

Challenges of Taobao System Workloads

Coly Li <colyli@gmail.com>

Content

- Outline
- Taobao System Workload
- Characteristic of Taobao System Workload
- Challenges & Opportunaties



Outline

Principles,

- Practical examples
- Problems cared by industries
- No explicit suggestion or solution

Material organization,

- Top 3 system workloads in Taobao
 - Puzzle in each workload
 - Workload characteristic from view of a system software engineer
 - Generalized challenges from but not limited in Taobao
 - Time for us to think/discuss
-
-

Taobao System Workload

Some Real Data

Unofficial Operation Disclosure^[1] (Million RMB)

| | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 |
|---------------------|-------|--------|---------|---------|---------|---------|
| Trading Volume | 4,000 | 99,600 | 200,000 | 400,000 | 600,000 | 800,000 |
| Conversion Rate | 0.02% | 0.40% | 0.85% | 1.00% | 1.20% | 1.50% |
| Gross Profit | -43 | -46 | 89 | 250 | 451 | 751 |
| Gross Profit Margin | | | 43% | 43% | 43% | 43% |

Key Time Mark^[2]

Aug 2008, Expense Balance Receipts

Year 2009, Gross Profit > 0

Sep 2010, Revenue Per Day > 10M RMB

Nov 2010, Peak Network Througput > 300Gbps

Key Number^[2]

80% market share of online retail

50%+ express service traffic

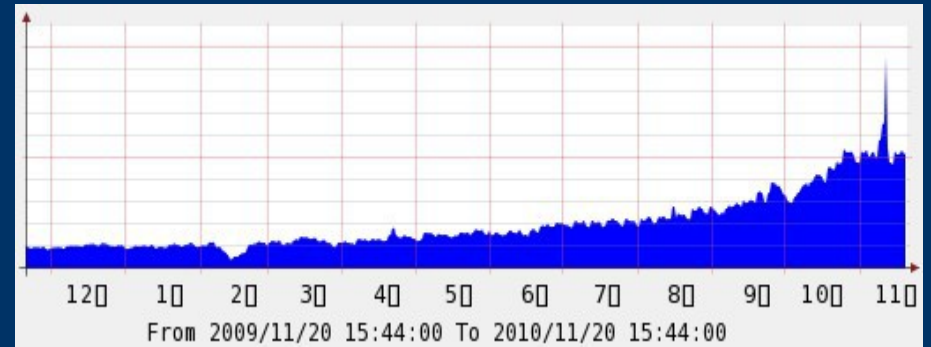
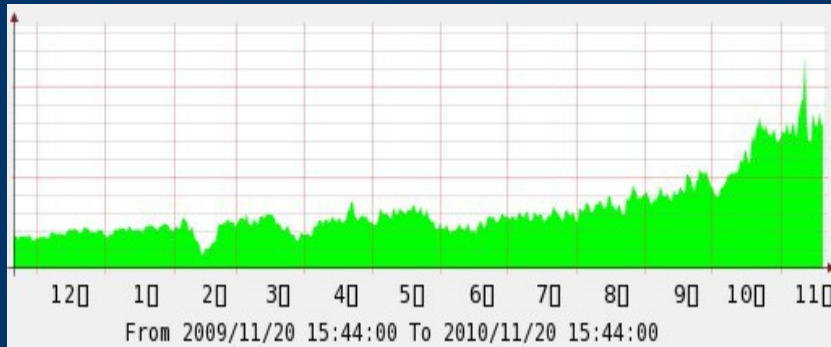
[1] reported by Goldman, Mar 2010

[2] unconfirmed info from internet

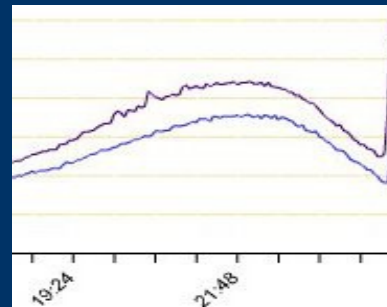
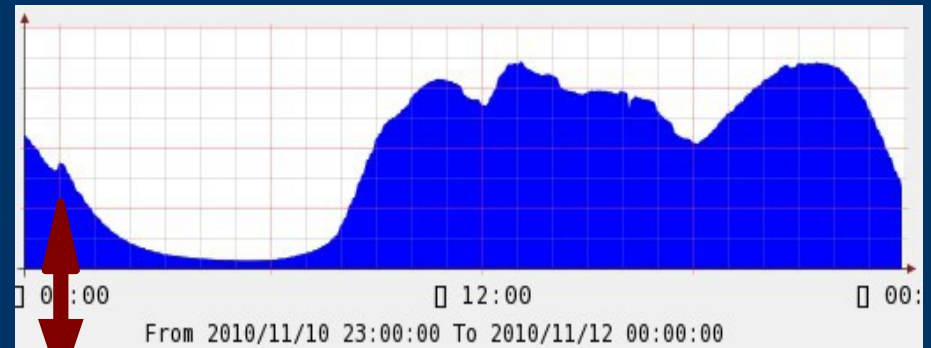
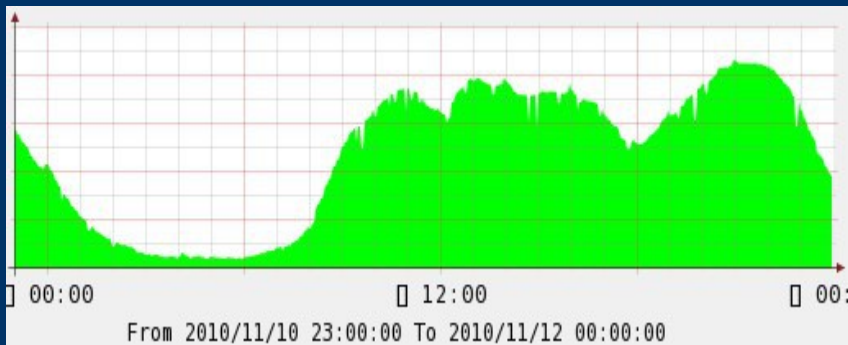
Taobao System Workload

Some Real Data (Cont.)

Taobao Total Network Traffic (2009 Nov ~ 2010 Nov)



Total Network Traffic on 1111 Celebration

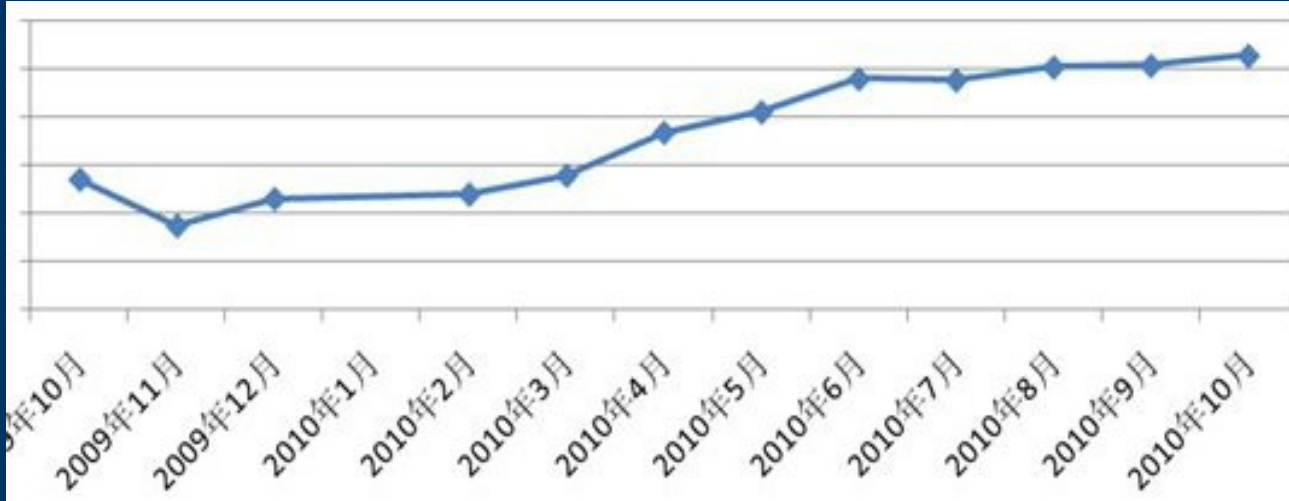


Pictures from Taobao internal traffic monitor systems

Taobao System Workload

Some Real Data (Cont.)

Taobao Servers Usage (upto 2010 Oct)



| | |
|-------------------------------|------|
| Search Engine | ??K+ |
| Content Delivery Network | ??K+ |
| Data Storage | ??K+ |
| Advertize, Goods and Analysis | ??K+ |
| Trade Core | ??K+ |
| R&D | ??K+ |
| ----- | |
| Total | ??K+ |

Picture from Taobao internal monitor system

Taobao System Workload

Top 3 Loads

The top 3 loads of Taobao are,

- Search Engine
- Content Delivery Network
- Storage and DB

They are the most important loads for Taobao,

- Serve most of online user and business operations
- Accommodate core trading data and information
- Consume most of hardware/software/internet resources
- Hot spots for scalability and performance optimization

All workloads are developed and running on Linux operating system.

This slide shares some observation to these workloads. Too much business process will not be touched, all descriptions are abstracted and focus on operating system level.



Taobao System Workload

Search Engine

A typical workload of search engine

Unlike internet search engine (google, baidu, etc), Taobao dose not use web crawler, all information is fetched from database.

Heavy read, locality existing.

- Well structured reversed indexing files
- Second level or third level indexing files
- Restricted read latency requirement
- Most of reading sizes are small
- Non-prefect streaming read

Frequent write, targets with different priority

- To log files
- To internal files for reserved indexing
- To reserved indexing files
- Blocks writing to log files should be served with highest priority
- Dirty pages reclaim is possible happening

Puzzle observed in typical search engine workload

Reading duplication cost,

- Heavy buffer duplicating cost between k/u address space in read(2).
- Development sophisticated user space buffer management code with direct I/O is hard to most of software engineers.
- Mmap based I/O does not update LRU in kernel page cache management code.

Non-priority writes,

- For heavy I/O to/from different files, I/O priority can only be served with fraction of processes or underlying storage devices.
- With a process, I/O to/from different files can not be served with different I/O priorities.

Non-priority page cache allocation & writeback,

- Some files are desired to map more page caches
- Current page cache allocation priorities are devices and state based
- In page cache reclaim, writing back is handled per block device interface, not per inode interface.

Resulted performance & business penalty.

A typical workload of content delivery network

Heavy writing load produced by heavy reading loads,

- CDN machines run optimized squid proxy for static web object caching.
- Cache missing when popular goods updated from main site
 - ◊ Iphone4, ipad, kindle 3, ...
 - ◊ Description info or pictures changed
 - ◊ Well priced goods for seckill
- Cache service new initialized

Heavy reading load almost 7x24,

- Locality existing
- Most of reads are relative small objects, < 500KB
- Machine relaxed from read loads in 3:00 - 5:00 AM
- 98%+ cache hit happened in SSD, 1%- cache hit on SAS has significant cost
 - ◊ 50%+ SAS disk I/O utilization was observed with high hit rate on SSD

CPUs are in idle state in 95%+ time,

- A top(1) output from an online server with ordinary load

```
top - 23:25:32 up 3 days, 7:11, 1 user, load average: 0.44, 0.60, 0.66
Tasks: 118 total,  2 running, 116 sleeping,  0 stopped,  0 zombie
Cpu0  :  3.0%us,  3.3%sy,  0.0%ni, 84.0%id,  6.7%wa,  0.0%hi,  3.0%si,  0.0%st
Cpu1  :  2.3%us,  1.7%sy,  0.0%ni, 86.4%id,  7.0%wa,  0.0%hi,  2.7%si,  0.0%st
Cpu2  :  2.3%us,  2.0%sy,  0.0%ni, 87.0%id,  4.3%wa,  0.0%hi,  4.3%si,  0.0%st
Cpu3  :  2.0%us,  3.3%sy,  0.0%ni, 89.7%id,  2.3%wa,  0.0%hi,  2.7%si,  0.0%st
Cpu4  :  4.0%us,  2.3%sy,  0.0%ni, 87.4%id,  2.6%wa,  0.0%hi,  3.6%si,  0.0%st
Cpu5  :  3.0%us,  1.7%sy,  0.0%ni, 89.7%id,  2.3%wa,  0.0%hi,  3.3%si,  0.0%st
Cpu6  :  2.0%us,  2.0%sy,  0.0%ni, 91.1%id,  2.3%wa,  0.0%hi,  2.6%si,  0.0%st
Cpu7  :  1.3%us,  2.0%sy,  0.0%ni, 92.0%id,  2.0%wa,  0.0%hi,  2.7%si,  0.0%st
Mem: 16466856k total, 16377304k used,  89552k free,  49048k buffers
Swap: 2096476k total,  0k used, 2096476k free, 13260804k cached
```

- A single core can handle 3+ times workload as the above example
- Other 7 cores are (almost) wasted
- Memory is mainly occupied by page caches

Puzzle observed in CDN workload

- Disbalance hardware expense according to the workload
 - ◊ Expense CPUs are wasted
 - ◊ More memory chips are desired
 - ◊ More SSD are desired
 - ◊ Inefficient power supply consuming
- Reading is not always prior then writing
 - ◊ Writing are produced by reading
 - ◊ NO QOS for writing demonded by multiple reading
- Locality is far from perfectly optimized
 - ◊ Hybrid storage with SSD, SAS, SATA disks
 - ◊ Little optimization in user space application level
 - ◊ No optmization for hybrid infrastructure in kernel level

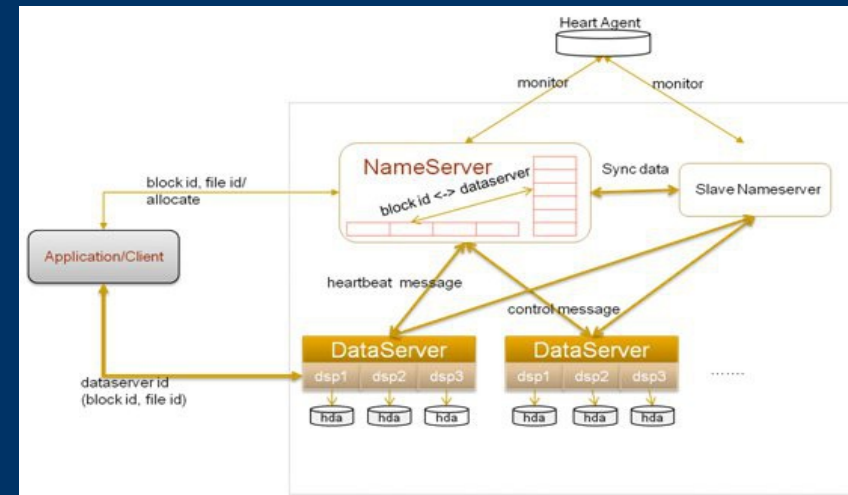
Taobao System Workload

CDN (Cont.)

A typical workload of Data Storage

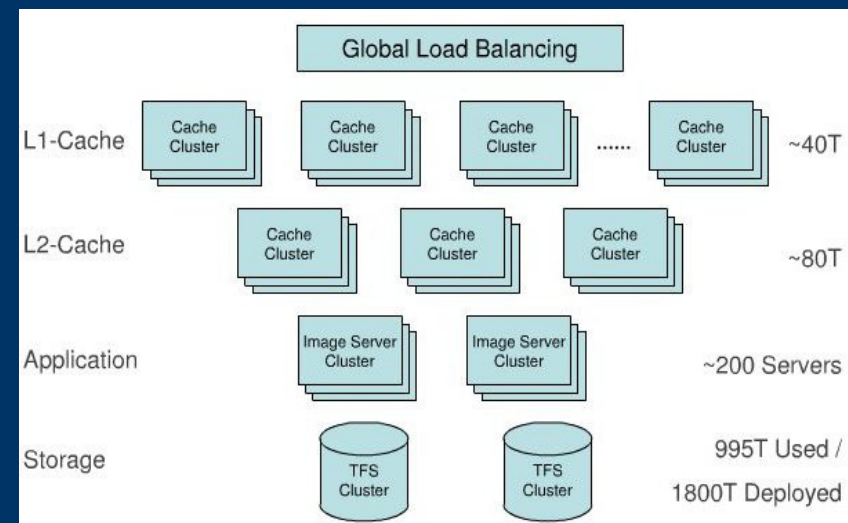
Reasonable data storage scalability is not a bottle neck any more,

- Bunch of solutions from Google papers
 - GoogleFS, Hadoop, Ceph, ...
 - In Taobao, it's TFS (Taobao FS)
- A meta data server with large number of numous data nodes
- The meta data server can be virtualized by a cluster for high throuput and avaiiability



Meta data loads increased while storage getting scaled,

- Meta data server becomes hot spot
- When meta-data-of-the-meta-data gets hot in memory, performance identified between using raw device or file system.
- Most of locality is hit in CDN, final Data request to TFS is almost random.
- A reliable & high performance repo for log I/O is key



Pictures from Taobao Massive Data Storage and CDN, Dr. Wensong Zhang (Taobao)

Puzzle observed in Data Storage workload

Meta data management,

- High performance and high available meta data management is difficult
 - ◊ Every component in the storage stack might be bottle neck
 - ◊ Underlying storage media is untrustable (silent error)
 - ◊ Underlying storage media is unreliable (hardware/link failure)
 - ◊ Underlying storage media topology awareness (commercial secret)
 - ◊ Redundent & application level checksum introduce performance penalty
- Meta data consistency checking,
 - ◊ Heavy random reading
 - ◊ Heavy memory consuming
 - ◊ Unbelievable seeking cost on hard disk (2-3 hours for 3TB ext3)
 - ◊ Offline checking is unacceptable
 - ◊ Backup & restore (raw device vs. file system)
- System is not ready for extremely high performance SSD
 - ◊ 100 million IOPS SSD can be found in market (640GB for ~8K US dollar)
 - ◊ Software: I/O queue in Linux kernel is not well parallelized
 - ◊ Hardware: multi-queue is not here yet

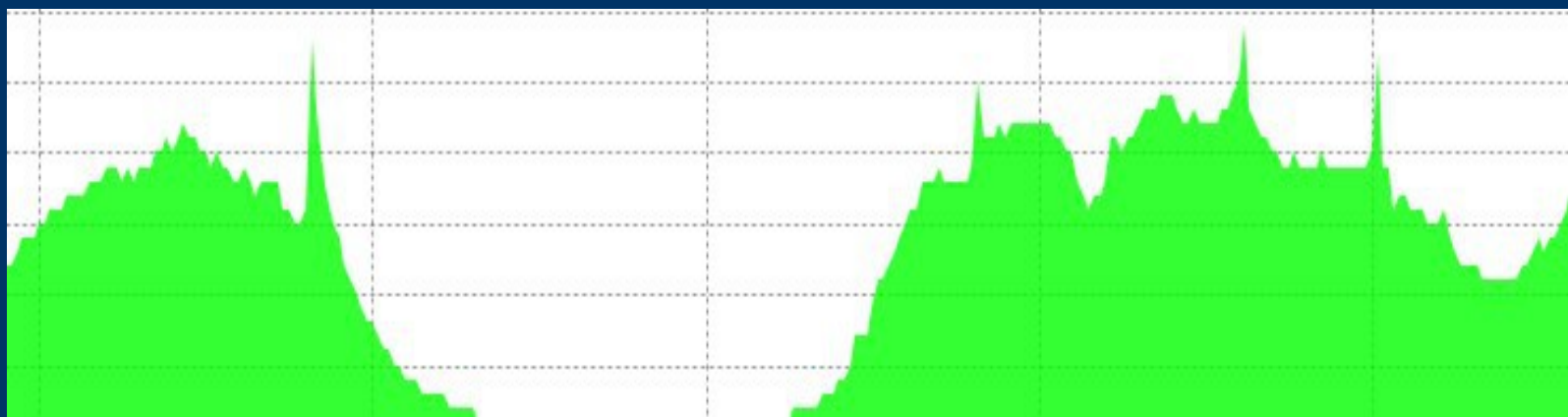
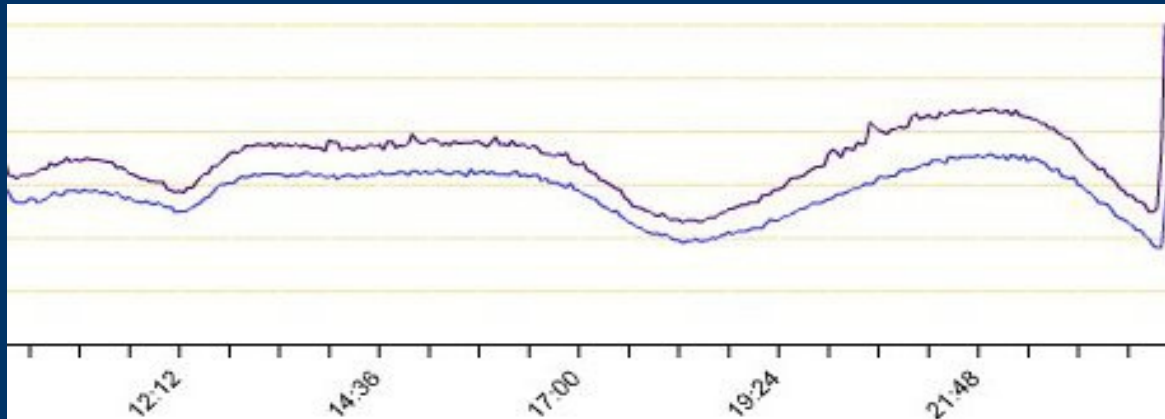
Characteristic of Taobao System Workload

From previous examples, we may make a conclusion on Taobao System Workloads,

- Most I/O binding VS. less CPU binding
- Data storage and meta data management are core of systems
- High loading in almost 7x24
- Real loads increase faster then system grows
- Caching is almost everywhere
- Virtualization has no usage case here
- Unbalance hardware components expense
- High power supply consuming

Characteristic of Taobao System Workload (Cont.)

- Don't forget --- seckill



Pictures from Taobao internal monitor system

Challenges and Opportunities

Challenges,

- Massive data I/O boosting in a little time
 - ◊ Really massive data (300+ Gbps)
 - ◊ 20+ Gbps random data request to main site
- Inefficient meta data operations
 - ◊ Consistency checking
 - ◊ Backup and restore
 - ◊ Intra nodes data synchronization
- Inefficient hybrid storage usage
 - ◊ Application level data migration between SSD and SAS/SATA disks
 - ◊ System is not designment ready for extremely high performance SSD
 - ◊ Need a better hybrid cache hierarchy with memory, SSD, SAS, SATA ...

| | |
|---------------------|------------------|
| DDR3 8G memory: | 30.00 USD per GB |
| X25 M 160G SSD: | 2.00 USD per GB |
| SG 15Krpm 300G SAS: | 0.60 USD per GB |
- Do we need so fast CPUs ?
 - ◊ Logical processing is not bottle neck
 - ◊ I/O on network, disks, memory is bottle neck
 - ◊ Cooling (more space in IDC) is bottle neck

Challenges and Opportunities

(Cont.)

Opportunities,

Every challenge is an opportunity, which one do we care ?

An open question still



Q & A



Thanks for your time !

