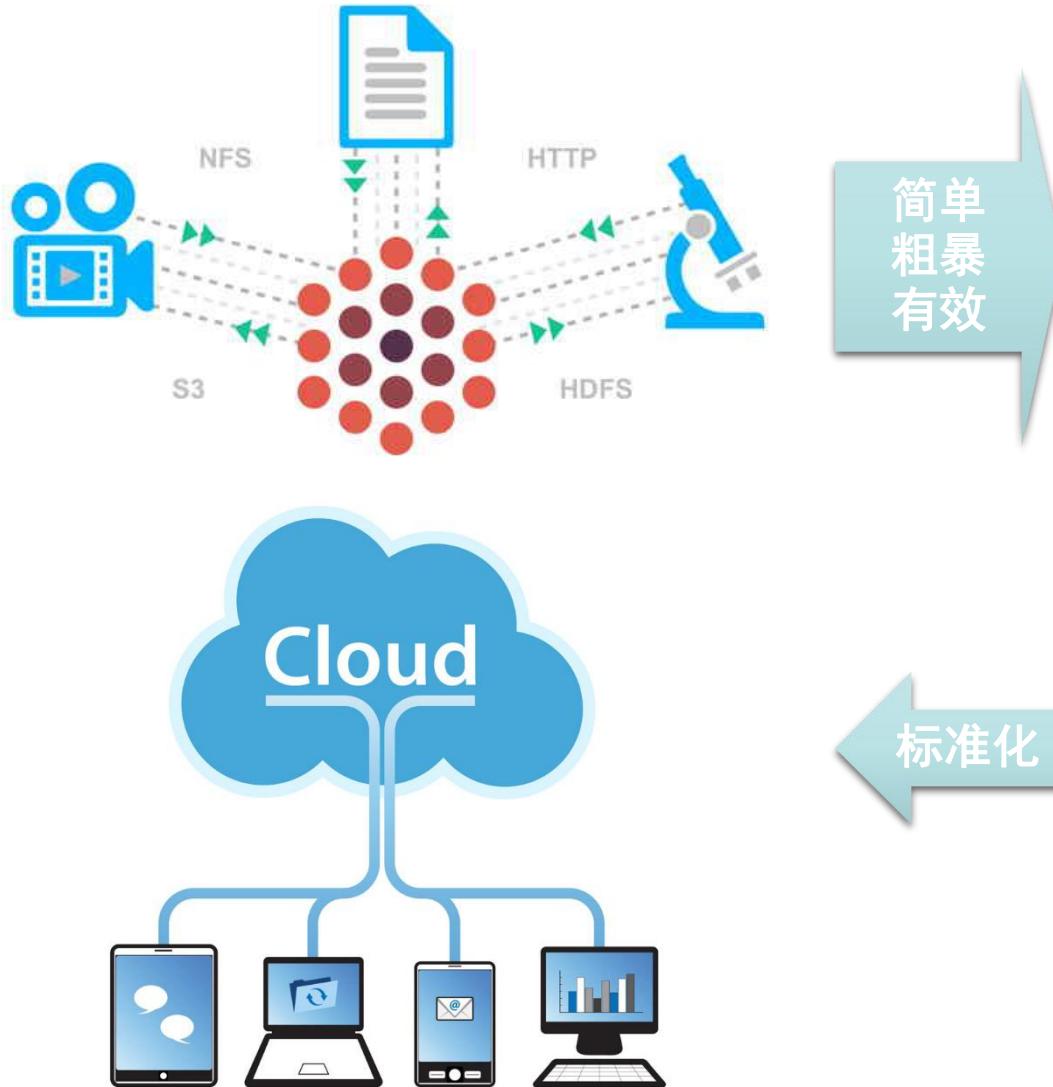


# 元数据由此解耦

# 主动对象存储



# 工业界广泛应用



Google Cloud Platform



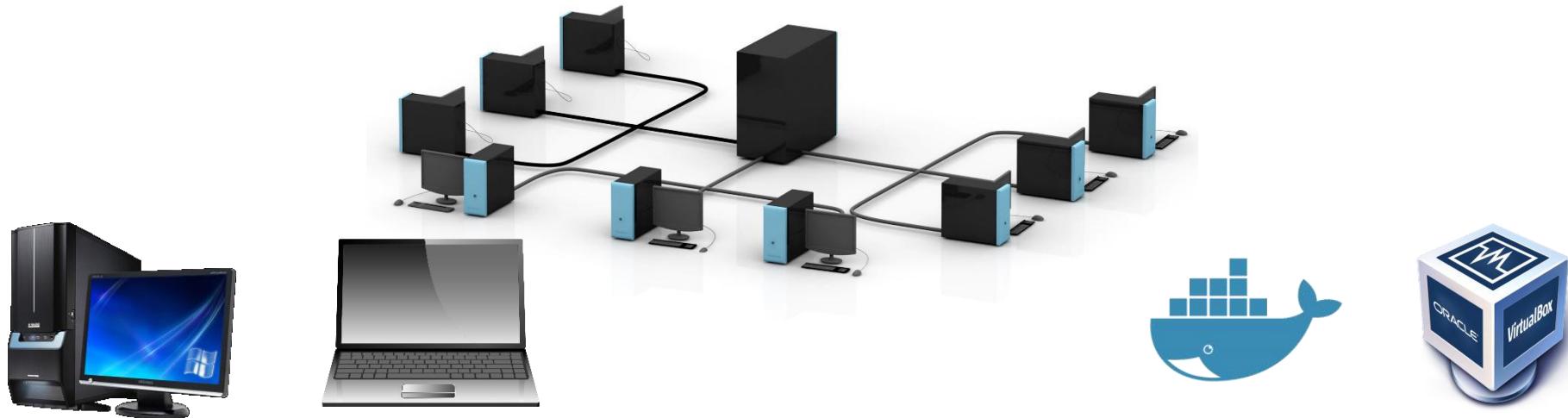
# 初步实践

## ◆ 说明

- [https://github.com/cs-course/obs-tutorial/blob/master/README.zh\\_cn.md](https://github.com/cs-course/obs-tutorial/blob/master/README.zh_cn.md)

## ◆ 原材料

- Git、Github/Gitlab/Bitbucket
- Python、Go、Java
- 可用虚拟机或容器



# 性能观测

## Test parameters

```
endpoint(s): [http://127.0.0.1:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0010 MB
numClients: 10
numSamples: 100
verbose: %!d(bool=false)
```

## Results Summary for Write Operation(s)

```
Total Transferred: 0.098 MB
Total Throughput: 0.26 MB/s
Total Duration: 0.381 s
Number of Errors: 0
```

---

```
Write times Max: 0.062 s
Write times 99th %ile: 0.062 s
Write times 90th %ile: 0.053 s
Write times 75th %ile: 0.042 s
Write times 50th %ile: 0.034 s
Write times 25th %ile: 0.030 s
Write times Min: 0.019 s
```

## Results Summary for Read Operation(s)

```
Total Transferred: 0.098 MB
Total Throughput: 2.18 MB/s
Total Duration: 0.045 s
Number of Errors: 0
```

---

```
Read times Max: 0.016 s
Read times 99th %ile: 0.016 s
Read times 90th %ile: 0.008 s
Read times 75th %ile: 0.005 s
Read times 50th %ile: 0.003 s
Read times 25th %ile: 0.003 s
Read times Min: 0.001 s
```

Cleaning up 100 objects...

Deleting a batch of 100 objects in range {0, 99}... Succeeded  
Successfully deleted 100/100 objects in 94.8232ms

## Test parameters

```
endpoint(s): [http://192.168.3.85:9000]
bucket: loadgen
objectNamePrefix: loadgen
objectSize: 0.0156 MB
numClients: 8
numSamples: 256
verbose: %!d(bool=false)
```

## Results Summary for Write Operation(s)

```
Total Transferred: 4.000 MB
Total Throughput: 1.54 MB/s
Total Duration: 2.604 s
Number of Errors: 0
```

---

```
Write times Max: 0.145 s
Write times 99th %ile: 0.145 s
Write times 90th %ile: 0.100 s
Write times 75th %ile: 0.089 s
Write times 50th %ile: 0.076 s
Write times 25th %ile: 0.068 s
Write times Min: 0.060 s
```

## Results Summary for Read Operation(s)

```
Total Transferred: 4.000 MB
Total Throughput: 58.21 MB/s
Total Duration: 0.069 s
Number of Errors: 0
```

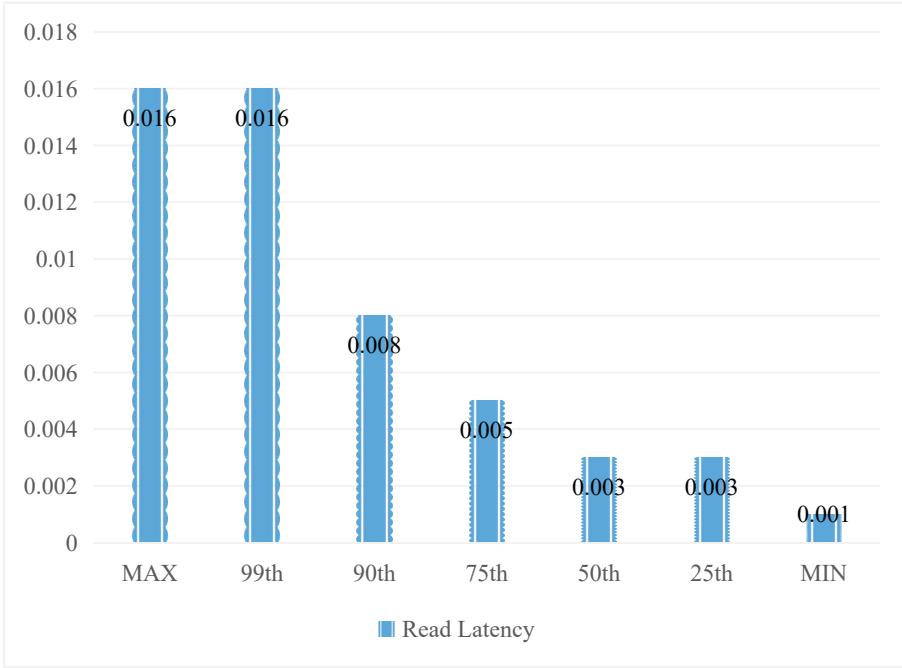
---

```
Read times Max: 0.005 s
Read times 99th %ile: 0.005 s
Read times 90th %ile: 0.003 s
Read times 75th %ile: 0.002 s
Read times 50th %ile: 0.002 s
Read times 25th %ile: 0.002 s
Read times Min: 0.001 s
```

Cleaning up 256 objects...

Deleting a batch of 256 objects in range {0, 255}... Succeeded  
Successfully deleted 256/256 objects in 21.493393ms

# 找出问题



## Results Summary for Read Operation(s)

Total Transferred: 0.098 MB

Total Throughput: 2.18 MB/s

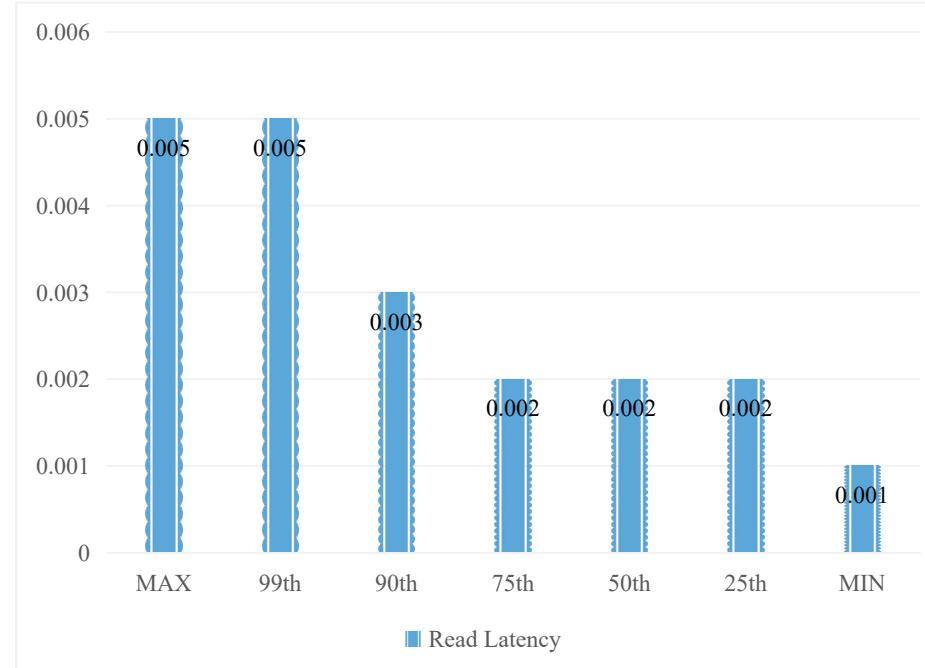
Total Duration: 0.045 s

Number of Errors: 0

Read times Max: 0.016 s  
Read times 99th %ile: 0.016 s  
Read times 90th %ile: 0.008 s  
Read times 75th %ile: 0.005 s  
Read times 50th %ile: 0.003 s  
Read times 25th %ile: 0.003 s  
Read times Min: 0.001 s

Cleaning up 100 objects...

Deleting a batch of 100 objects in range {0, 99}... Succeeded  
Successfully deleted 100/100 objects in 94.8232ms



## Results Summary for Read Operation(s)

Total Transferred: 4.000 MB

Total Throughput: 58.21 MB/s

Total Duration: 0.069 s

Number of Errors: 0

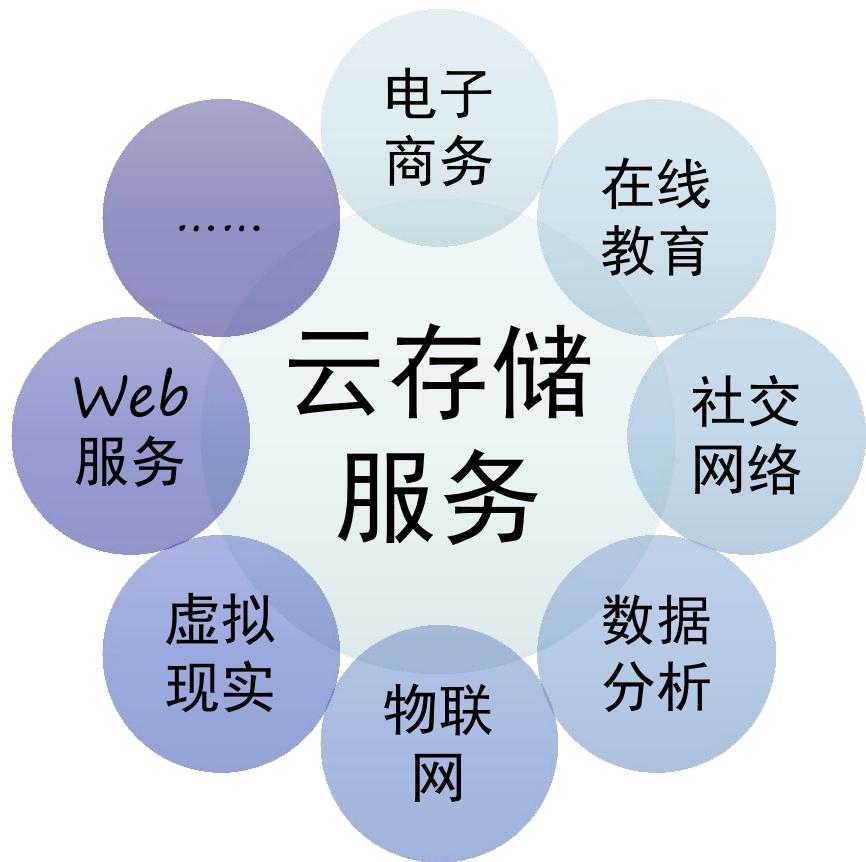
Read times Max: 0.005 s  
Read times 99th %ile: 0.005 s  
Read times 90th %ile: 0.003 s  
Read times 75th %ile: 0.002 s  
Read times 50th %ile: 0.002 s  
Read times 25th %ile: 0.002 s  
Read times Min: 0.001 s

Cleaning up 256 objects...

Deleting a batch of 256 objects in range {0, 255}... Succeeded  
Successfully deleted 256/256 objects in 21.493393ms

# 意味着什么

- 现代应用的**交互性**需求日益增强，要求其底层的云存储服务具有**低响应延迟**

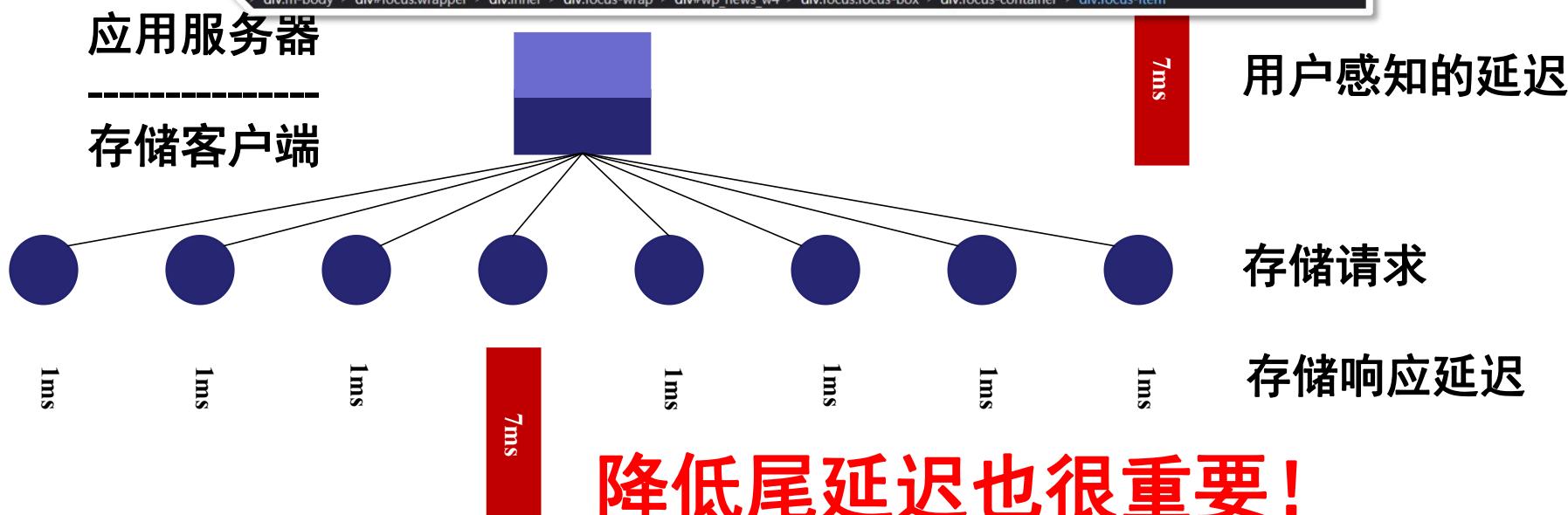


# 意味着什么

- 面一应



的



# 本讲小结

◆ 对象存储基础知识

➤ 为云存储数据

◆ 用于解决什么问题？

➤ 扩展性

◆ 当前存在什么问题？

➤ 性能保障

研究对象

要有矿

没挖光

研究目标