PerfKit
Benchmarking the Cloud

Ivan Santa Maria Filho
Senior Staff Engineer
Google Cloud Performance
Big Data

“Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization.”

- Gartner

“Big data uses inductive statistics and concepts from nonlinear system identification to infer laws (regressions, nonlinear relationships, and causal effects) from large sets of data with low information density to reveal relationships, dependencies and perform predictions of outcomes and behaviors.”

- Pierre Delort
Deliver the Best Performing Cloud Platform

Best Price/Performance, Efficient, Highest Throughput - Lowest Latency, Predictable, Scalable, Easy to Analyze

- Provide expert benchmarking
- Capture a competitive view of our products to support business decisions
- Guide our customers to better performance through talks, papers and tools
- Support sales on competitive deals
- Drive end-to-end performance improvements with internal teams
- Develop a vibrant community of benchmarkers and partners for cloud performance
Cloud Platform Performance Team

Open Source Tools

Blog Posts

Training

Conferences

Competitive Analysis

Analytics

Product Improvements
Cloud Computing

Packaged Software
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

IaaS
- Infrastructure-as-a-Service
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

PaaS
- Platform-as-a-Service
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

SaaS
- Software-as-a-Service
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

Google

You Manage  Vendor Managed
The Challenge - Competitive Analysis

- Incredible product complexity
- “Devices” that aren’t devices
- Data centers that are unlike anything else
- Custom virtualization software
- Geo distribution product design and SLAs
Approach One - Benchmarking

Benchmarking assumes the existence of a model that approximates your application

Instruction counts

*ips, kips, mips, mops, flops*

Synthetic workload simulation

(72) Whetstone, LINPACK (84) Dhrystone (85) GUI primitives

http://en.wikipedia.org/wiki/Instructions_per_second

http://www.top500.org/lists/
SPEC Score Changes With Utilization

Relative SPEC score depends on utilization
Approach Two - Capture and Playback

Hidden variables
Confidence intervals
Sampling distribution
Compensating for complexity
Platform obfuscation

Customers and data sources spread across telecom providers and cities
Approach Three - Monitoring Production

Circadian Rhythm of the Internet

Provider Specific Fluctuations
A Bit Different

Cloud is *elastic*, so we track deployment times

Step 1: Setup
Step 2: Warm up
Step 3: Pre-execute
Step 4: Execute
Step 5: Post-Execute
Step 6: Cleanup
Step 7: Publish results

Traditional benchmarking
Our Approach - All of The Above

Sliding window containing 10 billion benchmark results, each representing tens of thousands of data points, in close enough to real-time

Control for known variables
- Geography and topology
- Time of day
- Noisy neighbours

Proper sampling
- Define our target confidence interval
- Adjust by using the appropriate distribution tables

http://en.wikipedia.org/wiki/Student%27s_t-distribution
Hidden Variable - IO Bursting Models

Persistent Disk burst (no reload)  Networking throughput burst (reload)

Random Reads, 4KB blocks, Queue = 64

Random Writes, 4KB blocks, Queue = 64
Hidden Variable - Networking scales with VM Size
Hidden Variable - Image choice has a *BIG* Impact

Settings only!
Hidden Variable - Guest IO Scheduler

Comparison of storage variance by block scheduler in the guest

PD and guest scheduler have strange interactions
Hidden Variable - Choice of Zone Matters!

US-CENTRAL1-[A|B] and EUROPE-WEST1-[B]
  SandyBridge

ASIA-WEST1-[A|B|C] and US-CENTRAL1-F and EUROPE-WEST1-C
  IvyBridge - Can get 15% better computation for some workloads. Has AVX and SSE4

US-CENTRAL1-C and EUROPE-WEST1-D
  Haswell - Has AVX2
Hidden Variable - Zone Throughput can be Asymmetrical

Internal IPs are much faster!

Single Stream TCP Throughput Zone Comparison for the last 7 days (min, 5%, 95%, max)
PerfKit Benchmarker and Explorer

The Benchmarker for running benchmarks

The Explorer for dashboarding and analysis
**PerfKitBenchmarker**: The framework for running benchmarks

**Workloads**
- Aerospike
- Cassandra
- Hadoop
- HPCC
- MongoDB
- Oldisim
- Redis

**CPU**
- Coremark
- Spec CPU 2006

**Network**
- Iperf
- Mesh
- Network
- Netperf
- Ping

**Disk**
- Bonnie++
- Copy
- Fio
- Synthetic Storage

**Varied**
- Cluster Boot
- Object Storage
- Sysbench OLTP
- Unixbench
Example: Measuring boot time
./pkb.py --project=google.com:voellm --cloud=GCP -- benchmarks='cluster boot'

Example: Measuring VM to VM performance cross zone
./pkb.py --project=google.com:voellm --cloud=GCP --zones=us-central1-a,us-central1-b --benchmarks='iperf'

Example: Running in stages
Flags for stages: [--run_stage=prepare|run|publish|cleanup -- run_uri=<id>]

PerfKit Benchmarker: Example Runs
Dashboards are interactive. Click to edit.

Container is a row. Widget is an element. Use to control placement.

Save As to create your own copy

Click the X to delete a Widget
PerfKit Explorer: Analytics Dashboards

- Controls for the Widget Data and Transform
- Controls for the Dashboard and Widget Display
- Query Editor for the power user!
PerfKit Explorer: How It’s Built

Embracing of Gustafson’s Law
BigQuery

Unlimited storage
Fully managed
Massively parallel query processing
Unstructured, semi-structure, and a structured data
Rich support of analysis functions like ABS, COS, …, CUME_DIST, NTILE, PERCENT_RANK, …
Example BigQuery Select

```
SELECT
    USEC_TO_TIMESTAMP(UTC_USEC_TO_DAY(INTEGER(timestamp * 1000000))) as sorttime,
    MIN(value) as MIN_VALUE,
    AVG(value) as AVG_VALUE,
    MAX(value) as MAX_VALUE
FROM
    samples_mart.results, samples_mart.results_archive
WHERE
    timestamp > TIMESTAMP_TO_SEC(DATE_ADD(CURRENT_TIMESTAMP(), -180, "DAY")) AND
    product_name = "Artemis" AND
    test = "iperf" AND
    metric contains "Throughput" AND
    labels contains "ip_type:internal" AND
    labels contains "n1-standard-8" AND
    ((REGEXP_EXTRACT(labels, r'\|client_zone:(.*?)(\|') = REGEXP_EXTRACT(labels, r'\|server_zone:(.*?)(\|')) OR
    (REGEXP_EXTRACT(labels, r'\|sending_zone:(.*?)(\|') = REGEXP_EXTRACT(labels, r'\|receiving_zone:(.*?)(\|')))) AND
    owner = "jenkins"
GROUP BY sorttime
ORDER BY sorttime asc
```
BigQuery Fun Stats

Largest query by **rows**.... 10.5 Trillion
Largest query by **bytes**....... 2.1 PB
Largest **storage** customer....... 38 PB
Streaming **Ingest** at peak........ 2.4 Million rows per second

Stats For A Complex 100B Row Query  Backend Resources
Execution Time... 11 Seconds  3600 HDD (@ 100 MBps)
Data Scanned... 4 TB  8800 vCPUs (@ 1 μsec each)
Regexes Executed... 100 Billion  200 Gbps
Data Shuffled... 278 GB
BigQuery Peeking Under the Hood

BigQuery Column IO Storage

Record Oriented Storage

Column Oriented Storage

Read query column data, not records (row based storage)

Data Sharding and Replication
- Performance
- Durability
- Better Compression
- Reducing Tail latency (hedge and tie)
BigQuery Under the Hood: Key Concepts

- Mixer 0
- Mixer 1
- Leaf
- Columnar Storage

Partial Reduction
Diskless data flow
Long lived shared serving tree
Columnar Storage

Distributed Storage
PerfKit End to End

Streaming API - 100k rows/sec

Google Cloud Storage

Google BigQuery

Metadata Cubes (AppEngine)

Spreadsheets

Coworkers

Explorer (AppEngine)
Q & A

Ivan Santa Maria Filho
Senior Staff Engineer
Google Cloud Performance

cloud.google.com