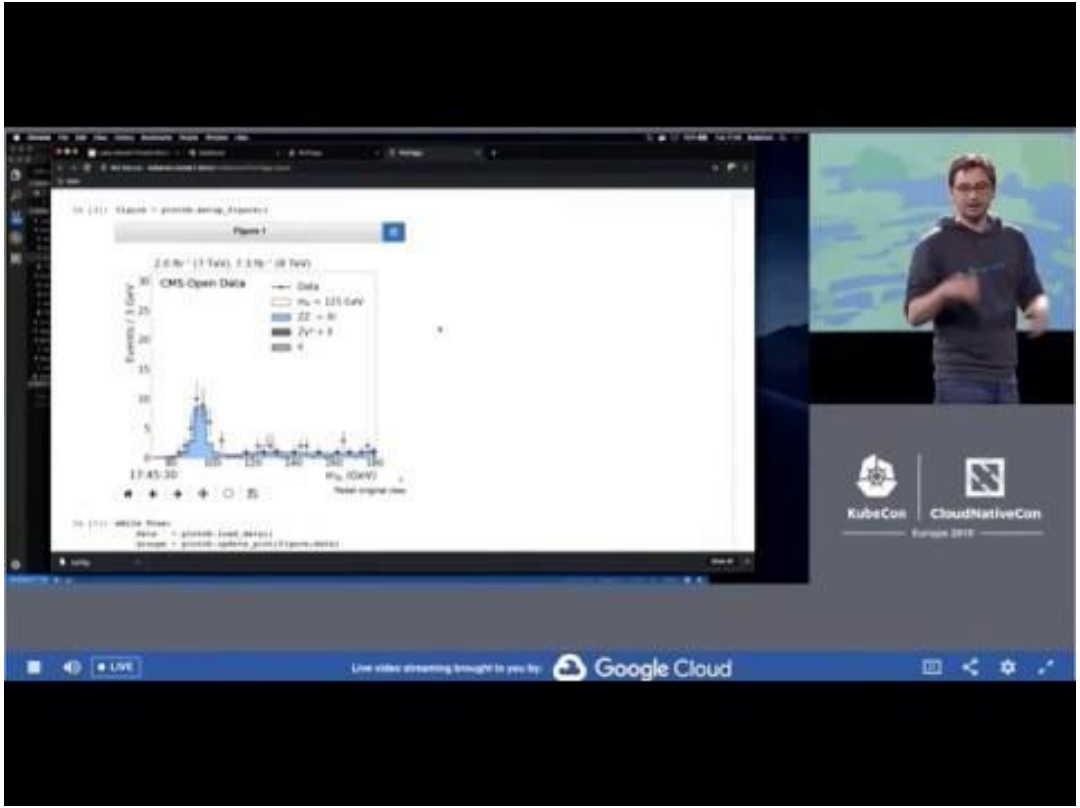
Reperforming a Nobel Prize Discovery on Kubernetes and the Google Cloud

Clemens Lange, Lukas Heinrich, Ricardo Rocha

Google Cloud

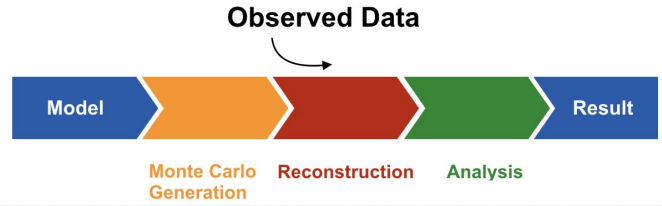


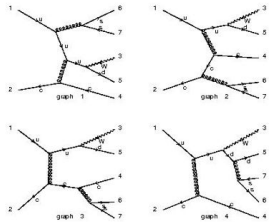
https://www.youtube.com/watch?v=CTfp2woVEkA

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

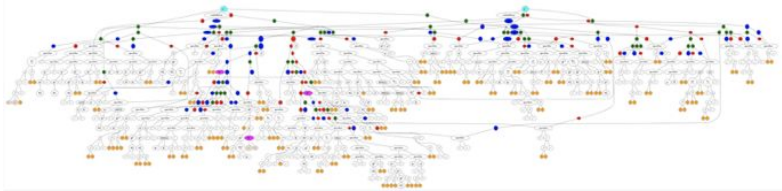
Fundamental problem 1: looking for rare phenomena. Needs lots of data.
Fundamental problem 2: do not have a simple way to predict what data would look like under **different theories / assess compatibility**

Solution:
Use **large scale compute** to process data
+ **deep stack of software** to brute-force what data looks like under theories (Monte Carlo)

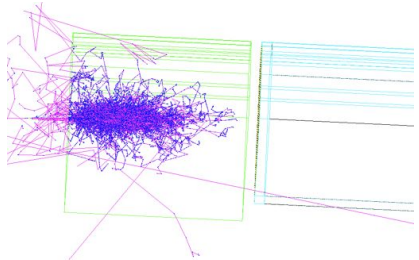




High Energy Physics

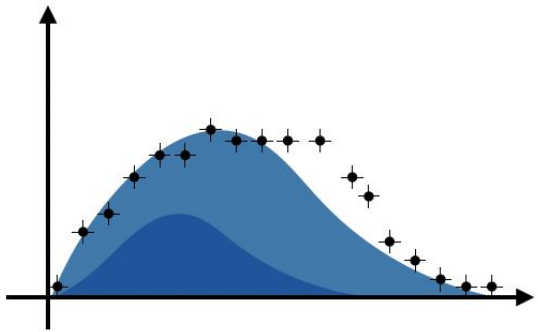


Event Evolution

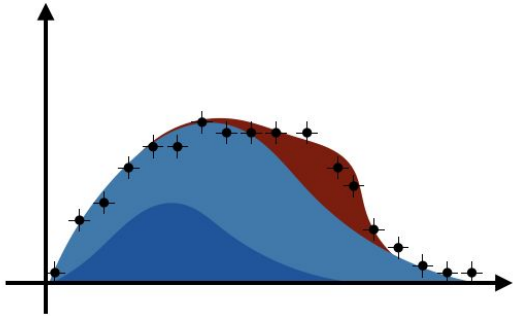


Detector Interactions

Baseline only

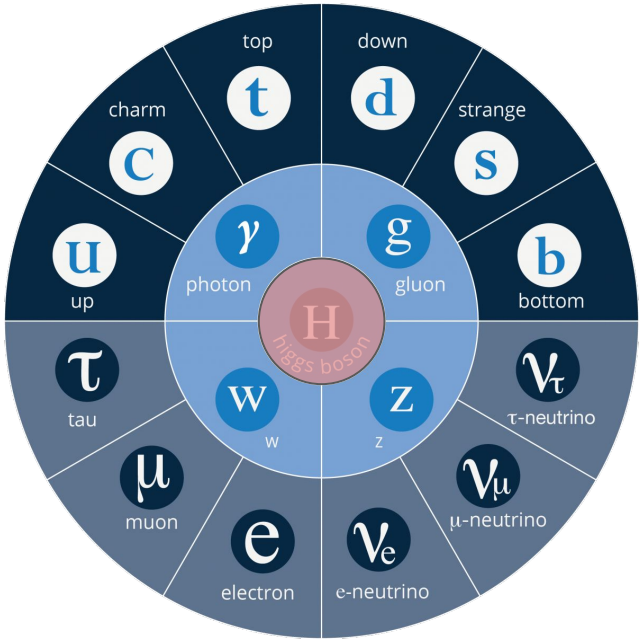


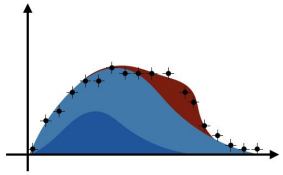
Baseline +
New Physics

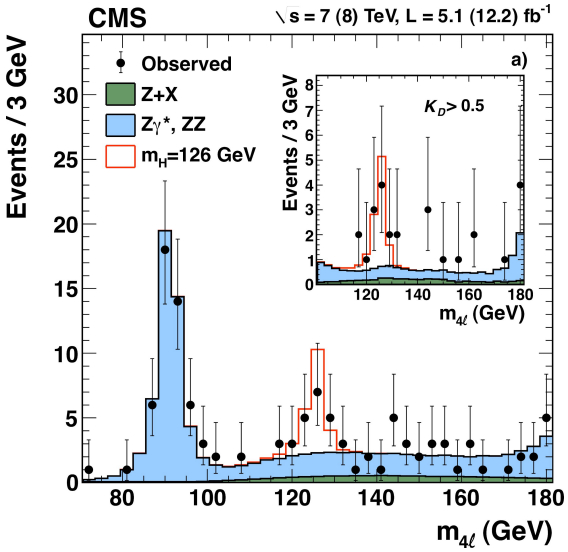


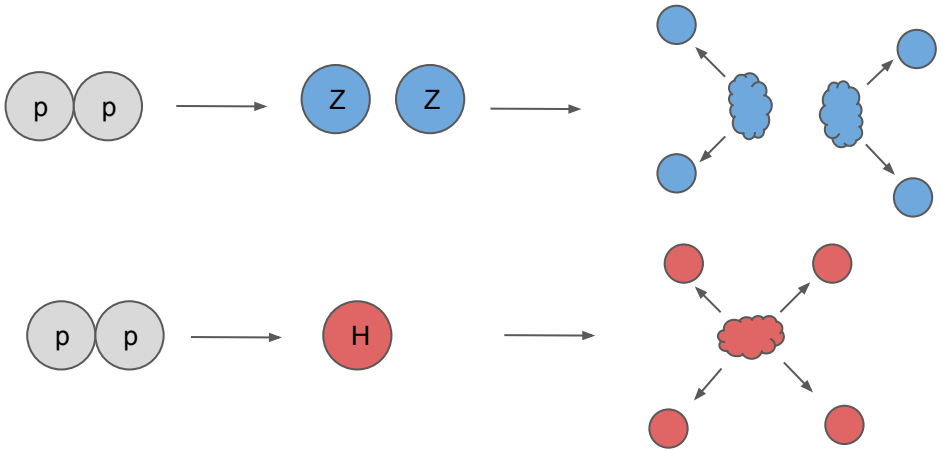
Baseline cannot describe data
… but baseline + new physics theory does -> Discovery!

Demo Idea: reproduce Higgs discovery









Open Data accessible to everyone at scale

LHC experiments part of growing list of experiments with complex open data problems: data complexity, data volume. large collaborations.

Can it be used? And can we use the public cloud to scale on demand?

```
[16:01:21] cmsusr@e6f7bea2253e /Users/lukasheinrich/Code/awesomedemo/higgs-demo/CMSSW_5_3_32/src $ \root -b
  ****************************************
  *                                      *
  *       W E L C O M E  to  R O O T      *
  *                                      *
  *   Version   5.32/00   2 December 2011   *
  *                                      *
  *  You are welcome to visit our Web site  *
  *         http://root.cern.ch          *
  *                                      *
  ****************************************

ROOT 5.32/00 (branches/v5-32-00-patches@42372, Jun 10 2014, 18:26:00 on linuxx8664gcc)

CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
```

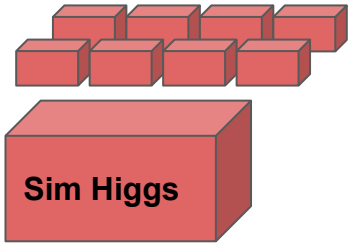
Beyond a VM: Containerized CMSSW

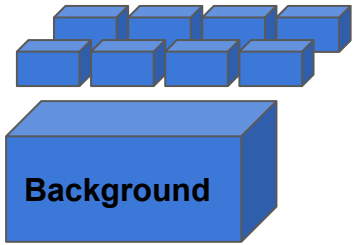
~decade old software to reproduce results

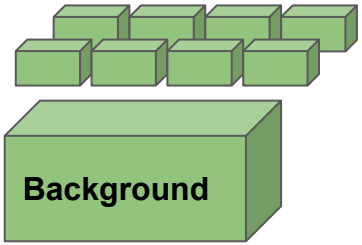
cmsopendata/cmssw_5_3_32 ☆

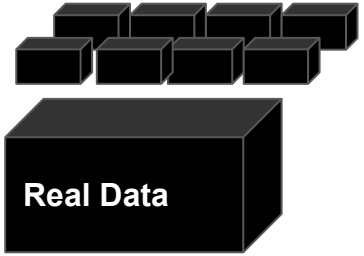
By cmsopendata • Updated 4 months ago

Container

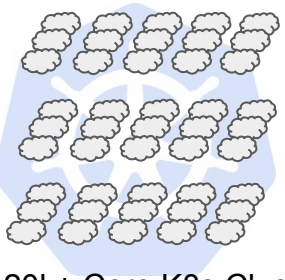








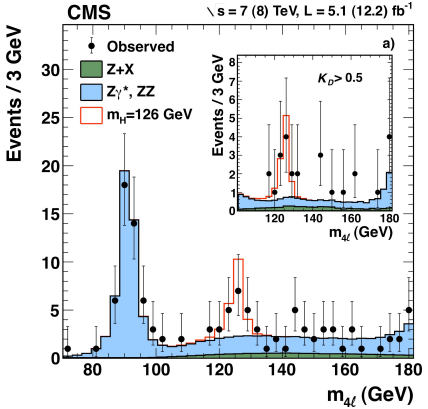
Event Data



20k+ Core K8s Clusters

Summary Data

Make Plot!



70 TB of Physics Data

~25000 Files

Kubernetes

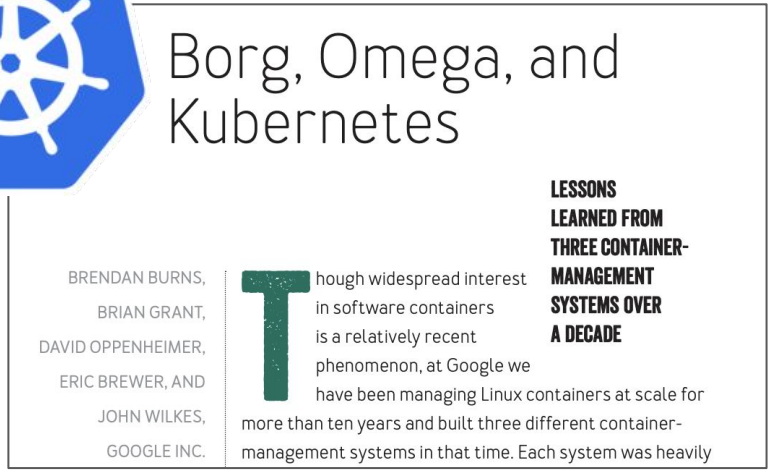
Largest open source project after kernel

- **35.000** contributors, **148.000** code commits
- **83.000** pull requests, **1.1M** contributions
- **2000+** contributing companies

Google, RedHat, VMware, Huawei, Microsoft, IBM, Fujitsu, …

Open community welcome to contributions

Special Interest Groups (SIGs) : Auto-Scaling, Multi-Cluster, Scheduling, ...



Borg, Omega, and Kubernetes

LESSONS LEARNED FROM THREE CONTAINER-MANAGEMENT SYSTEMS OVER A DECADE

BRENDAN BURNS, BRIAN GRANT, DAVID OPPENHEIMER, ERIC BREWER, AND JOHN WILKES, GOOGLE INC.

Though widespread interest in software containers is a relatively recent phenomenon, at Google we have been managing Linux containers at scale for more than ten years and built three different container-management systems in that time. Each system was heavily

Kubernetes

Google Cloud Platform

OPENSHIFT

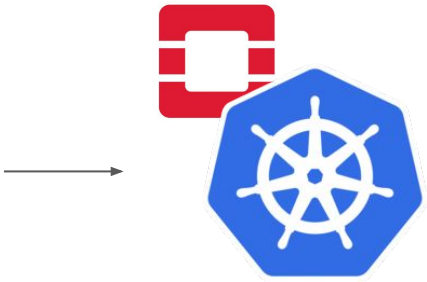
Rancher
Kubernetes Engine

MAGNUM
an OpenStack Community Project

kubeadm

Common declarative API for basic infrastructure : compute, storage, networking

Healthy ecosystem of tools offering extended functionality



70 TB Dataset

OpenStack Magnum

25000 **Kubernetes Jobs**

Job Results

Interactive Visualization

Aggregation

CERN openlab

Google Cloud

70 TB Dataset

Cluster on GKE

Max **25000 Cores**

Single Region, 3 Zones

25000 **Kubernetes Jobs**

Job Results

Interactive Visualization

Aggregation

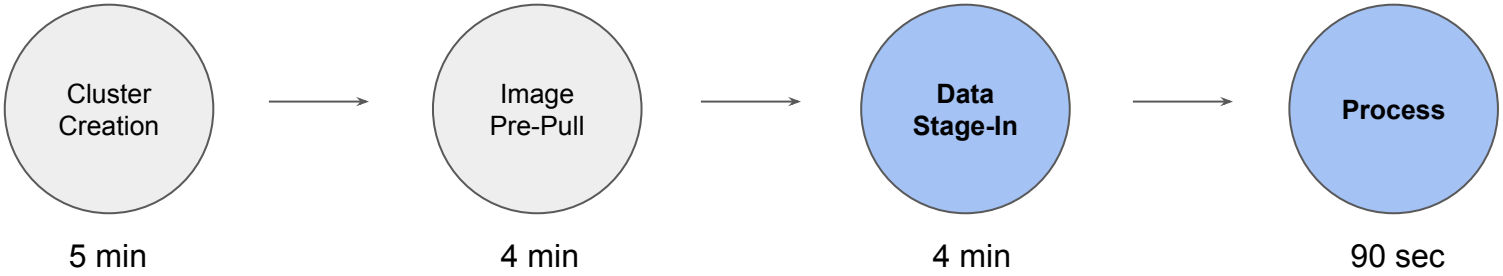
GCP Analysis Run

Kubernetes clusters on GKE (Managed Kubernetes service on GCP)

Run included

~1200 nodes: n1-highmem-16, 104 GB RAM

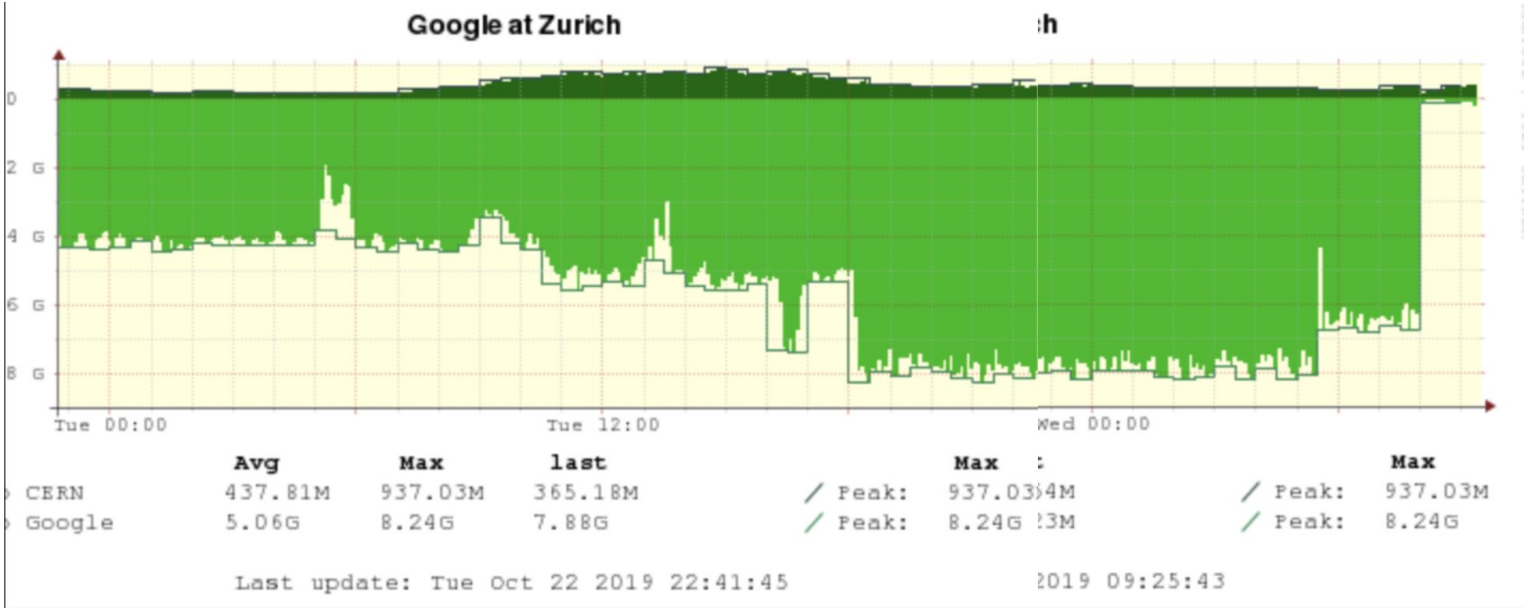
~20000 cores, ~120 TB RAM



Data Upload

Initial dataset (opendata) available on /eos

Ingress is free, Ingress is free...



GCP Analysis Run

Network guarantees 2Gb/core up to 16 core nodes (**32 Gbit per VM** !)

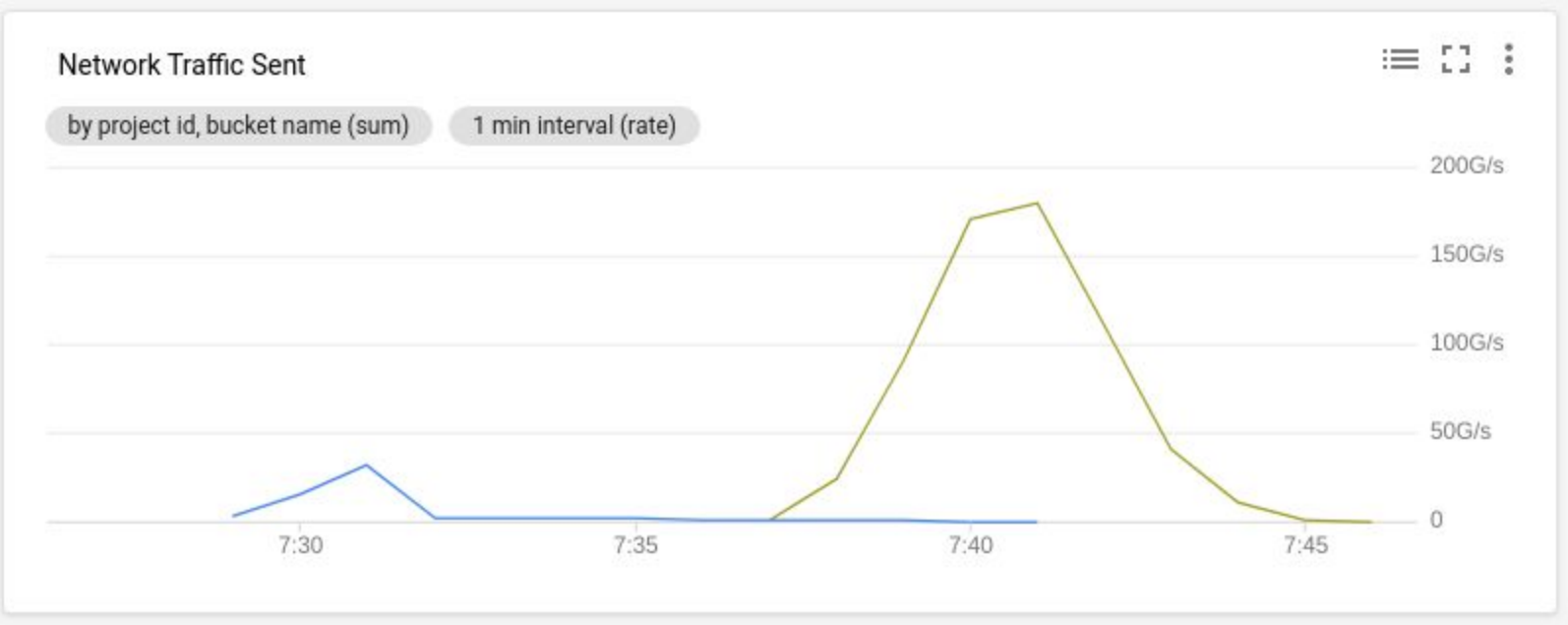
GCS can handle these rates somehow, and we end up bound by local i/o

Ended up using in-memory filesystems to go around this

	Zonal standard persistent disks	Regional persistent disks	Zonal SSD persistent disks	Regional SSD persistent disks	Local SSD (SCSI)	Local SSD (NVMe)
Maximum sustained IOPS						
Read IOPS per GB	0.75	0.75	30	30	266.7	453.3
Write IOPS per GB	1.5	1.5	30	30	186.7	240
Read IOPS per instance	3,000	3,000	15,000 - 60,000*	15,000 - 60,000*	400,000	680,000
Write IOPS per instance	15,000	15,000	15,000 - 30,000*	15,000 - 30,000*	280,000	360,000

GCP Analysis Run

Network guarantees 2Gb/core up to 16 core nodes (**32 Gbit per VM** !)



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 * 1530 / 6 = **$260** (**~5x cheaper if using pre-emptibles**)

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

$0.020 * 70000 = $1400

Total under $300 usd

Running on credits, **no Committed Use or Sustained Compute discounts**

Conclusions

Kubernetes and other tools in its ecosystem are making our life easier

- We are not alone … and can focus more on the upper layers, and physics
- Kubernetes is a great candidate for a common layer

Public cloud seems like a viable option in different cases

- Opportunities to efficiently scale out, access to otherwise scarce resources
- “Infinite” scale, pay for actual use
- Disaster Recovery

In [2]:
```
import json
import matplotlib.pyplot as plt
import plotting.plotnb as plotnb
```

In [3]:
```
figure = plotnb.setup_figure()
```