云计算与标签化

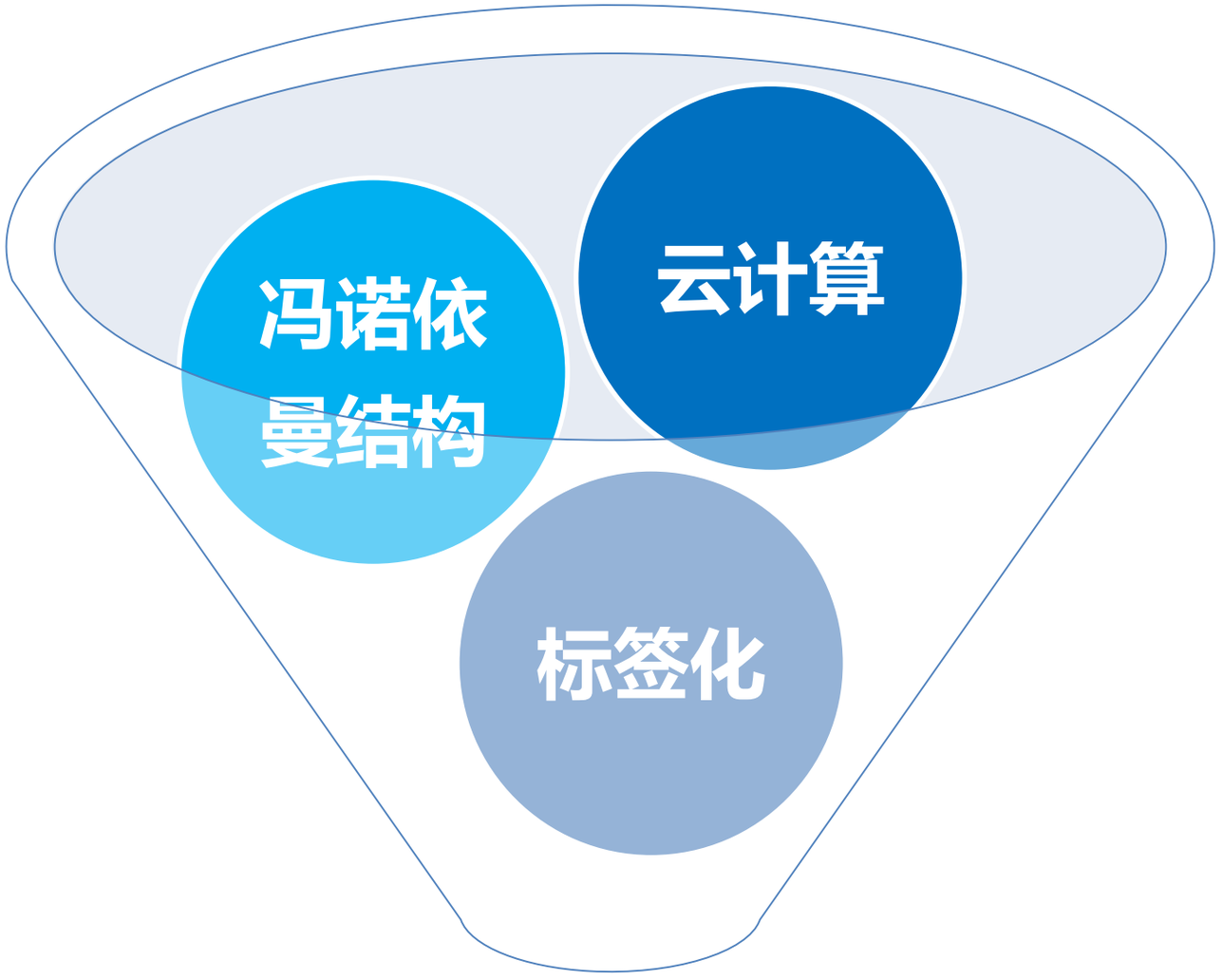
冯诺依曼结构

包云岗

中科院计算所

2016.10.21 @ 太原

一个目标，三个关键词

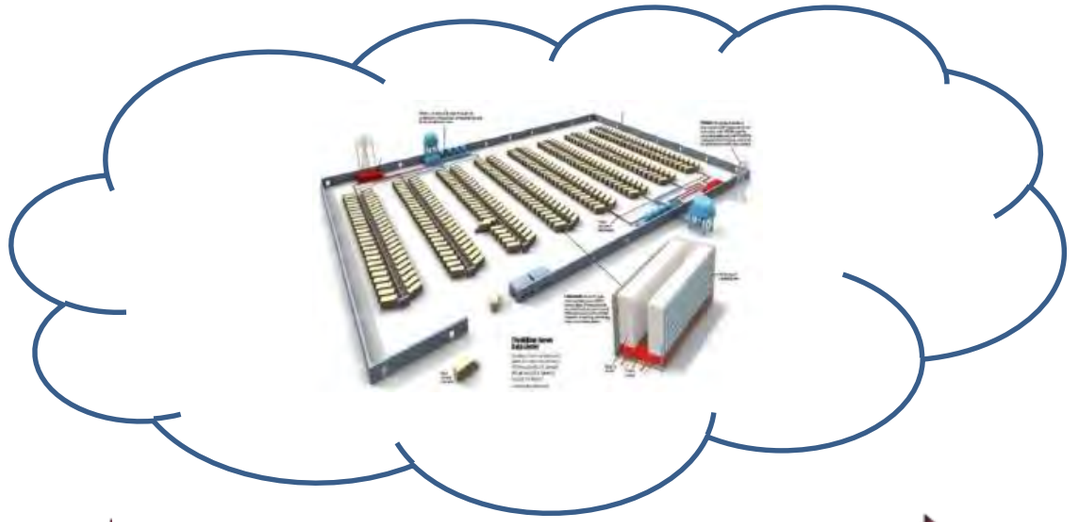


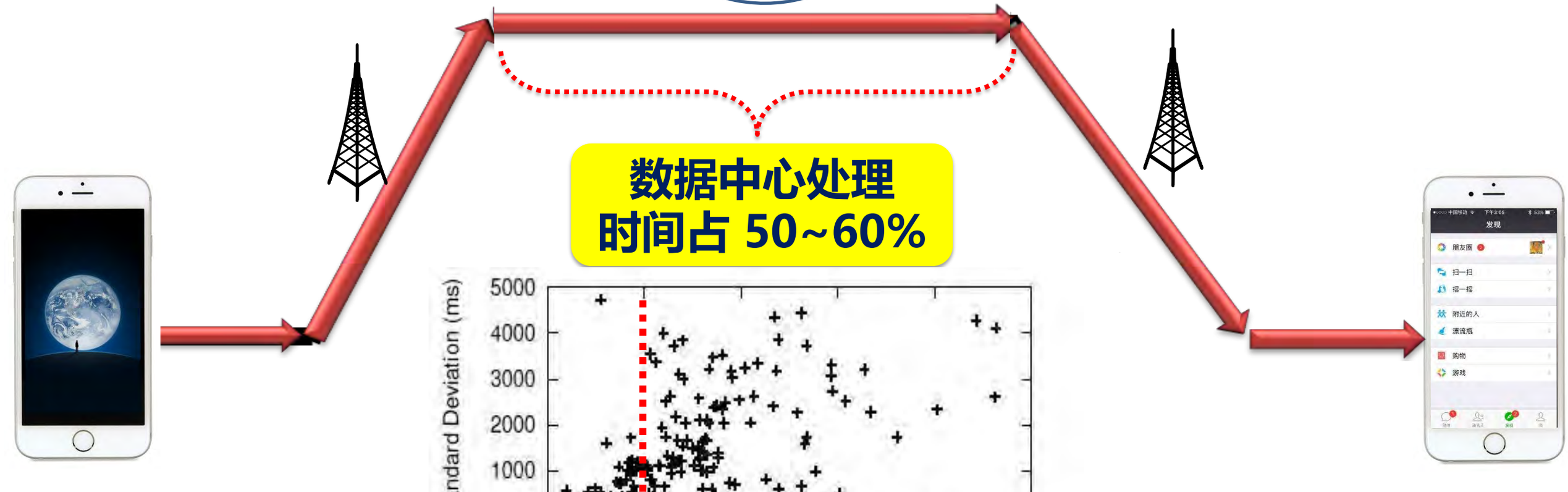
降低云计算数据中心成本


SUMMER PARTY
Bai du 百度
2014百度最高奖颁奖典礼


Bai du 百度
挑战巅峰 脱颖而出
2012百度最高奖颁奖典礼

数据中心支撑“移动+云”新模式





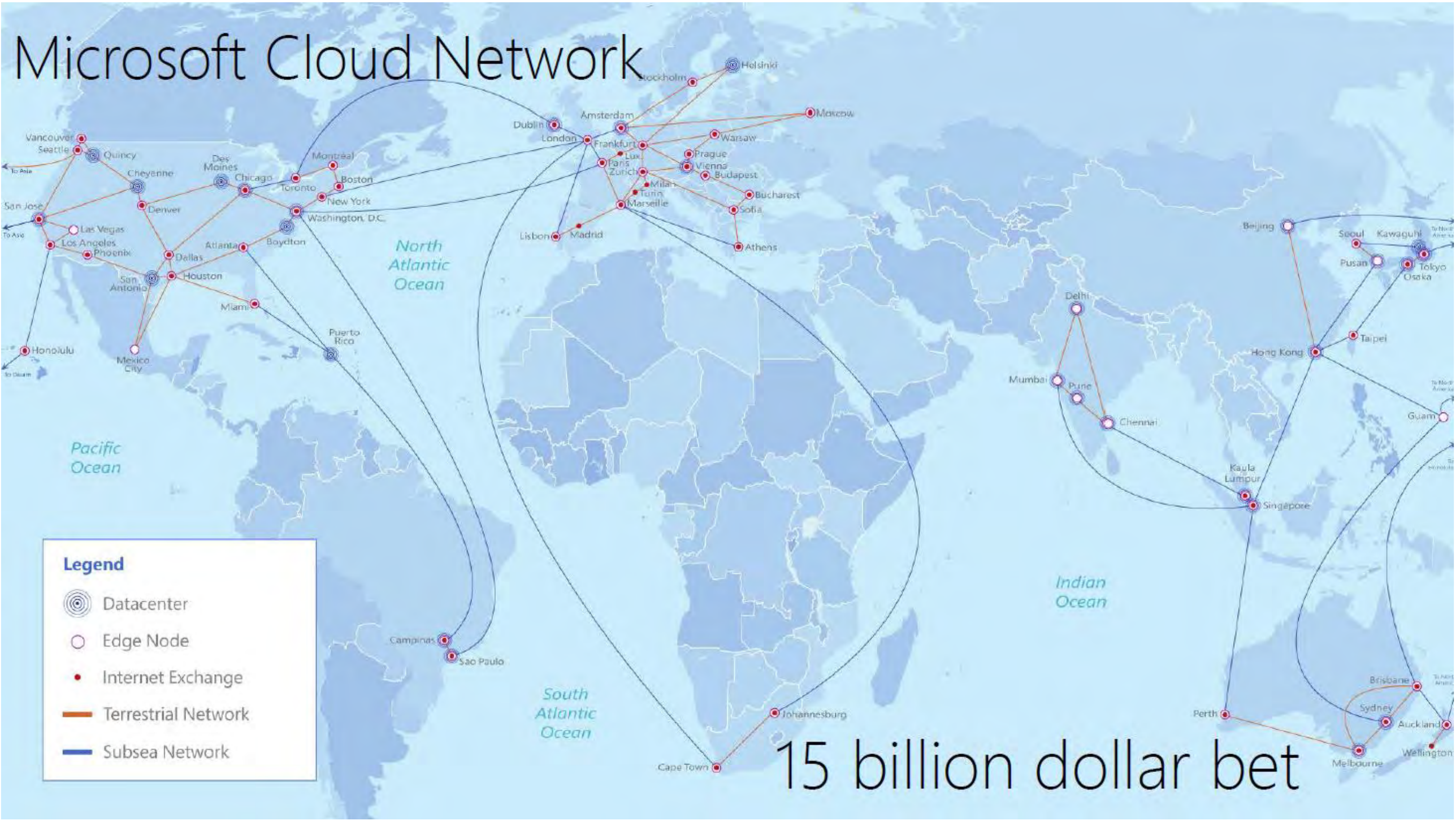
[1] L. Ravindranath et al., Timecard: Controlling User-Perceived Delays in Server-Based Mobile Applications, SOSP, 2013.

数据中心是云计算核心竞争力

- I claim there really are almost no companies in the world, just a handful, that are really investing in scaled public cloud infrastructure. **（我敢说世界上几乎没有公司能投资大规模公有云基础设施，屈指可数）**
- We have something over a million servers in our data center infrastructure. Google is bigger than we are. Amazon is a little bit smaller. ... So the number of companies that really understand the network topology, the data center construction, the server requirements to build this public cloud infrastructure is very, very small. **（我们的数据中心已超过100万台服务器，谷歌比我们更大一些，亚马逊比我们稍小一点，..., 所以真正能理解和管理如此大规模数据中心提供公有云服务的企业非常非常少）**

—鲍尔默，微软前CEO，2013

微软数据中心耗资150亿美元



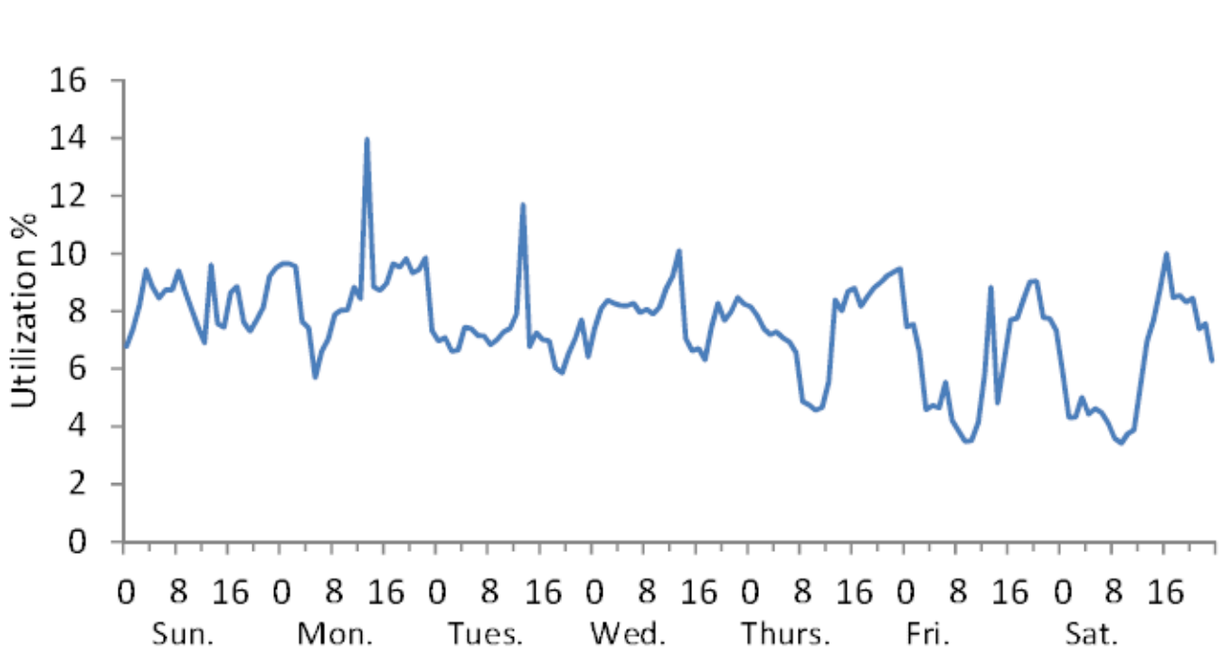
[1] L. Albert Greenberg, SDN for the Cloud, SIGCOMM Keynote, 2015.

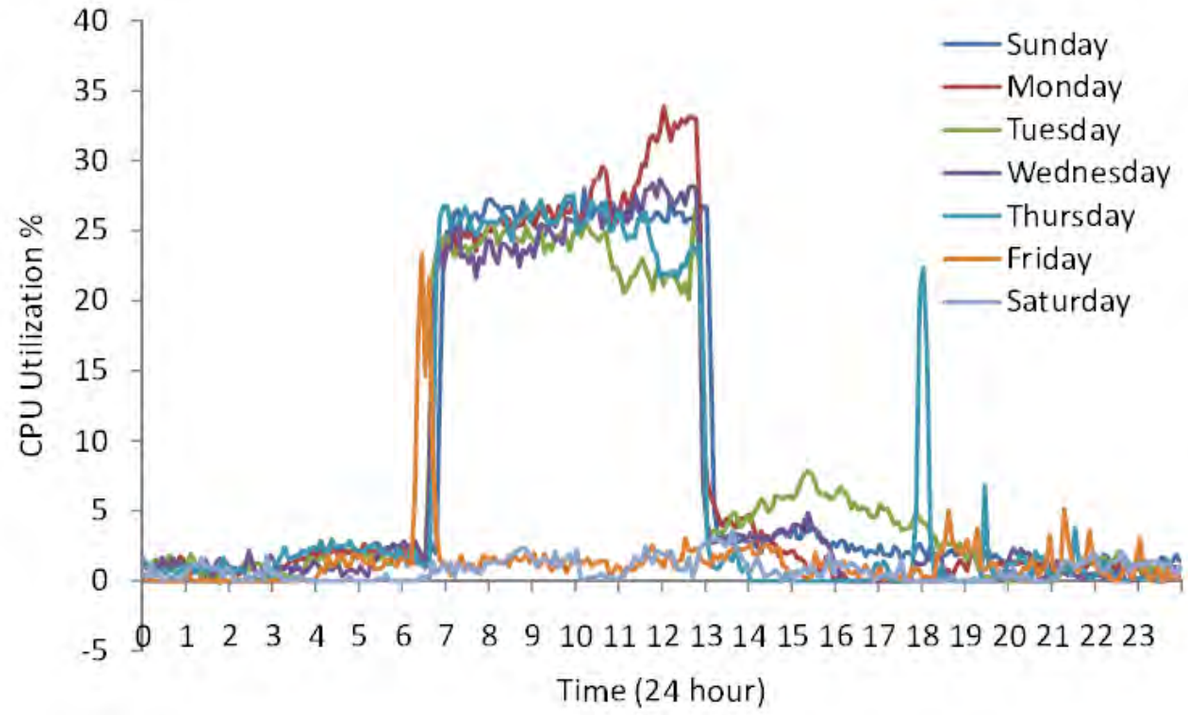
阿里张北数据中心耗资￥180亿

[1] “阿里绿色智能数据中心落户张北 将成北方数据心脏”, 阿里云资讯, 2016.

利用率？

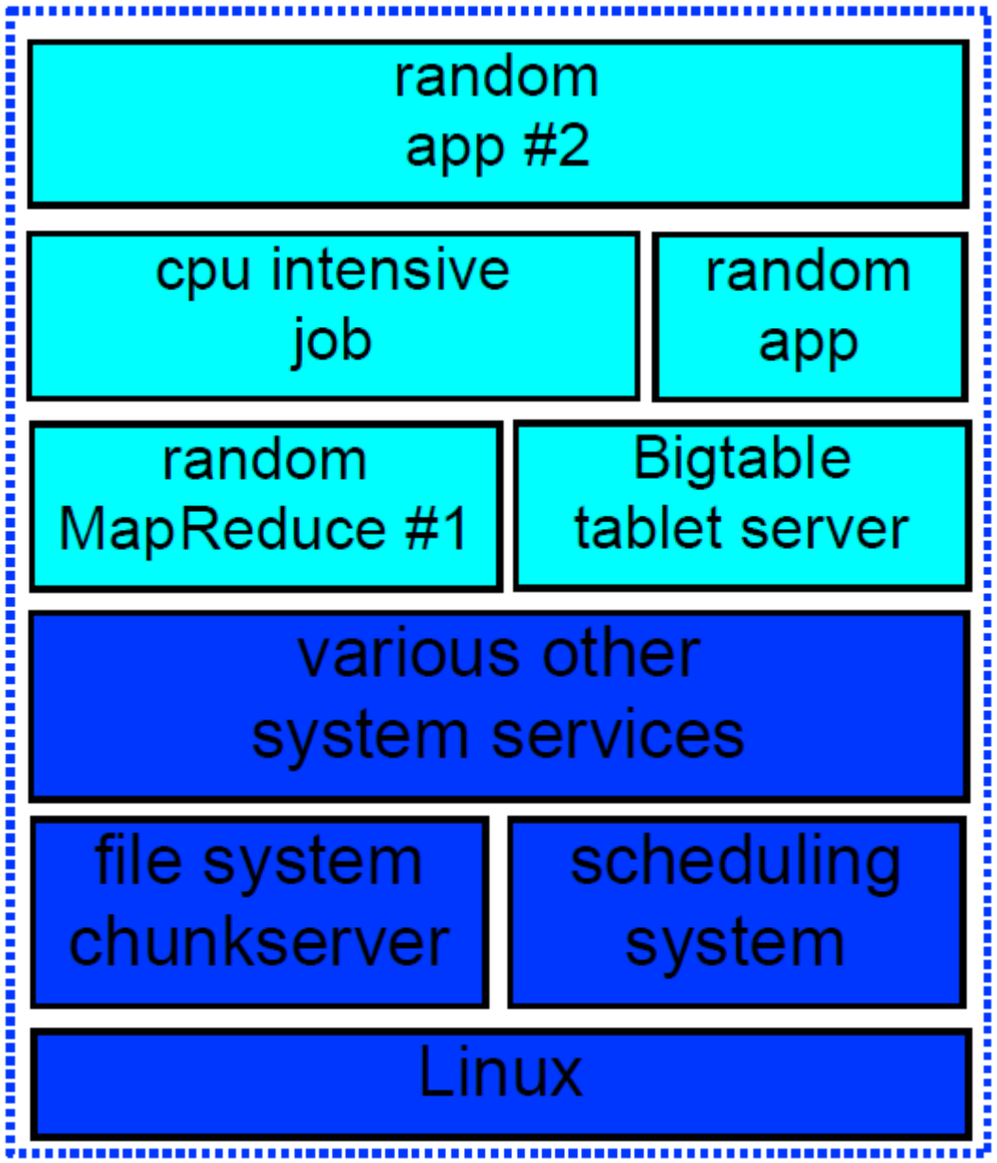
- 盖特纳/麦肯锡调研数据: **6%~12%**
- 亚马逊AWS EC2平均CPU利用率：**7%~17%**





[1] http://www.gartner.com/newsroom/id/1472714.
[2] J. M. Kaplan, W. Forrest, and N. Kindler. Revolutionizing data center energy efficiency. McKinsey & Company, 2008.
[3] Huan Liu, A Measurement Study of Server Utilization in Public Clouds, 2011.

共享！提高资源利用率



服务器硬件

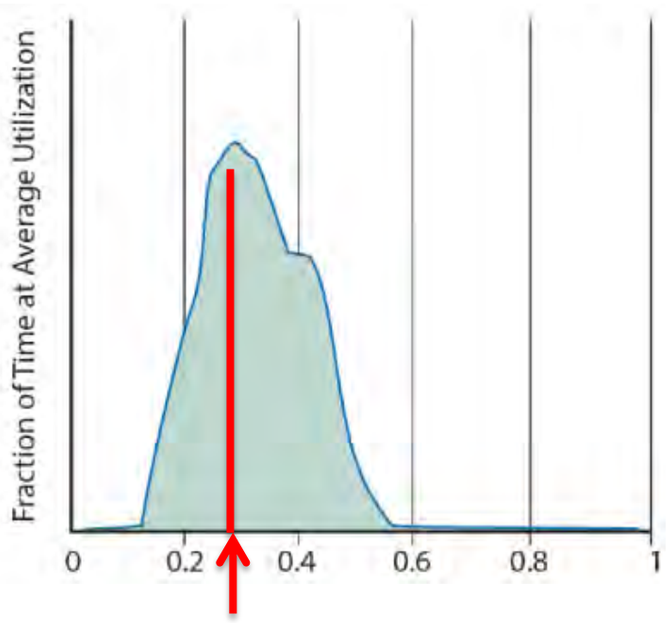
CPU、内存、磁盘、网络

[1] J. Dean, “Achieving Rapid Response Times in Large Online Services”, talk at Berkeley, 2012.

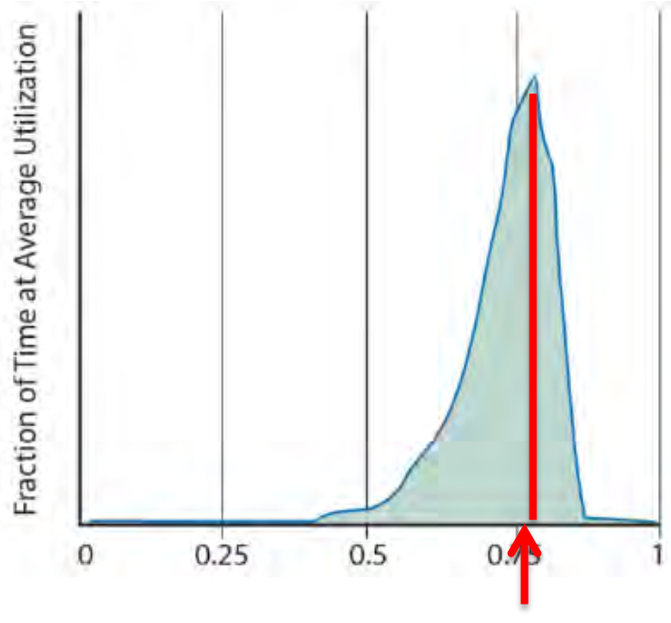
谷歌数据中心CPU利用率

- 谷歌将数据中心分为两类：**在线应用与批处理作业**
- **2013年1-3月，**两类数据中心（各两万台服务器）CPU利用率：

在线应用



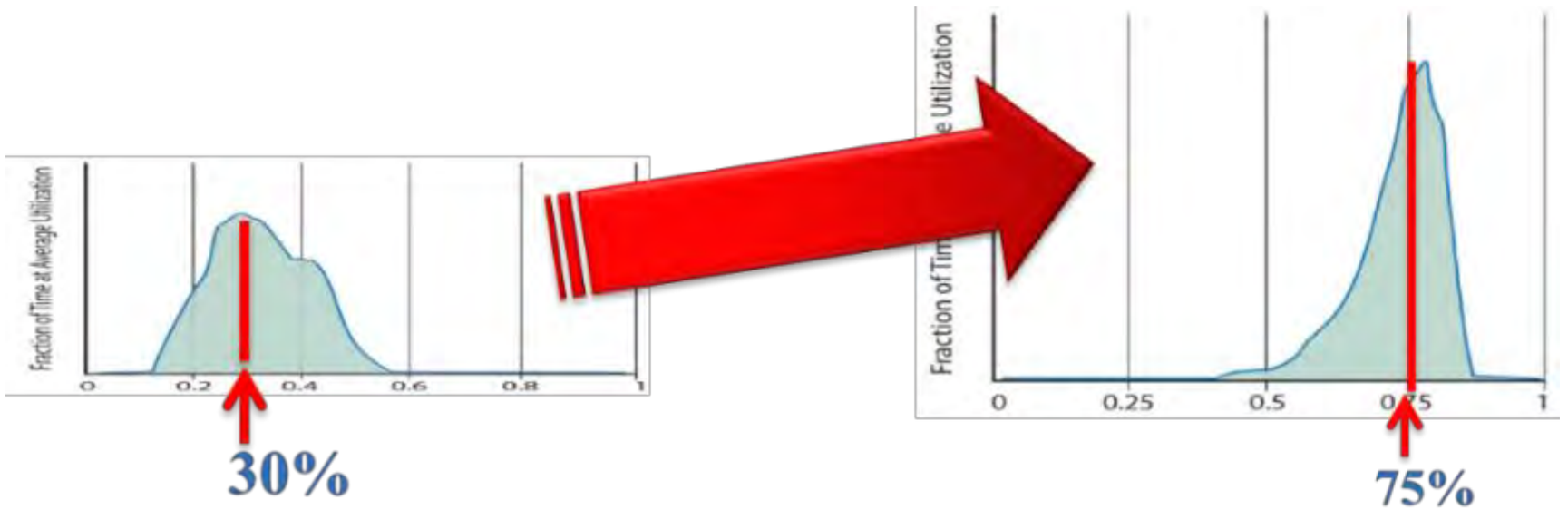
VS.



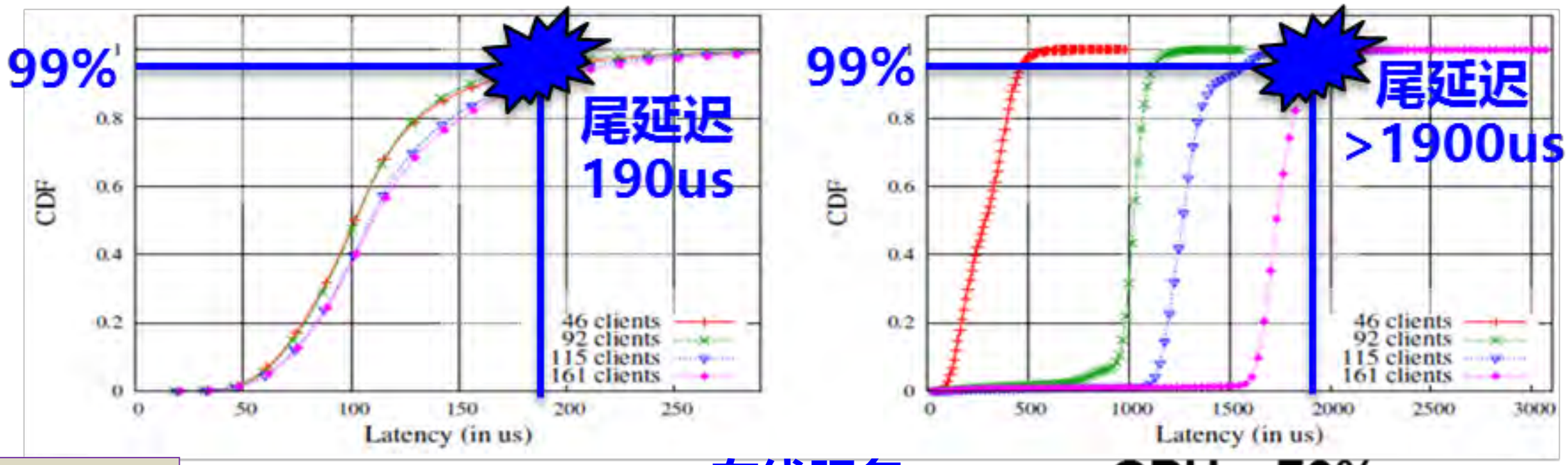
批处理作业

[1] L. Barrosa, J. Clidaras, U. Holzle, The Datacenter as a Computer (2nd Edition), July, 2013.

为何不都提升至75%？



CPU利用率**30%→70%**，响应时间尾延迟**增加10倍**



CPU： 30%　　在线服务 Memcached　　CPU： 70%

R. Kapoor et al. Chronos: Predictable Low Latency for Data Center Applications, SOCC, 2012.

北四环 → 北京南站

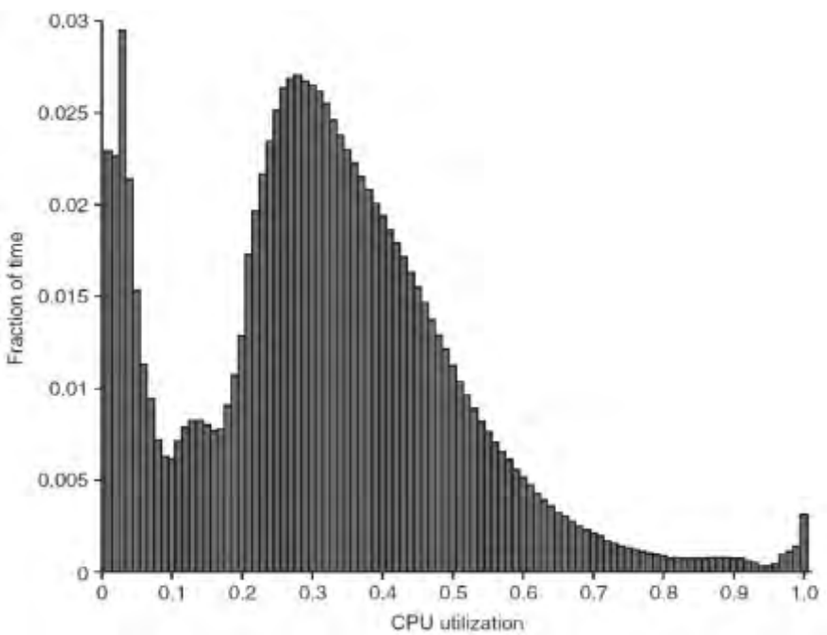
6点，需22分钟

15点，需1小时

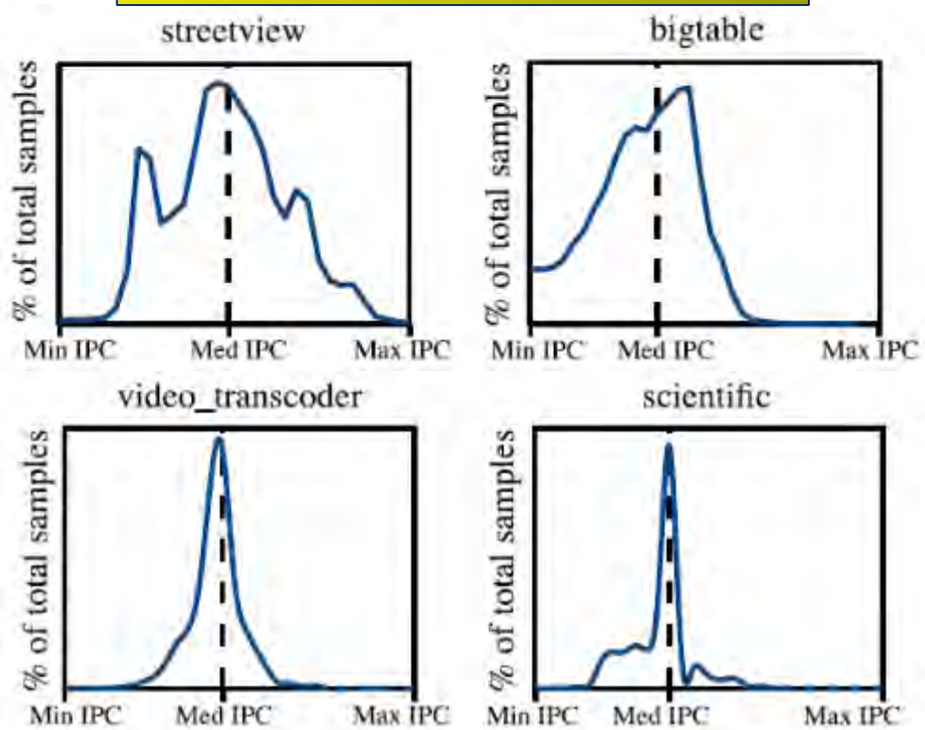
数据中心面临的难题

- 当前数据中心无法同时保障用户体验和高资源利用率，只能二选一

数据中心
资源利用率



响应时间
用户体验



时间就是金钱

- 搜索响应时间
  0.4s→0.9s

- 广告收入下降20%

- 当前云平台以低效率手段来保障用户体验——采用专用分区云+人工优化

Google's Marissa Mayer: Speed wins

Summary: *Marissa Mayer of Google gave a testimonial to speed. Her key insight for the crowd at the Web 2.*

By Dan Farber for Between the Lines | November 9, 2006 -- 15:01 GMT (07:01 PST)

Follow @ZDNet

Marissa Mayer of Google gave a testimonial to speed. Her key insight for the crowd at the Web 2.0 Summit is that "slow and steady doesn't win the race." Speed is a huge component and big market driver of Web 2.0, she said.

In testing out the user interface for Google search, Mayer said that with more results for a query, users were spending less time on the site. It turned out that the cause wasn't just the paradox of choice--paralyzed by too many choices--but the fact that a page with 10 results was half a second faster that the page with 30 results. So, Google set about making the page with more results faster, and the rest is history.

全软件栈优化

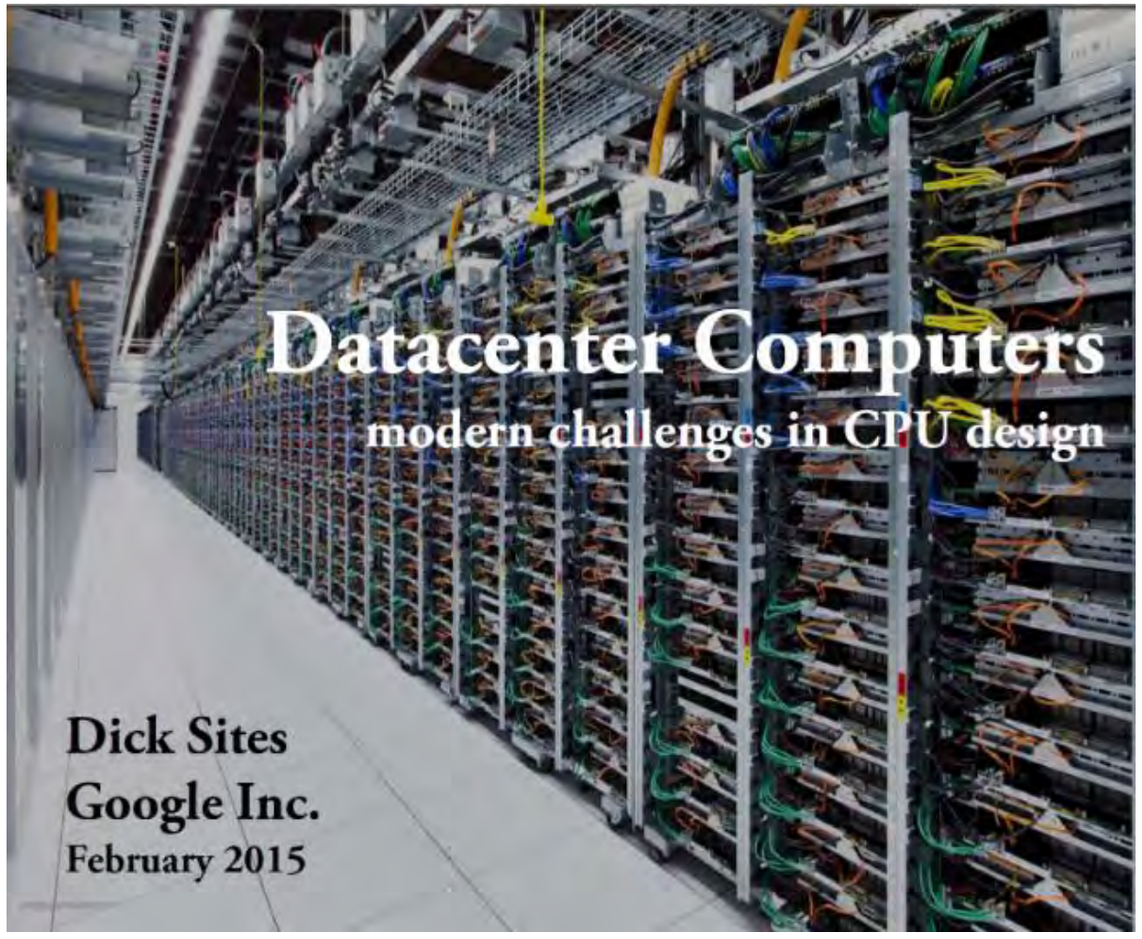
- 过去十年谷歌对数据中心整个软件栈进行优化，从底层虚拟化、操作系统，到上层分布式架构

Borg,
Linux Container,
Cgroups,
Backup Requests,
Priority,
Sync-back-tasks,
……

[1] J. Dean, L. Barroso, “The tail at scale”, Communication of the ACM, Feb. 2013.
[2] J. Dean, “Achieving Rapid Response Times in Large Online Services”, talk at Berkeley, 2012.
[3] Abhishek Verma et al., Large-scale cluster management at Google with Borg, EuroSys, 2015.

亟需硬件支持

- 2015年，美国工程院院士、谷歌数据中心专家Dick Sites在多个学术报告中指出数据中心服务器面临的新挑战——包括如何消除竞争、实现隔离，亟需体系结构创新



Modern challenges in CPU design

- Isolating programs from each other on a shared server is hard
- As an industry, we do it poorly
  - Shared CPU scheduling
  - Shared caches
  - Shared network links
  - Shared disks

- More hardware support needed
- More innovation needed

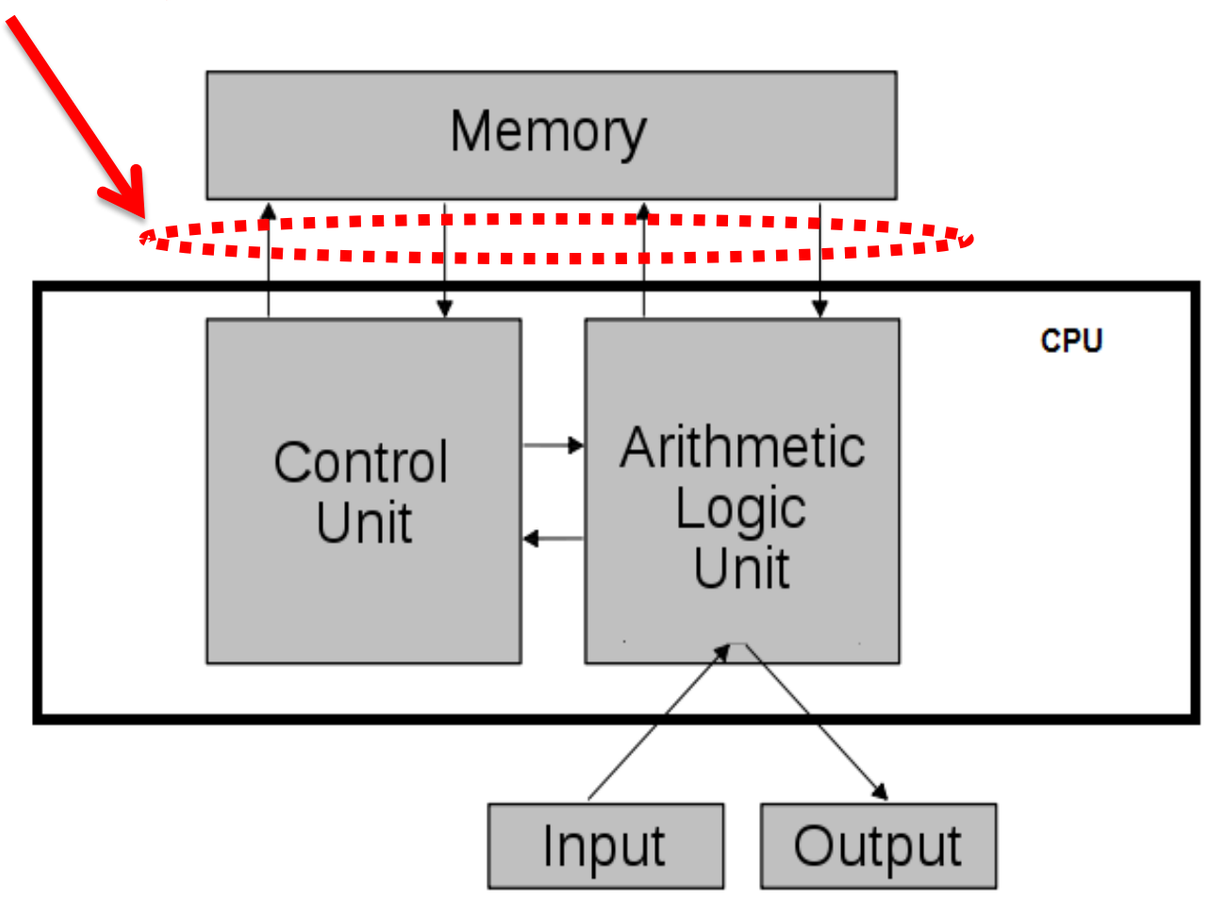
冯诺依曼瓶颈

- 一台冯诺依曼计算机包含三部分：CPU、存储以及连接CPU与存储用于读写的管道（tube）。我建议将该管道称为“冯诺依曼瓶颈”。

— John Backus，1978 Turing Award Lecture

冯诺依曼瓶颈

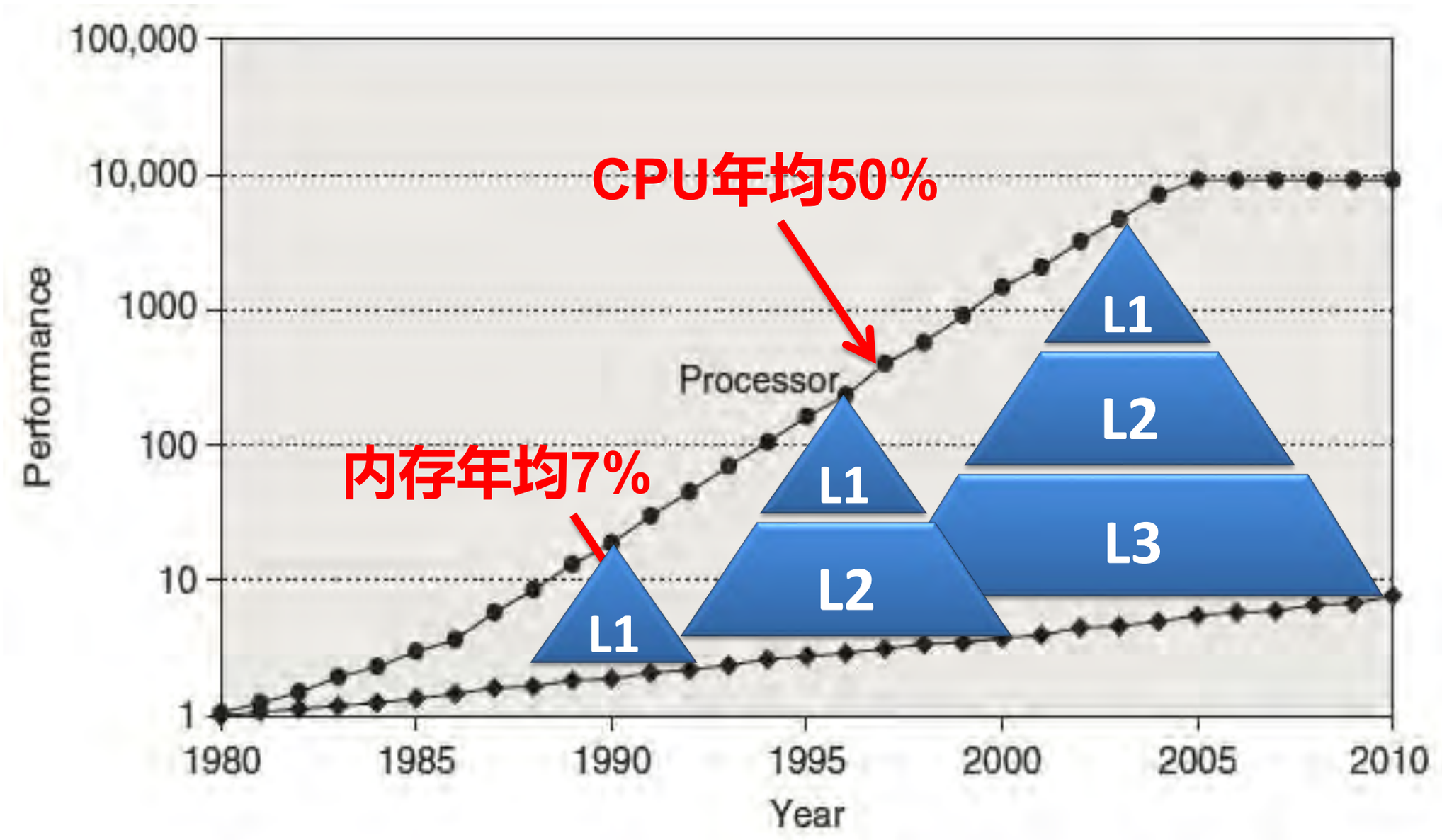
John von Neumann



John Backus

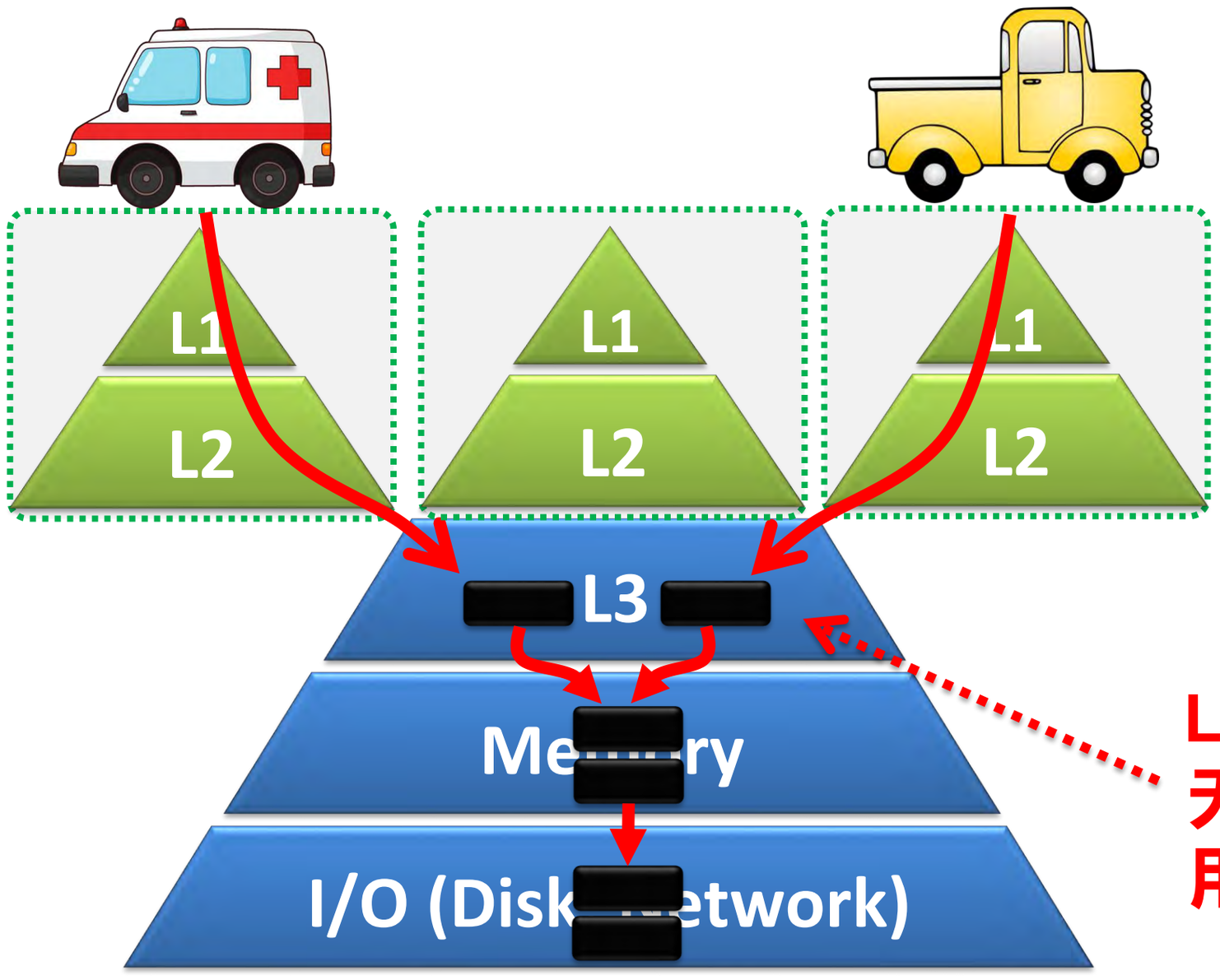
冯诺依曼瓶颈的恶化与应对方法

- 1980年来，CPU和内存提高速度出现不一致，导致**内存墙（Memory Wall）**问题出现
- 体系结构界不断**增加CPU内的高速缓存层次**来应对



当存储层次遇到多核结构

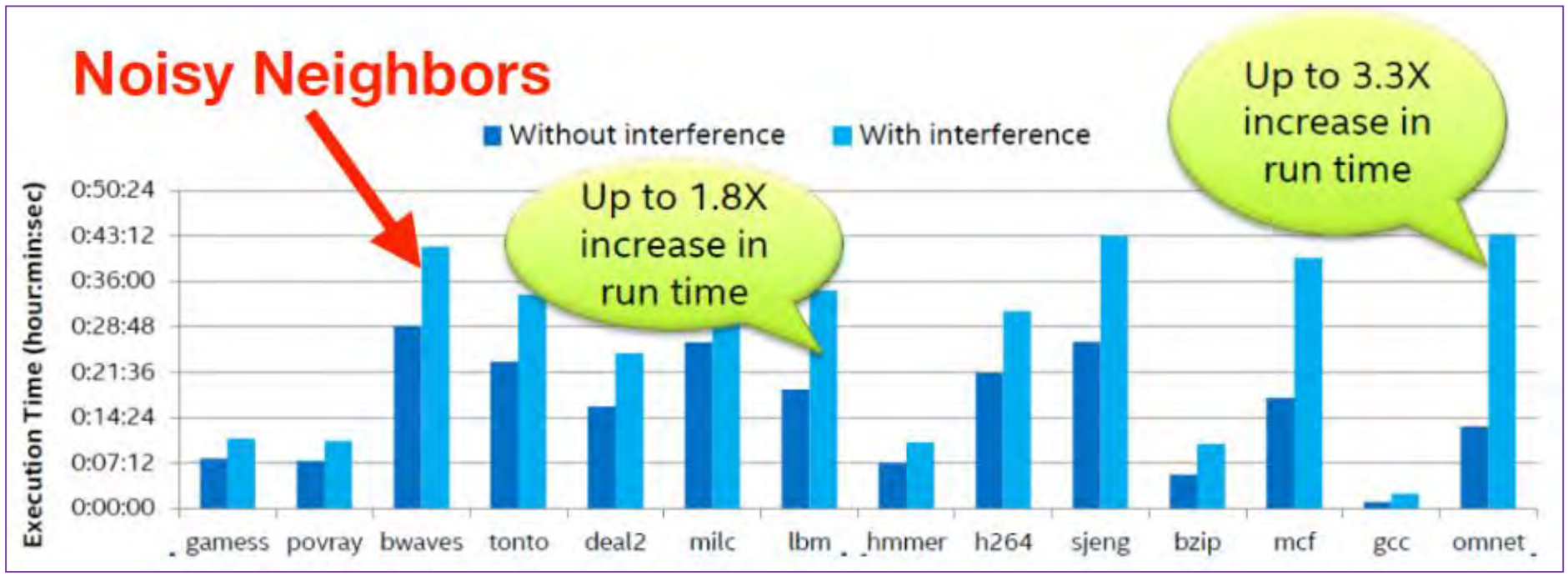
- 多核结构将存储层次分化：核内 + 核外
- 核内L1/L2复制多份，共享核外L3/Mem/IO



L3无序共享
无法判断应
用紧急程度

无序共享导致性能无法得到保障

- 实验：三级缓存共享会造成高达**3倍以上性能下降**，引起云计算“**吵闹的邻居**”问题

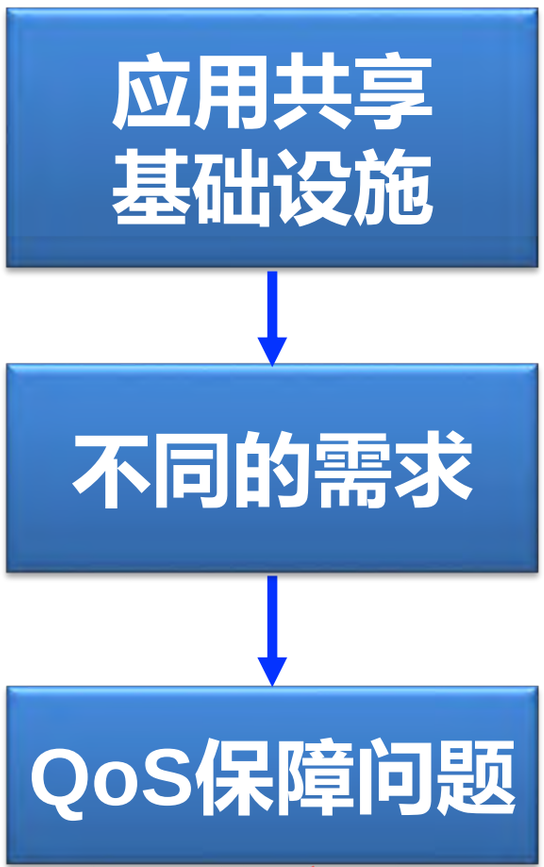


Christine Wang. Intel® Xeon® Processor E5-2600 v3 Product Family Performance & Platform Solutions. 2014.

相似的历史

数据中心时代 2010s

- Search, On-line shopping, Cloud computing, …
- Priority, Throughput, Latency, …
- QoS v.s. Utilization

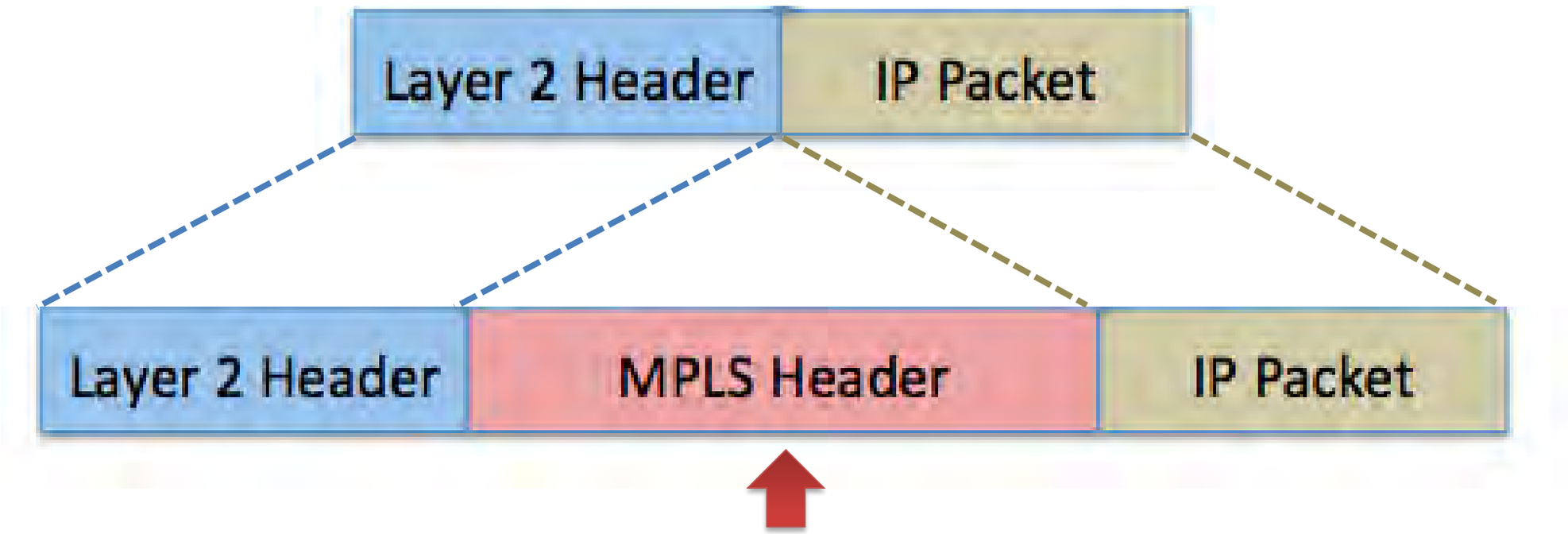


因特网时代 1990s

- HTTP, FTP, VoIP, Stream Media, Game, …
- VoIP, Game, …: Latency-critical
- FTP, VoD,…: Bandwidth-sensitive
- Email: Best Effort
- QoS

标签化网络

- **细粒度对象**：每个网络包增加一个标签（Label）
- **关联语义**：标签值与用户的网络需求关联
- **携带传播**：网络包传播过程中携带标签
- **转发控制**：网络设备基于标签控制网络包转发



多协议标签交换（MPLS）应用于VPN、QoS等

体系结构需要新的抽象接口

21st Century Computer Architecture

A community white paper

May 25, 2012

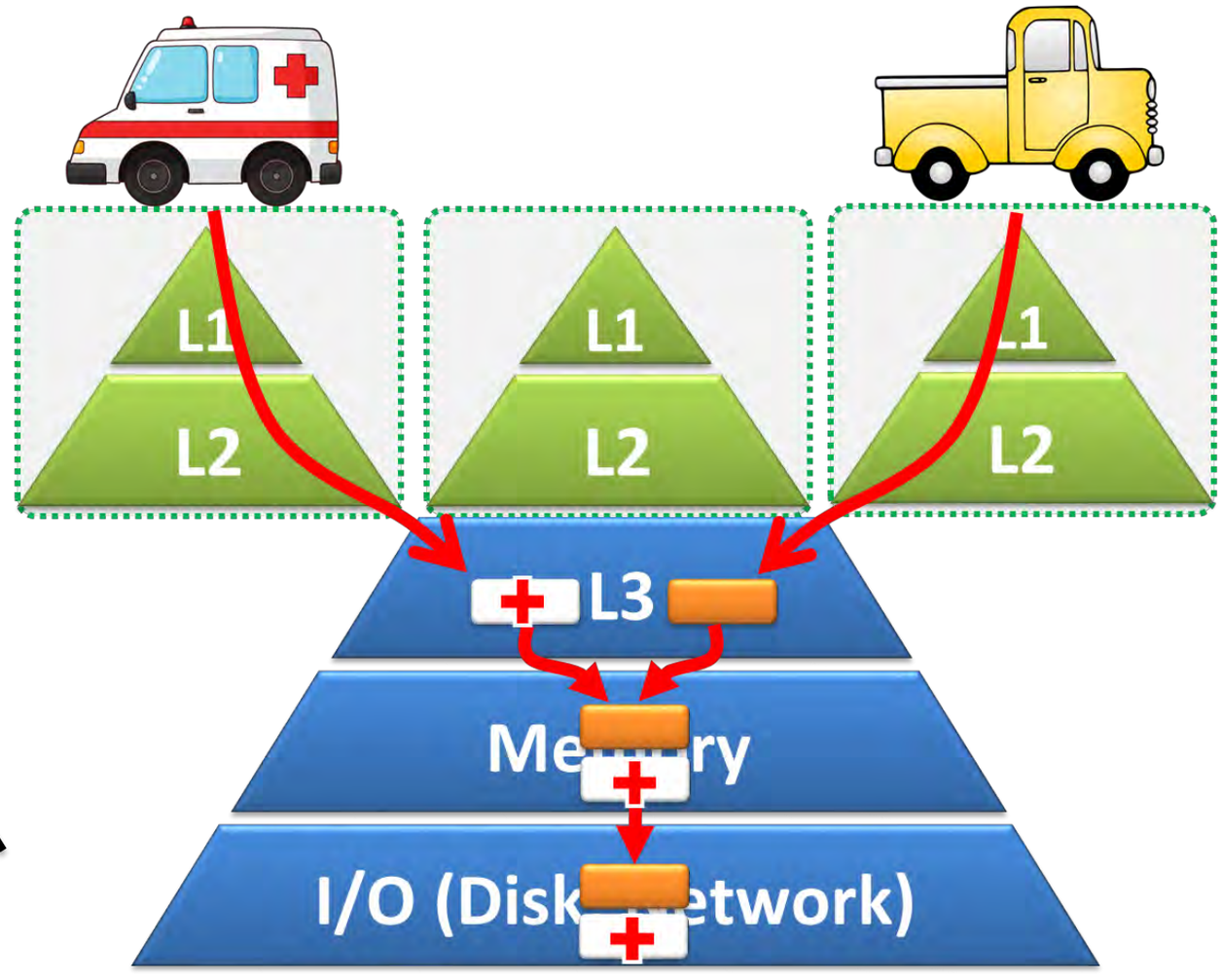
Crosscutting Interfaces

Current computer architectures define a set of interfaces that have evolved slowly for several decades. These interfaces—e.g., the Instruction Set Architecture and virtual memory—were defined when memory was at a premium, power was abundant, software infrastructures were limited, and there was little concern for security. Having stable interfaces has helped foster decades of evolutionary architectural innovations. We are now, however, at a technology crossroads, and these stable interfaces are a hindrance to many of the innovations discussed in this document.

Better Interfaces for High-Level Information. Current ISAs fail to provide an efficient means of capturing software-intent or conveying critical high-level information to the hardware. For example, they have no way of specifying when a program requires energy efficiency, robust security, or a desired Quality of Service (QoS) level. Instead, current hardware must try to glean some of this information on its own—such as instruction-level parallelism or repeated branch outcome sequences—at great energy expense. New, higher-level interfaces are needed to encapsulate and convey programmer and compiler knowledge to the hardware, resulting in major efficiency gains and valuable new functionality.

New, high-level interfaces are required to convey programmer and compiler knowledge to the hardware. **（需要新的高层抽象接口将程序员、编译器的信息传递到底层硬件）**

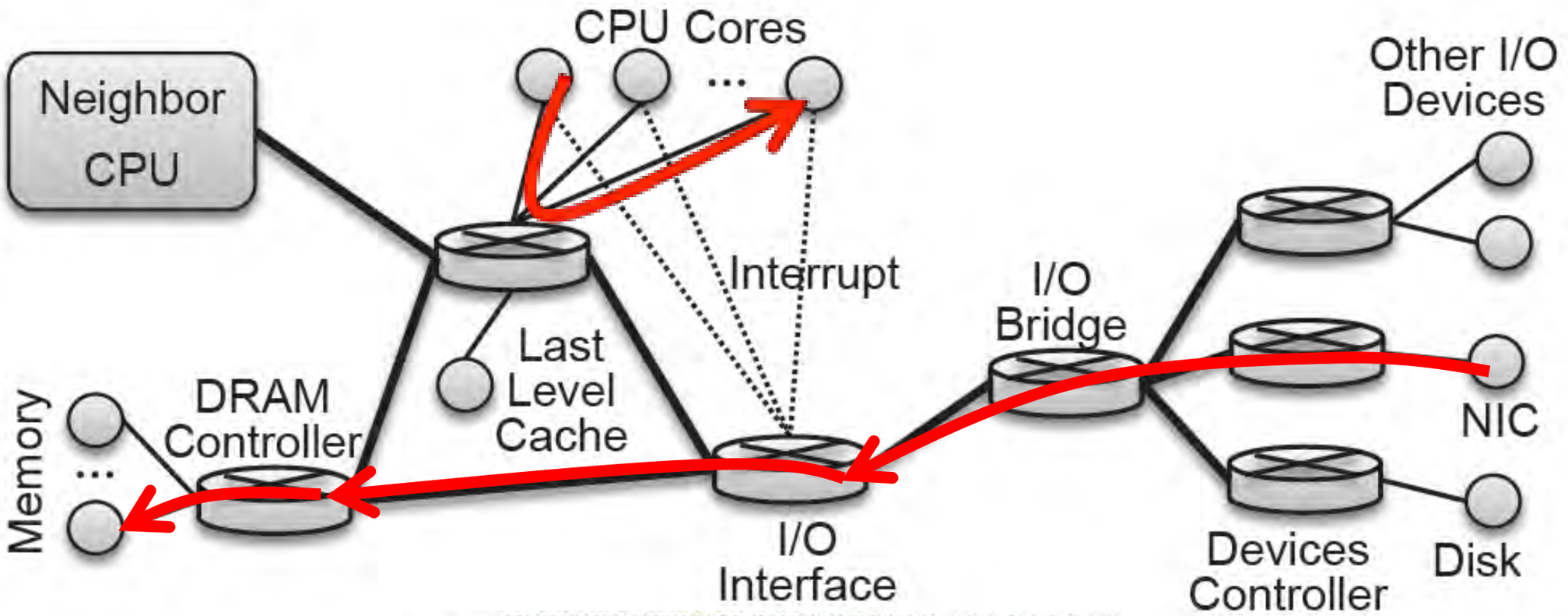
21st Century Computer Architecture

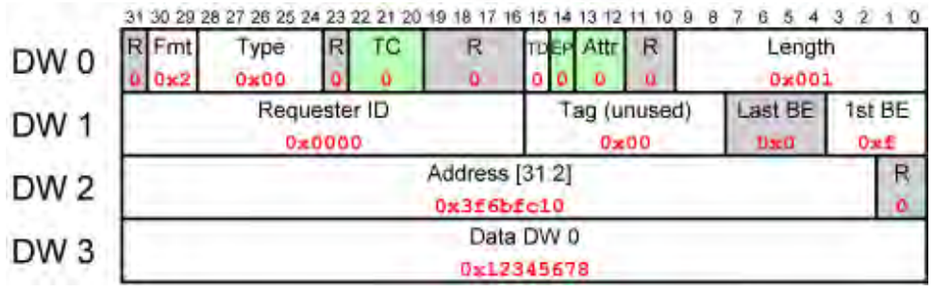


体系结构标签化？

体系结构标签化？可行！

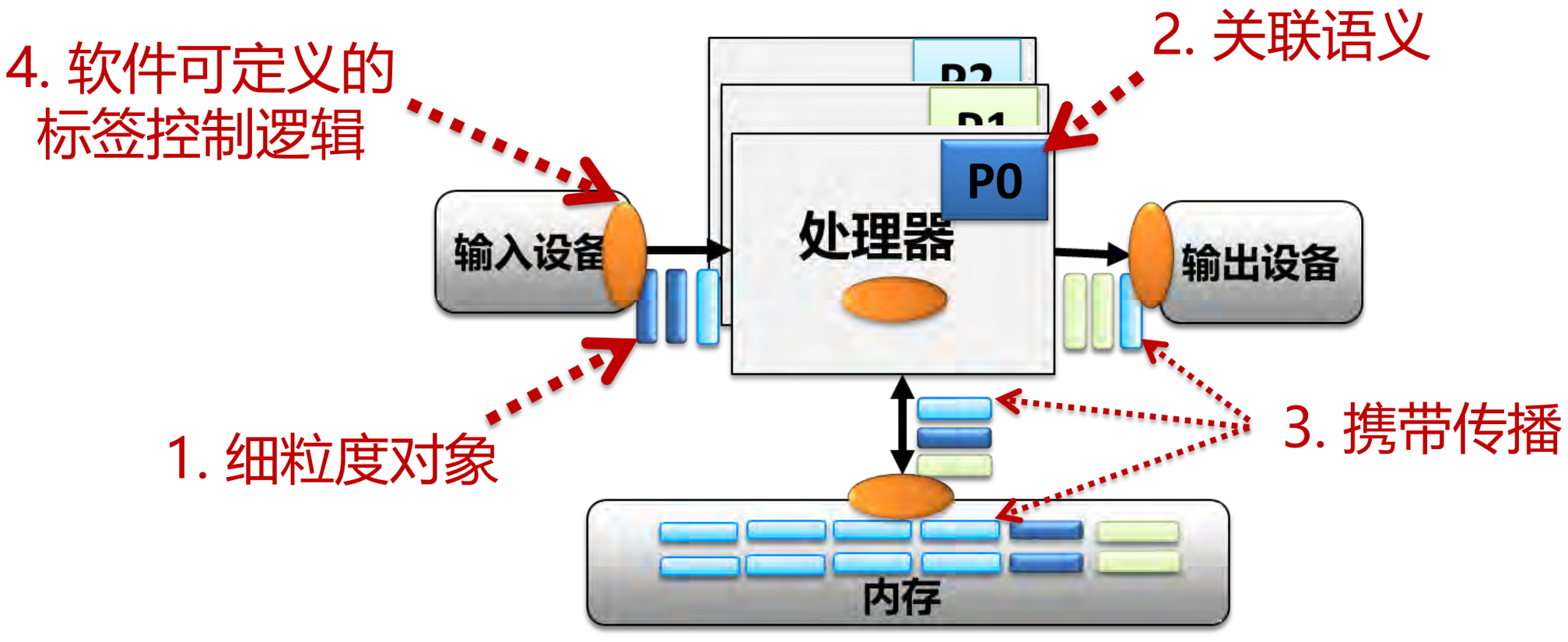
- **计算机即网络（Computer as a Network），**
  计算机本质上就是一个网络，内部的不同部件通过包进行通信——PCIe包、QPI包、NoC包，……





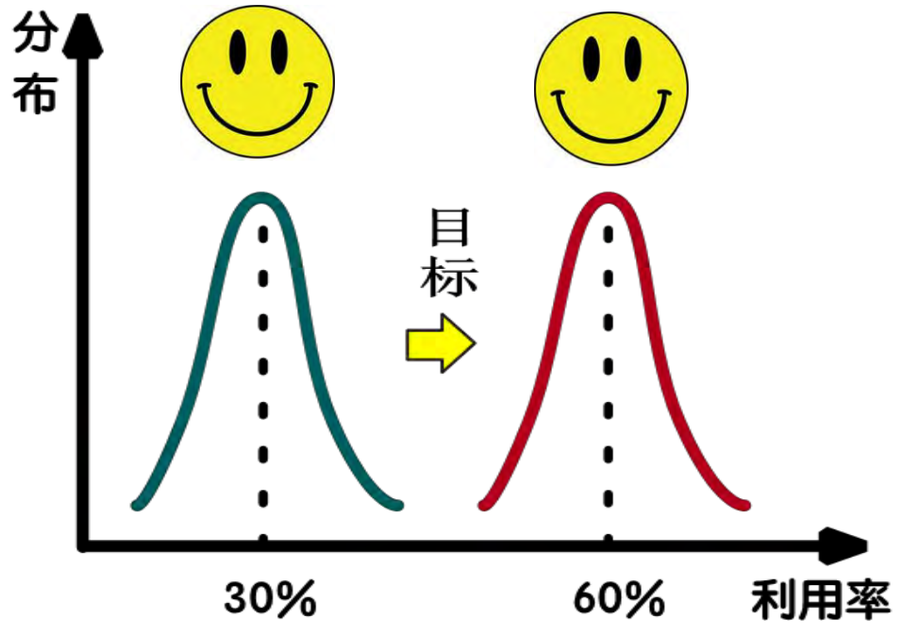
标签化冯诺依曼结构（标冯结构）

- **Labeled von Neumann Architecture（LvNA）**
- **细粒度对象：**每个内部请求（如访存请求）增加一个标签
- **关联语义**：标签值与上层VM/进程/线程/变量等关联
- **携带传播**：标签在计算机访问各个存储层次过程中全程携带
- **软件定义的标签控制逻辑**：根据用户需求对标签控制逻辑编程，实现基于标签的请求区分处理（Differentiated Service）



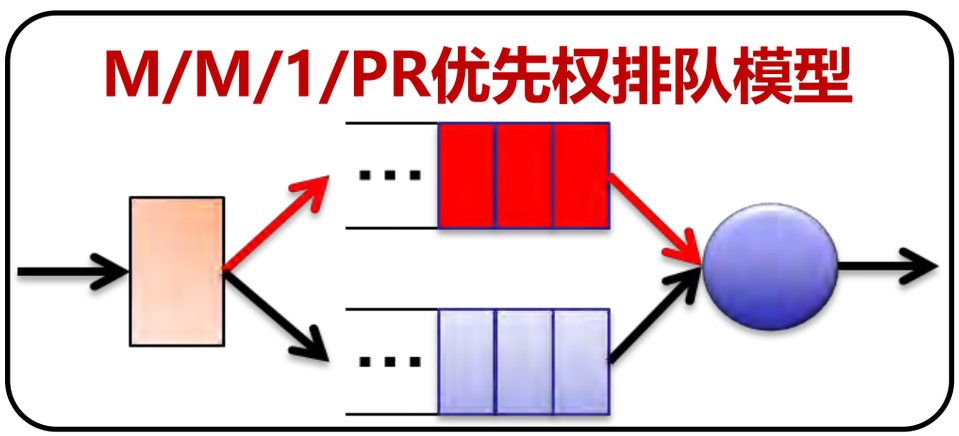
标冯结构的目标与理论分析

- 在共享云环境中，保障在线应用用户体验的前提下，将CPU利用率从30%提高到60%以上



- 优先权排队模型表明，提高系统负载，不会增加高优先级请求平均等待时间

- 标冯结构通过**标签化机制实现了优先权队列**



$$\overline{W_1} = \frac{\overline{R}}{1-\rho_1}$$

$$\overline{W_2} = \frac{\overline{R} + \rho_1 \overline{W_1}}{1-\rho_1-\rho_2}$$

标冯结构的开放问题

- **理论**：标冯结构对传统的RAM、 PRAM、 LogP等计算模型有何影响？是否需要发展新的计算模型？
- **硬件/体系结构**：如何在现有计算机体系结构上实现标冯结构？CPU、内存、存储、网络等硬件部件如何支持？
- **编程模型与编译**：用户需求如何表达，并通过标签传递到硬件？编译器需要做哪些修改以支持标冯结构？
- **操作系统/Hypervisor**：如何将标签与VM、容器、进程、线程关联？如何为可编程的标签控制逻辑定义抽象接口？如何与控制逻辑交互？
- **分布式管理**：如何在分布式环境下实现关联语义与携带传播？如何利用标签化机制管理分布式资源？
- **度量评测**：如何利用标签机制进行对共享资源的度量、审计、追溯？

获科技部“十三五”重点研发计划支持

国家重点研发计划
“云计算与大数据”重点专项

软件定义云计算理论与方法

项目负责人：徐志伟

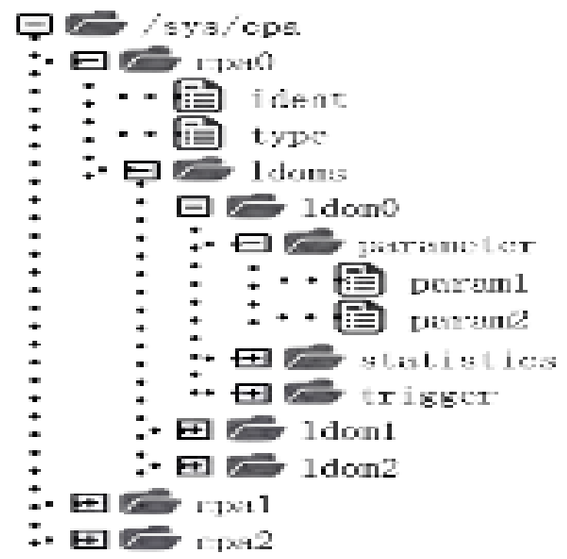
牵头单位： 中科院计算所

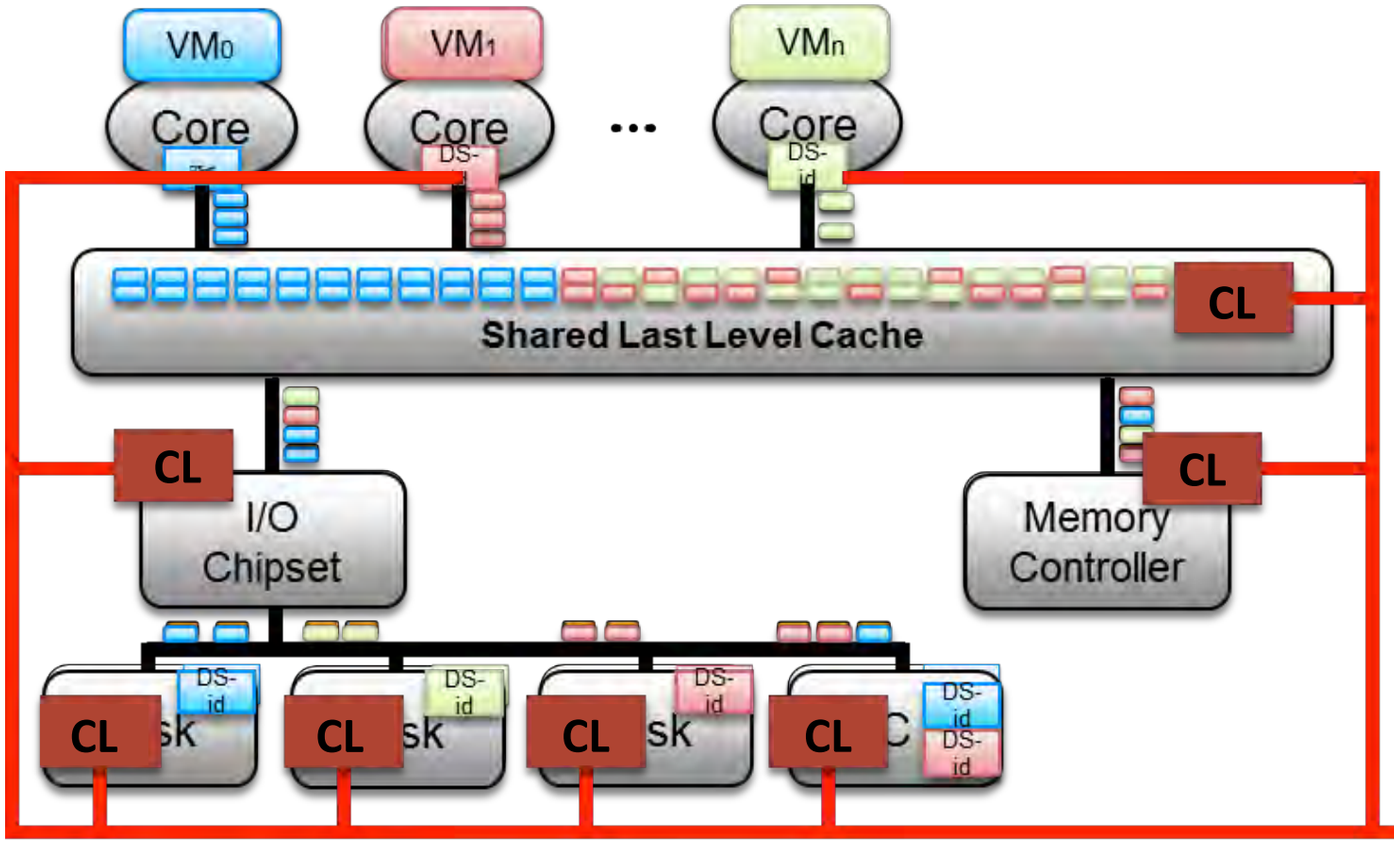
参加单位： 清华大学
天津大学
大连理工大学
中国计量科学研究院
中科院深圳先进院
华中科技大学
工信部电子四院

硬件探索初步结果

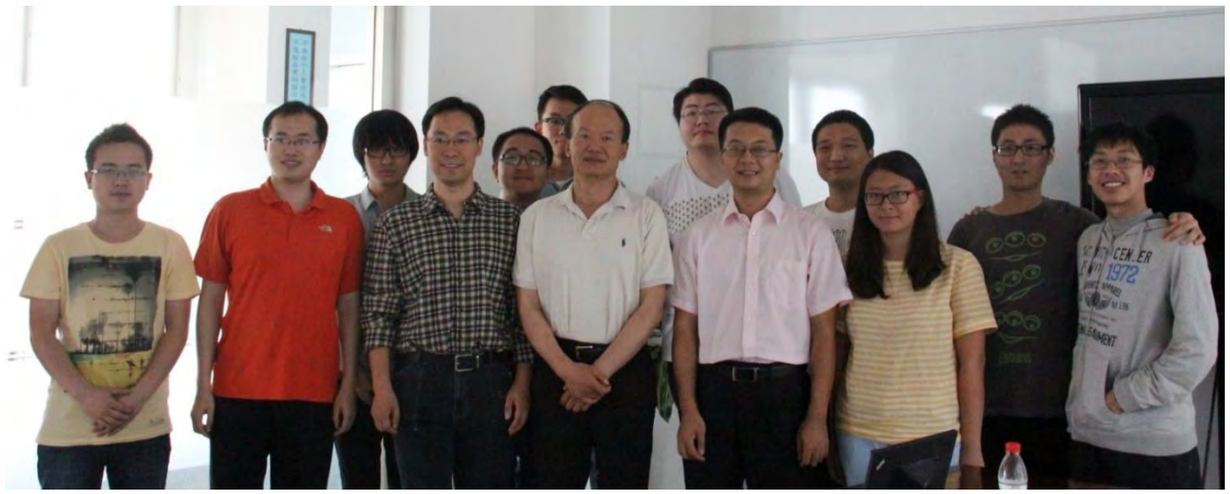
Programmable **A**rchitecture for **R**esourcing-on-**D**emand

PARD

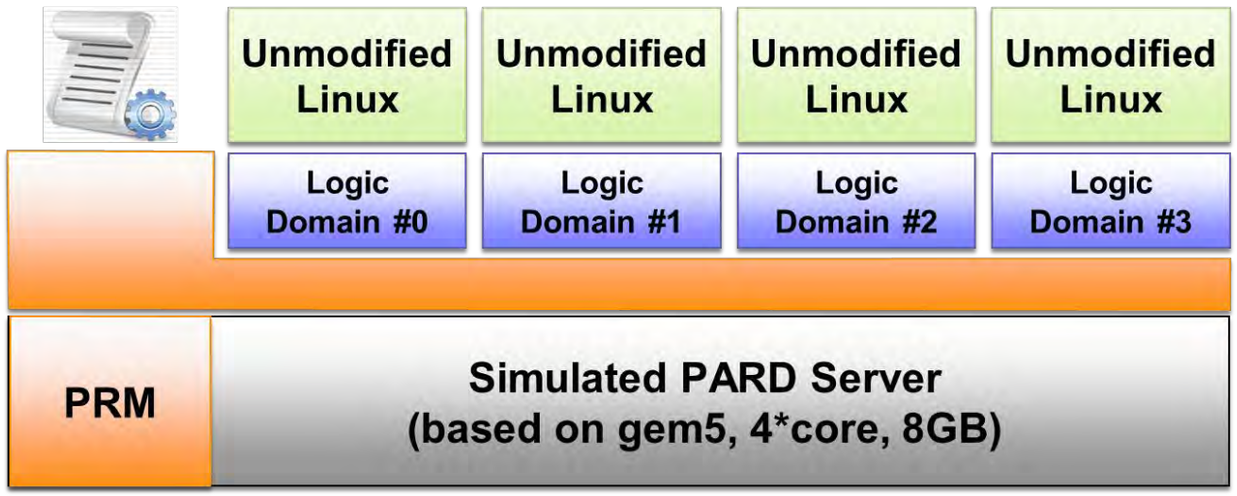




Centralized PRM



历时三年，完成两个原型系统验证



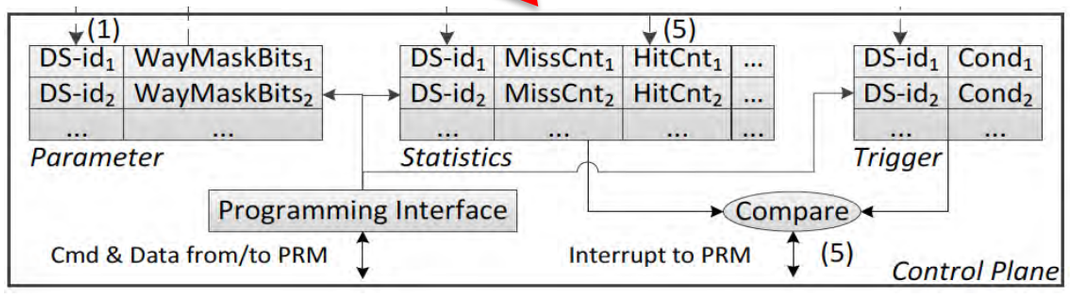
***已开源：http://github.com/fsg-ict/PARD-gem5**

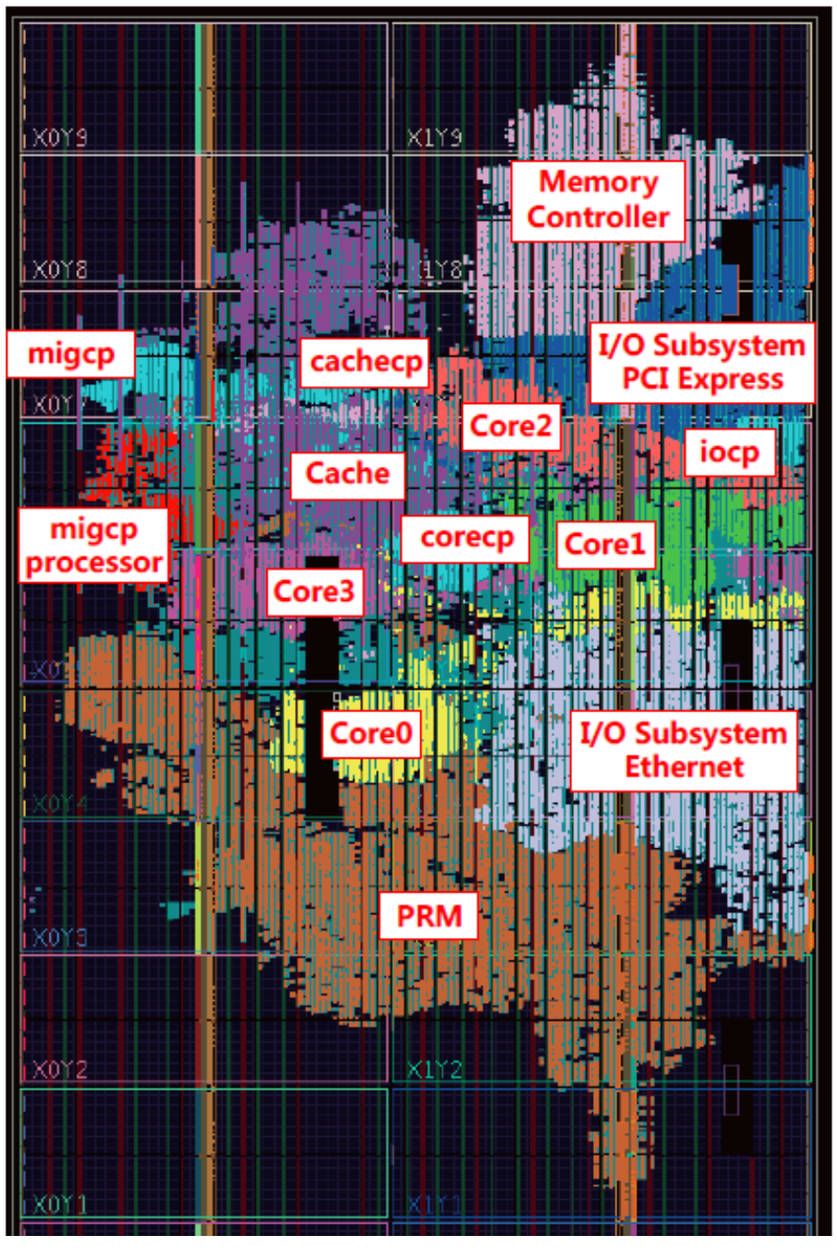
基于FPGA的PARD原型系统

- 4核/16路共享2级缓存/内存控制器/千兆网/PCIe
- **实现了标冯结构**
  - 所有内部请求增加一个标签
  - 标签值与VM关联
  - 标签在所有数据通路传播
  - Core/Cache/Memory/IO增加

可编程标签控制逻辑

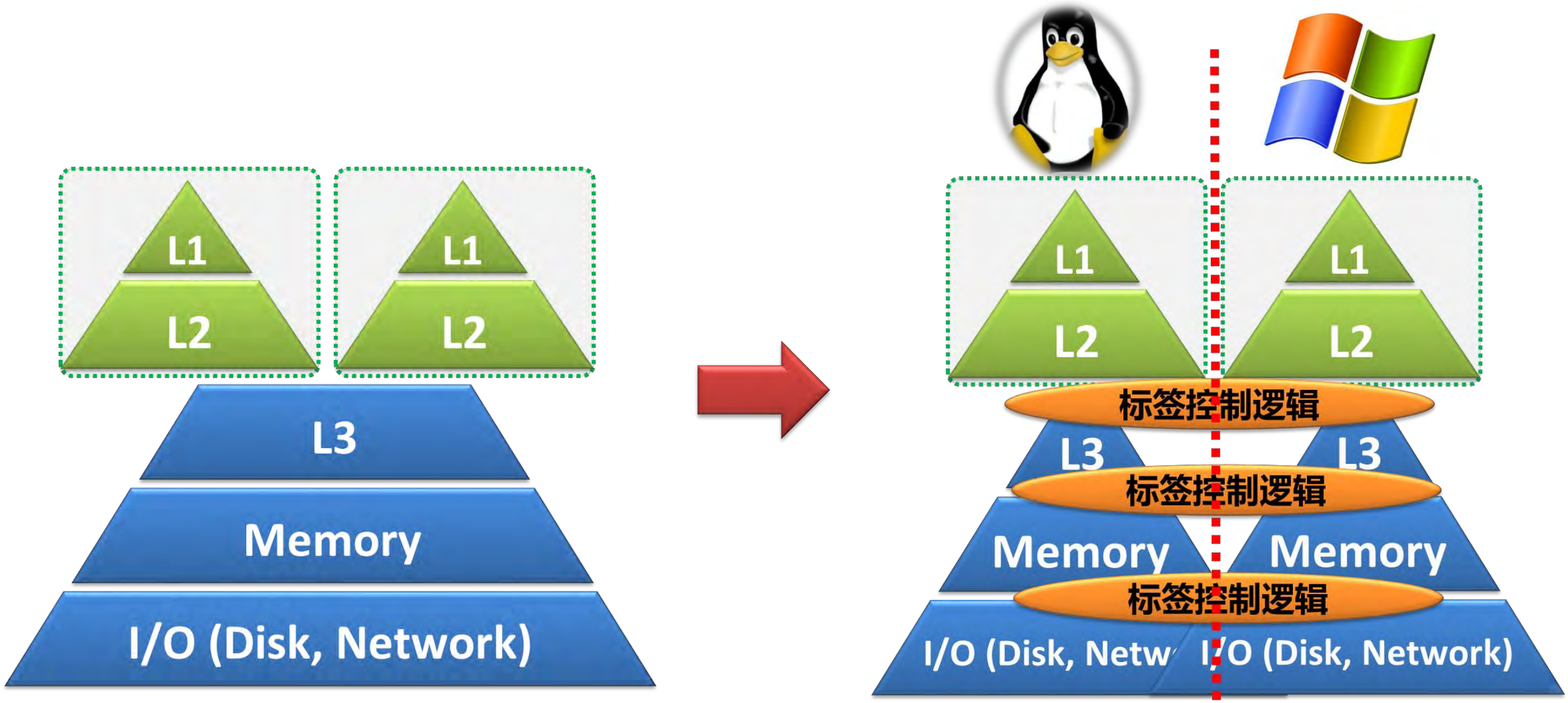
标签控制逻辑模板





示例1：标签控制逻辑加入地址映射

- 不需要软件Hypervisor，利用硬件标签将计算机直接划分为多个相互独立的子机器（称为**分区**），**每个分区运行独立的操作系统**



划分为4个分区，启动4个独立OS

分区1

```
root@prm_core_bd:~# sh kszh 1
Download required files from server ...
Connecting to 192.168.1.1 (192.168.1.1:80)
u-boot-s.bin         100% |*******************************************************************************|   217k  0:00:00 ETA
Connecting to 192.168.1.1 (192.168.1.1:80)
314.ub               100% |*******************************************************************************|  5225k  0:00:00 ETA
Connecting to 192.168.1.1 (192.168.1.1:80)
system-mv-eth.dtb    100% |*******************************************************************************|
Configure KLoader for logic domain ...
copying uboot using CDMA ...
1+1 records in
1+1 records out
copying kernel image using CDMA ...
40+1 records in
40+1 records out
copying device tree file using CDMA ...
0+1 records in
0+1 records out
startup ldom0 ...
Run bootm 0x84000000 - 0x90000000 in uboot to startup system
root@prm_core_bd:~# ping 192.168.1.124
PING 192.168.1.124 (192.168.1.124): 56 data bytes
64 bytes from 192.168.1.124: seq=0 ttl=64 time=5.598 ms
64 bytes from 192.168.1.124: seq=1 ttl=64 time=2.506 ms
64 bytes from 192.168.1.124: seq=2 ttl=64 time=2.578 ms
64 bytes from 192.168.1.124: seq=3 ttl=64 time=2.534 ms
64 bytes from 192.168.1.124: seq=4 ttl=64 time=2.502 ms
```

分区3

```
Creating /dev/flash/* device nodes
random: dd urandom read with 1 bits of entropy available
starting Busybox inet Daemon: inetd... done.
update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge (continuing)
 Removing any system startup links for run-postinsts ...
  /etc/rcS.d/S99run-postinsts
INIT: Entering runlevel: 5
Configuring network interfaces... net eth0: Promiscuous mode disabled.
net eth0: Promiscuous mode disabled.
net eth0: Promiscuous mode disabled.
udhcpc (v1.22.1) started
Sending discover...
libphy: 48000000:01 - Link is Up - 1000/Full
Sending discover...
Sending select for 192.168.1.124...
Lease of 192.168.1.124 obtained, lease time 600
/etc/udhcpc.d/50default: Adding DNS 8.8.8.8
done.

Built with PetaLinux v2014.4 (Yocto 1.7) fsg_mbcore_bd /dev/ttyUL0
fsg_mbcore_bd login: root
Password:
login[313]: root login on 'ttyUL0'
root@fsg_mbcore_bd:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0A:35:00:A2:01
          inet addr:192.168.1.124  Bcast:0.0.0.0  Mask:255.255.255.0
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2408 (2.3 KiB)  TX bytes:1172 (1.1 KiB)

root@fsg_mbcore_bd:~#
```

Linux-3.14.2

分区2

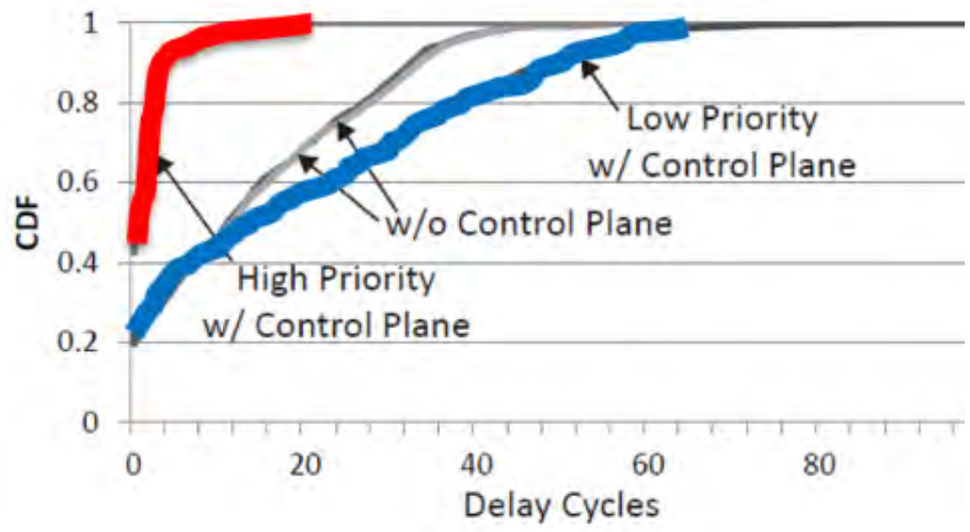
```
INIT: Entering runlevel: 5
Configuring network interfaces... net eth0: Promiscuous mode disabled.
net eth0: Promiscuous mode disabled.
net eth0: Promiscuous mode disabled.
udhcpc (v1.22.1) started
Sending discover...
libphy: 48000000:01 - Link is Up - 1000/Full
Sending discover...
Sending select for 192.168.1.125...
Lease of 192.168.1.125 obtained, lease time 600
/etc/udhcpc.d/50default: Adding DNS 8.8.8.8
done.

Built with PetaLinux v2014.4 (Yocto 1.7) fsg_mbcore_bd /dev/ttyUL0
fsg_mbcore_bd login: root
Password:
login[313]: root login on 'ttyUL0'
root@fsg_mbcore_bd:~# wget 192.168.1.1/yzh/test
Connecting to 192.168.1.1 (192.168.1.1:80)
random: nonblocking pool is initialized
test                 100% |*******************************| 65536k  0:00:00 ETA
root@fsg_mbcore_bd:~#
```

分区4

```
root@fsg_mbcore_bd:~# free
             total         used         free       shared      buffers
Mem:        514044        15276       498768            0            0
-/+ buffers:              15276       498768
Swap:            0            0            0
root@fsg_mbcore_bd:~# uname -a
Linux fsg_mbcore_bd 3.14.2 #3 Tue Sep 8 09:54:18 CST 2015 microblaze GNU/Linux
root@fsg_mbcore_bd:~#
```

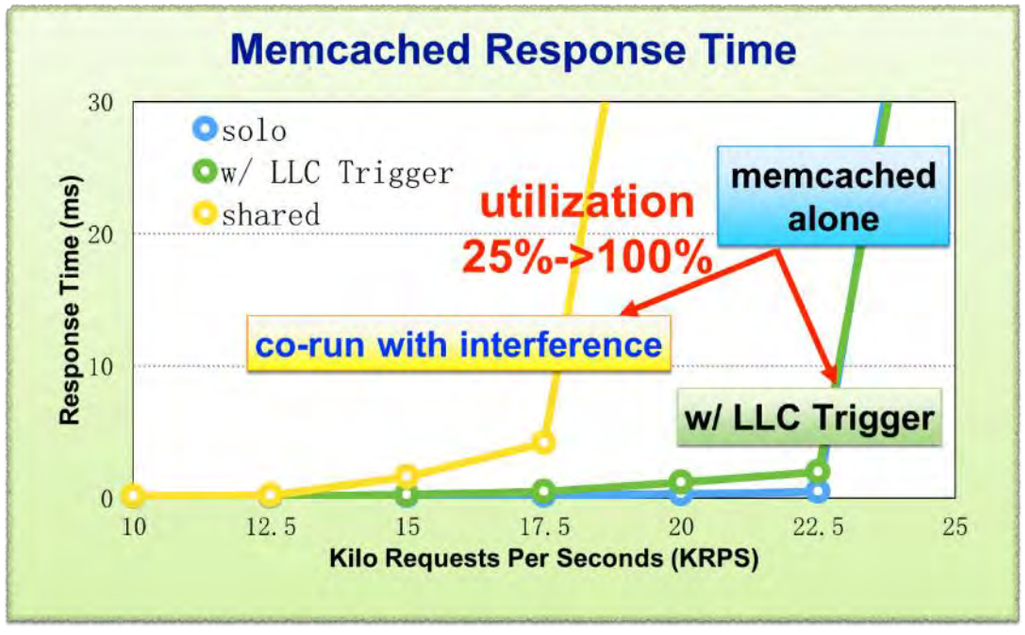
示例2：控制逻辑加入优先权控制

- 50%的带宽利用率情况下，内存控制器请求排队时间
  - **高优先级降低5.6倍**
  - 低优先级增加33%



可实现保障用户体验前提下提高资源利用率

- CPU利用率提升4倍
- 在线服务尾延迟增长<20%



初步影响力，还需更深入研究

- 2015年发表于体系结构会议ASPLOS，与华为联合申请了专利群
- 受邀参加“数据中心架构”德国达堡论坛（ Dagstuhl Seminar ）
- 作为华为全球合作代表成果之一入选华为2015年报

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

2015年11月，华为在欧洲企业社会责任协会(CSR Europe)企业2020峰会讲述华为的合作创新平台——华为创新研究计划(HIRP)。HIRP已有一百家左右的学术机构、逾千名学者参与，并资助研究性数千名，2015年HIRP新资助一百多个研究项目，进一步加大基础研究和技术创新。

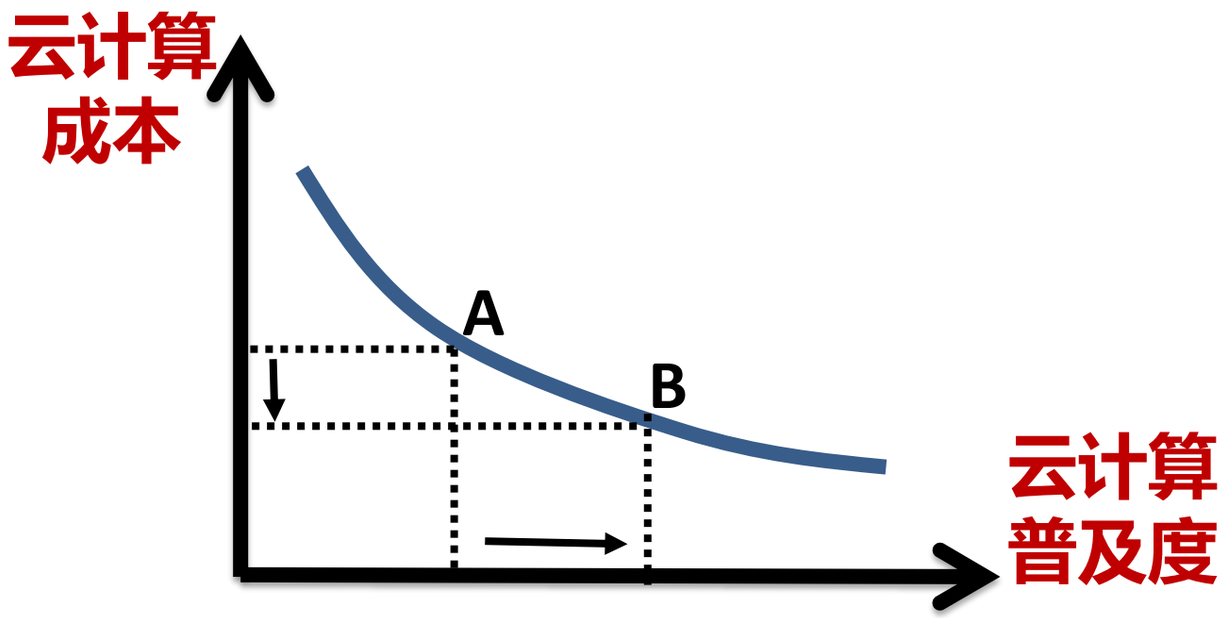
域长期领先的市场和技术地位。针对未来基站密集组网场景，在微基站智慧路灯方面也进行了众包合作建网商业模式的有益探索。

- 与研究机构在国际上首次创造性地提出了数据中心领域支持全硬件虚拟化与硬件资源按需分配的服务器新架构(PARD)，从硬件层次为解决数据中心无法同时达到高利用率与高服务质量的难题提供关键支持，得到学术界的高度评价。
- 与全球多所知名高校、技术创新型公司、开源组织合作，在分布式存储、SCM存储系统、云计算平台、大数据、人工智能与知识库、高清视频等领域开展广泛合作，推动技术创新。

- Collaborated with research institutes to propose a novel server architecture that enables a Programmable Architecture for Resourcing on-Demand (PARD) and full hardware virtualization for data centers. The proposed architecture, which provides critical hardware-level support to address the challenge of achieving both high utilization and high quality of service (QoS) in data centers, won high acclaim from academics.

总结

- 标签化冯诺依曼结构的目标是提高数据中心资源利用率，降低数据中心建设与维护成本
- 有人问：若标冯结构成为主流，数据中心成本降低了，设备商的收益是否会降低？
- **杰文斯效应（Jevons Effect）**：技术进步会增加技术的消费量，达到一定规模必然会盈利
- 我们相信通过努力降低云计算成本，必然会进一步促进云计算快速发展



谢谢！

云岗，生日快乐！