Topics

Tutorial 1: basic tools for IO measurements and caching effects
interactive tutorial

Exercise 1: implementation of Cloud Storage

Exercise 2: Basics of Encryption & RAID technology

Schedule

Wednesday 1 hour

Wednesday 2 hours

Thursday 0-1 hour

Thursday 1-2 hours
What do you know already?

Please open this anonymous online poll with your phone or laptop …

http://etc.ch/84Yd
Interactive Tutorial
IO Measurements

in non-virtual machines

Measurement Tools

read, write, time ...

strace

vmstat, iostat

Since we measure here, we measure all together down to the device (HD)

Here is the meta data cache implemented! (not important for the exercises)

Here is the data cache implemented! Important for the exercises!
Linux Performance Tools

- strace
- perf
- trace
- sysdig

Applications
- VFS
- File Systems
- TCP/UDP
- Scheduler
- Virtual Memory
- Clocksource
- Device Drivers
- Firmware

System Libraries
- System Call Interface
- Linux Kernel
- VFS
- File Systems
- TCP/UDP
- IP
- Ethernet

Hardware
- CPU 1
- GPU
- I/O Bridge
- FAN
- Power Supply

Observability tools
- static performance tools
- perf-tools/bcc tracing tools

These can observe the state of the system at rest, without load.

https://github.com/brendangregg/perf-tools
https://github.com/lovisor/bcc

style inspired by reddit.com/u/redct
http://www.brendangregg.com/linuxperf.html 2017
IO Performance
Throughput & Latency & IOPS

Throughput = IO volume / time
IOPS = IO operations / time
IO Performance

Latency with multiple clients

Single IO client

Multiple IO clients
IO Performance

IOPS improvements using queuing

IO Queuing allows to trade an average increase in latency for more IOPS! Requests can be reordered to match their physical location on disk and reduce the seek time.
Latency
Sequential Operations
Latency Optimization

Compound operations - vector read
Latency Optimization
Asynchronous Requests - read-ahead
Caching

Write-Through Cache
1. Commit
2. Acknowledge

Write-Back Cache
1. Acknowledge
2. Commit

Faster Performance Adds Risk
More Reliable Less Performance

Disk

Cache

commit

acknowledge

write

acknowledge

commit

Disk
Useful Commands

Linux

Realtime, CPU, System Time Measurement

time <program>

Copy/Block IO tool

dd
Useful Commands

Linux

- Buffer Cache/Block IO Activity
  vmstat -n 1

- Device Activity
  iostat -x 1
dstat

- Task Overview (proc interface)
top, htop, iftop ...
Useful Commands

Linux

- Trace system commands of a program
  strace <program> [program args]
  -f follow forks
  -c count sys calls
  -ttt show high time resolution
  → man strace!
Useful Commands

Linux

- Flush dirty pages as user or root:
  - `sync`

- Clean page cache:
  - as user:
    - write a new file bigger than the memory size
  - as root:
    - `echo 3 > /proc/sys/vm/drop_caches`
Exercise 1

http://cern.ch/go/9JGC
Cloud Storage Setup

BashCloud

REDIS Server

REDIS protocol

Storage Logic:
→ implemented in client by you!

scalable & fault tolerant
Cloud Storage

Object Storage

basic principles for the exercise

- sharding: files are placed and located using a distributed hash table (DHT)
  
  - the DHT can be changed to change the storage configuration
  
  - files are located computing the SHA1 checksum of their filename in hex representation
  
  - files get replicated to each neighbouring node e.g. every file has 3 copies

- files can be listed using a 'bucket'
File Location using Consistent Hashing

**Consistent Hashing**

- 160-bit integer keyspace
- divided into fixed number of evenly-sized partitions
- partitions are claimed by nodes in the cluster
- replicas go to the N partitions following the key

Hash Table = "Recipe to find a file by name"

<table>
<thead>
<tr>
<th>Hash Value</th>
<th>Node Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
</tr>
</tbody>
</table>

**primary copy**

**'left' copy**

**'right' copy**
Bucket Implementation

- buckets are represented by a set of file names e.g.
  ```
  ls /
  1.jpg
  2.jpg
  3.jpg
  ```
  
- a set is more suitable than a list because it does not allow duplicated file names

- one can also shard buckets for scalability purposes
  
- to list a directory one combines the listing of all participating servers
Basic Ingredients of Cloud Storage

**Objects:**
- K-V Store API
- GET SET DELETE KEYS

**Collections:**
- SET API
- ADD DELETE LISTMEMBERS

**Scalability:**
- Sharding of Objects and Collections

**Redundancy:**
- Replication & Erasure Encoding
KV Stores
Simple and Complex Data Types available in REDIS

Strings/Blobs/Bitmaps
Hash Tables (objects!)
Linked Lists
Sets
Sorted Sets

Files
Key
Bucket
File

DHT

Node 1
- count=1
- count=1

Node 2
- count=1

Node 3
- count=1
- count=1

Node 4
- count=2

hash by content
According to the CAP theorem, is that an CA, CP or AP system?

C = Consistency  A = Availability  P = Partition Tolerance
According to the CAP theorem, is that an CA, CP or AP system?

C = Consistency  A = Availability  P = Partition Tolerance
Exercise 2
Parity

Parity defined by XOR Operation

\[ P = X \oplus Y \]

<table>
<thead>
<tr>
<th>XOR</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Parity algebra for reconstruction:

\[ P = X \oplus Y \]
\[ Y = P \oplus X \]
\[ X = P \oplus Y \]
Data Striping

Example RAID-0 doubles avg. throughput for rw double avg. IOPS