Kubecon Higgs Analysis

Demo Deep Dive

Lukas Heinrich, Ricardo Rocha
Our job: look at data and check against multiple theories (Higgs, SUSY, …)

Fundamental problem 2: do not have a simple way to predict what data would look like under **different theories / assess compatibility**

Solution:
Use **large scale compute** to process data 
+ **deep stack of software** to brute-force what data looks like under theories (Monte Carlo)
Baseline cannot describe data
… but baseline + new physics theory does -> Discovery!
Back in 2012, CERN announced one of its most important achievements, the discovery of the Higgs boson leading to the 2013 Nobel Prize in Physics.

In this presentation, we will redo the data analysis that led to it, this time on top of Kubernetes, the new infrastructure stack growing in popularity in the laboratory.
Demo Idea: reproduce Higgs discovery

![Higgs boson diagram](image)

CMS Observed
- Z+X
- Z^+, ZZ
- m_h=126 GeV

Events / 3 GeV

<table>
<thead>
<tr>
<th>m_W (GeV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>120</td>
</tr>
<tr>
<td>140</td>
</tr>
<tr>
<td>160</td>
</tr>
<tr>
<td>180</td>
</tr>
</tbody>
</table>

\[ \frac{s}{7 (8) \, \text{TeV}, L = 5.1 \, (12.2) \, \text{fb}^{-1}} \]
Demo

https://github.com/cernops/higgs-demo
Sim Higgs

Background

Background

Real Data

Event Data

20k+ Core K8s Clusters

Summary Data

Make Plot!

70 TB of Physics Data

~25000 Files

~25000 Files
OpenStack Magnum

70 TB Dataset

25000 Kubernetes Jobs

CERN

Aggregation

Interactive Visualization
Moving data from EOS to CERN S3

Initial dataset (opendata) available on /eos

S3 is more cloudy, we wanted to test with that to ease transition to GCP

https://gitlab.cern.ch/rbritoda/eos2s3

(a Kubernetes backed dummy file transfer service)

16 parallel transfer processes
Moving data from EOS to CEPH/S3

- s3cmd (single transfer)
- s3cmd (parallel)
- aws cli (multipart)

~160MB/s
Moving data from EOS to CEPH/S3

Big thanks to the CEPH team
CERN Analysis Run

Kubernetes 1.12

61 Nodes (VMs)

- 4 Cores / 8 GB nodes
- 40GB disks (SSDs)

Running on 36TB (half the dataset)

Total time: 19h with 244 cores (~1GB/s)

Goal for GCP: 250x speedup to run it in <10min
CERN Analysis Run
Cluster on GKE
Max 25000 Cores
Single Region, 3 Zones
25000 Kubernetes Jobs

70 TB Dataset

Google Cloud Storage

Job Results

Interactive Visualization

Aggregation

Google Cloud
First Transfer to Zurich Region

Added GCS support to eos2s3 (using gsutil)

Zurich because closer is better?

But using the Internet 70TB will take a long time… or?
First Transfer to Zurich Region

Added GCS support to eos2s3 (using gsutil)

Zurich because closer is better?

But using the Internet 70TB will take a long time… or?

Network Traffic Received

Late night good news!
First Transfer to Zurich Region

Added GCS support to eos2s3 (using gsutil)

Zurich because closer is better?

But using the Internet 70TB will take a long time… or?

Sample destination: zrh04s15-in-f10.1e100.net
Traceroute from kubecon-demo-012-2elgq755lsas-minion-77 (i692316327109xx.cern.ch):
traceroute to zrh04s15-in-f10.1e100.net (172.217.168.74), 30 hops max, 46 byte packets
1 10.100.104.1 (10.100.104.1) 0.005 ms 0.005 ms 0.003 ms
2 l513-v-rbrmx-1-xxx.cern.ch (10.42.xx.x) 1.351 ms 0.147 ms 0.123 ms
... 8 e773-e-rbrxl-2-xxx.cern.ch (192.65.xx.xx) 1.152 ms 1.172 ms 0.996 ms
9 google-zurich-10g.cern.ch (192.65.184.202) 6.102 ms 6.176 ms 6.339 ms
10 74.125.243.113 (74.125.243.113) 7.166 ms 7.001 ms 6.996 ms
11 172.253.50.5 (172.253.50.5) 15.032 ms 64.233.175.167 (64.233.175.167) 7.797 ms
8.035 ms
12 zrh04s15-in-f10.1e100.net (172.217.168.74) 7.012 ms 6.924 ms 7.109 ms

Late night good news!
First Transfer to Zurich Region

Added GCS support to eos2s3 (using gsutil)

Zurich because closer is better?

But using the Internet 70TB will take a long time… or?

https://network.cern.ch/sc/fcgi/sc.fcgi?Action=SearchForDisplay&DeviceName=cixp-google

Late night good news!
Retransfer to NL Region

On Google’s request

Higher flexibility in terms of available capacity

Why retransfer? Can’t GCP replicate cross regions?

Ingress is free, Ingress is free

( even when running on credits, this counts )
Retransfer to NL Region

1 day to transfer the full 70TB dataset

Still going through Google Zurich first

Direct NL/100Gb possible?

Similar rate as for Zurich
Retransfer to NL Region

Google at Zurich

<table>
<thead>
<tr>
<th></th>
<th>Avg</th>
<th>Max</th>
<th>last</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>to CERN</td>
<td>170.71M</td>
<td>425.05M</td>
<td>234.68M</td>
<td>Peak: 3.63G</td>
</tr>
<tr>
<td>to Google</td>
<td>114.34M</td>
<td>6.49G</td>
<td>98.46M</td>
<td>Peak: 8.49G</td>
</tr>
</tbody>
</table>

Last update: Thu Jul 04 2019 12:53:11
Learning the Ropes

Dress Rehearsals
(looking good)

Live Demo
GCP Analysis Run

Kubernetes clusters on GKE (Managed Kubernetes service on GCP)

Today’s run included (real demo run was ~2x that)

660 nodes: n1-highmem-16, 104 GB RAM

10560 cores, 69 TB RAM
Single cluster, slow scheduling

Kubernetes components throttle queries to the api-server

Defaults are very conservative (20 QPS)

Would mean a very slow job scheduling rate (5 pods/sec?)

We knew this from previous scale tests we’ve done at CERN

Currently we cannot tune this in GKE, coming soon

Decision: split the load into multiple clusters
# Storage choices

<table>
<thead>
<tr>
<th></th>
<th>Zonal standard persistent disks</th>
<th>Regional persistent disks</th>
<th>Zonal SSD persistent disks</th>
<th>Regional SSD persistent disks (SCSI)</th>
<th>Local SSD (NVMe)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximum sustained IOPS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read IOPS per GB</td>
<td>0.75</td>
<td>0.75</td>
<td>30</td>
<td>30</td>
<td>266.7</td>
</tr>
<tr>
<td>Write IOPS per GB</td>
<td>1.5</td>
<td>1.5</td>
<td>30</td>
<td>30</td>
<td>186.7</td>
</tr>
<tr>
<td>Read IOPS per instance</td>
<td>3,000</td>
<td>3,000</td>
<td>15,000 - 60,000*</td>
<td>15,000 - 60,000*</td>
<td>400,000</td>
</tr>
<tr>
<td>Write IOPS per instance</td>
<td>15,000</td>
<td>15,000</td>
<td>15,000 - 30,000*</td>
<td>15,000 - 30,000*</td>
<td>280,000</td>
</tr>
</tbody>
</table>
Storage choices

We first chose persistent SSDs (~15000 IOPS)

And saw huge amount of io wait, nodes evicted

As we were network and storage service bound at CERN, this was not obvious…
Storage choices

GCP network guarantees 2Gb/core up to 16 core nodes

With 16 core nodes, we get **32 Gb per VM**!

GCS can handle these rates somehow, and we end up bound by local i/o

Network throttling?

```
trickle -s -d $DOWNLOAD_MAX_KB -u $UPLOAD_MAX_KB gsutil cp ... - | cat > $DESTFILE
```

Helps, but not to stay under 10min… local SSDs? That’s 280000 IOPS, enough!

But one only gets full disks (375GB) and it’s a scarce resource
Storage choices

In the end we relied on shared memory for storage

Exploring high memory instances

*medium: Memory*
GCP Pricing

Billing is updated daily, though there are APIs to query for details

Considering a ~10 minutes run it implies (compute table prices, NL region)

\[
$1.043 \times 1530 / 6 = $260 \text{ (~5x cheaper if using pre-emptibles)}
\]

Parking storage cost for the dataset (monthly cost, lots of room for creativity)

\[
$0.020 \times 70000 = $1400
\]

Total under $300 usd

Running on credits, no Committed Use or Sustained Compute discounts
Open Data, Reproducibility, Reusability

The LHC is a unique machine. Likely that no other machine will probe the same physics regime. Preserving our work for future use-cases is crucial.

Two approaches:

The “Museum”: preserve by archiving / documenting

The “Hangar”: preserve to be reused / stay operational
Open Data, Reproducibility, Reusability

When Preserving for reusability two choices

- Open Ended New Research (Open Data)
- Reinterpretation of analyzed data (RECAST)
Lot of Love for CERN Open Data. But common reservation: Is it realistic to even analyze PB scale data as an individual without much infra (e.g. non-CERN physicists)?

Demo proved that you can get the necessary scale easily on public cloud. Expect prices to drop.
But data is not enough! Need to have software too!

What does it mean? Spacetime invariance.

Preserved if you can run it in the future in a different data center

Preservation == Reproducible Deployment…
Containerized Workflows

But Software Preservation is not enough
Need to know what to do with it?

Declarative, cloud-native pipelines are portable way to preserve analyses

Used in CERN Analysis Preservation and REANA to re-execute old analyses
Credits

Clemens Lange

Thomas Hartland

Google and CERN openlab for the credits and support

And of course the Kubernetes community
Questions?
<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
<th>Cost (CHF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul 1 - 4, 2019</td>
<td>Compute Engine N1 Predefined Instance Ram running in Netherlands: 24588.579 Gibibyte-hours</td>
<td>112.07</td>
</tr>
<tr>
<td></td>
<td>[Currency conversion: USD to CHF using rate 0.977] (Source:Kubecon Demo [nimble-valve-236407])</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Compute Engine N1 Predefined Instance Core running in Netherlands: 3845.211 Hours</td>
<td>130.78</td>
</tr>
<tr>
<td></td>
<td>[Currency conversion: USD to CHF using rate 0.977] (Source:Kubecon Demo [nimble-valve-236407])</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cloud Storage Regional Storage Netherlands: 6984.399 Gibibyte-months</td>
<td>136.51</td>
</tr>
<tr>
<td></td>
<td>[Currency conversion: USD to CHF using rate 0.977] (Source:Kubecon Demo [nimble-valve-236407])</td>
<td></td>
</tr>
</tbody>
</table>