IT for Research: a Journey from In-House HPC Clusters to Public Cloud Infrastructure

Riccardo Murri,
riccardo.murri@gmail.com

2017-11-06

Two disclaimers

Opinions and views expressed here are mine only, and may not reflect the official stance of UZH, its IT services, or my colleagues.

Although I have tried to report on scientific research accurately, there can still be errors and inaccuracies. They are all my faults.

What is Research IT?

. . . in the words of those who do it

> “S3IT supports UZH researchers
> in using IT to empower their research,
> from consultancy to application support
> and access to cutting-edge cloud, cluster
> and supercomputing systems.”
>
> (source: https://www.s3it.uzh.ch/)

. . . in the words of those who *use* it

From: some.one@uzh.ch
Subject: computing power

Dear Madam/Sir,

I have been invited to submit a revision of the attached paper. There are some missing numbers in Table 1, since I did not have enough computing power on my office computer to carry out these computations. A referee has asked us for them, therefore I need access to a supercomputer.

Many thanks,
Some One

We're the ones with the big computers!

Man and woman working with IBM type 704 electronic data processing machine used for making computations for aeronautical research.

Image source: Wikimedia

Traditional options for scientific computing

- Personal workstations
  - Limited: how much computing power can fit under your desk?
  - More freedom, more responsibilities
- Large shared batch-queuing systems
  - Centrally provided and administered
  - Typically a GNU/Linux cluster nowadays.

Large, shared, batch-queuing clusters

Large

Shared

Batch-queuing clusters

Can allow much larger degree of parallelism compared to own workstation.

Limit is institutional money. (Dept., University, National Computing Centre)

Large, shared, batch-queuing clusters

Large

Shared

Batch-queuing clusters

In particular: *same* OS and *same* set of installed software for all, *same* scheduler configuration for all, *same* filesystem(s) for all . . .

So, installed software and usage is subject to **policies**.

Large, shared, batch-queuing clusters

Large

Shared

Batch-queuing

clusters

"Cluster" is the architecture:

- standard ("commodity") servers as compute nodes
- high-performance network interconnecting them
- shared filesystem(s)
- job scheduler to allocate resources

Reference: D. Becker, Th. Sterling, et al.: *BEOWULF: A parallel workstation for scientific computation*, in: Proceedings, International Conference on Parallel Processing vol. 95, (1995). http://www.phy.duke.edu/~rgb/brahma/Resources/beowulf/papers/ICPP95/icpp95.html

Large, shared, batch-queuing clusters

Large

Shared

Batch-queuing

clusters

Extremely successful architecture, took over HPC since the late 1990's:

- replacing (expensive, inflexible) HW with SW
- worked because of economies of scale: more parallelism, more affordable

Reference: D. Becker, Th. Sterling, et al.: *BEOWULF: A parallel workstation for scientific computation*, in: Proceedings, International Conference on Parallel Processing vol. 95, (1995). http://www.phy.duke.edu/~rgb/brahma/Resources/beowulf/papers/ICPP95/icpp95.html

Large, shared, batch-queuing clusters

Large

Shared

Batch-queuing
clusters

Commands to be executed are submitted to a job scheduler. (A command to be run is named a "job".)

The scheduler maintains a queue of pending jobs.

The scheduler's purpose is:

- to prioritize jobs in the queue,
- to allocate resources to starting jobs.

Site policies affect both stages.

PKDGRAV3

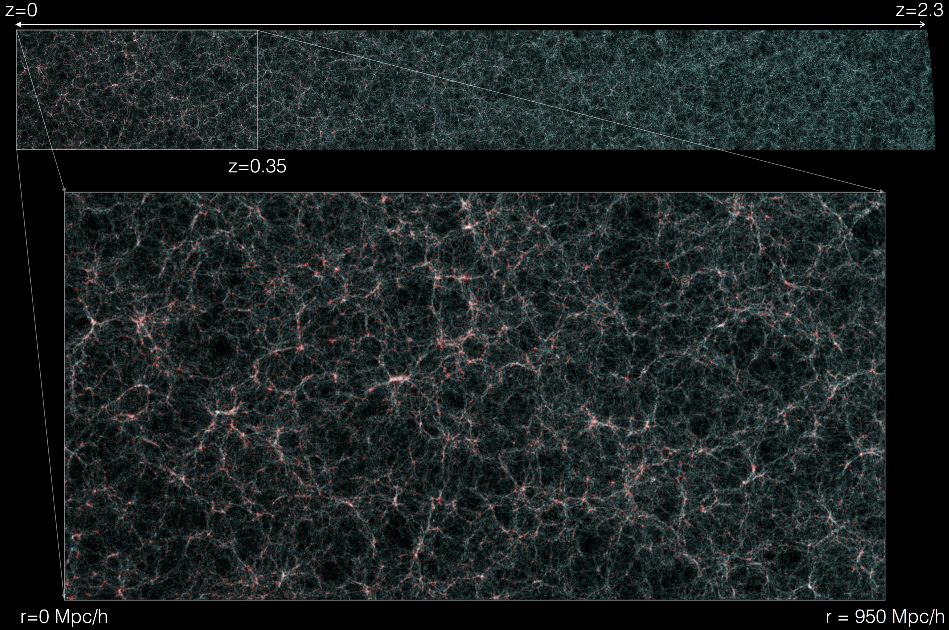
Large N-body simulation code.

- Fast Multipole Method: $O(N)$
- GPU Acceleration (Hybrid Computing)
- Very efficient memory usage per particle
- Asynchronous direct I/O for checkpoints, the light cone data and halo catalogs
- Communication overlaps with computation

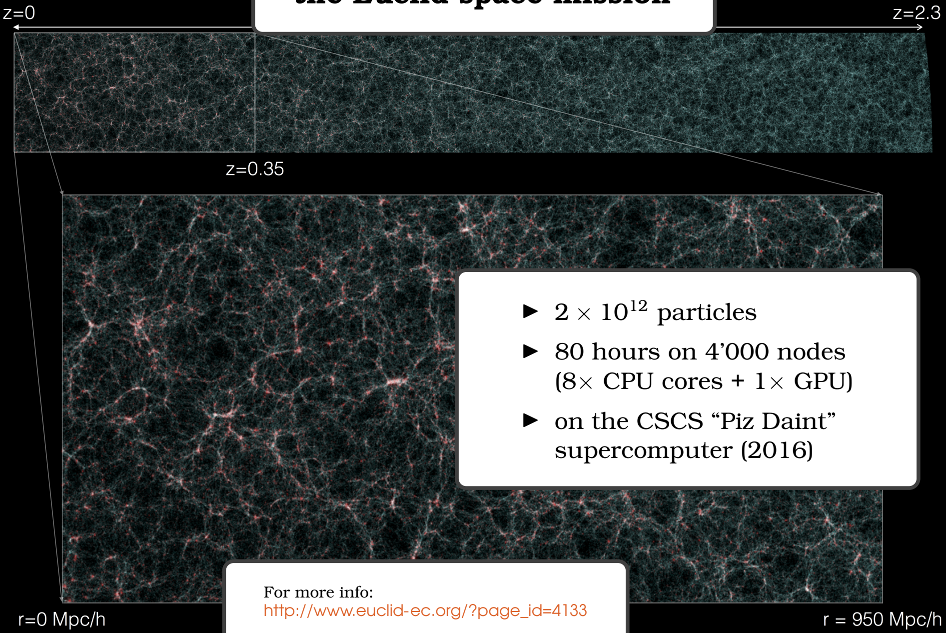
Written by Joachim Stadel, Doug Potter, and collaborators at UZH.

PKDGRAV3: beyond trillion particle cosmological simulations for the next era of galaxy surveys
D. Potter, J. Stadel, R. Teyssier - Computational Astrophysics and Cosmology, 2017

Flagship mock galaxy catalog



Create test dataset for the Euclid space mission



- 2×10^{12} particles
- 80 hours on 4'000 nodes ($8\times$ CPU cores + $1\times$ GPU)
- on the CSCS "Piz Daint" supercomputer (2016)

For more info:
http://www.euclid-ec.org/?page_id=4133

PKDGRAV3: communication

Communication is continuous during the computation.

- one CPU core dedicated to MPI communication
- latency is more important than bandwidth!
- supported by Cray's custom "Aries" interconnect

PKDGRAV3: filesystem I/O

- Light-cone: 240 TB total over 150’000 files.
  - “Final” output, post-processed in further steps of the pipeline
- Checkpoints: 20× 48 TB spread over 20× 28’000 files.
  - *Synchronous:* calculation must stop and wait until file is dumped
  - approx. 2GB per file
  - 1 file per computing thread

PKDGRAV3: filesystem I/O

- Light-cone: 240 TB total over 150’000 files.
  - “Final” output, post-processed in further steps of the pipeline
- Checkpoints: 20× 48 TB spread over 20× 28’000 files.
  - *Synchronous:* calculation must stop and wait until file is dumped
  - approx. 2GB per file
  - 1 file per computing thread

Checkpoints are *needed* to overcome the 24h max runtime policy!

TissueMAPS

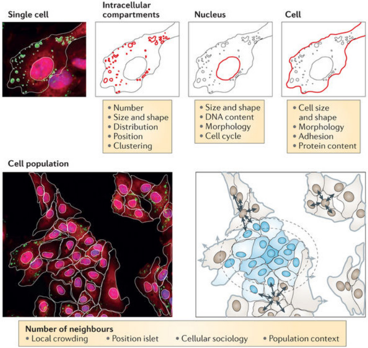
Scalable platform for image analysis of microscopy images.

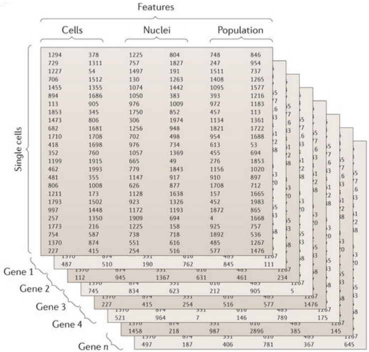
- Developed for image-based cell profiling
- Automated workflow for microscopy image processing
- Browser-based client to explore results and command further analysis

Reference: “Computational Methods and Tools for Reproducible and Scalable Bioimage Analysis”

— M. D. Herrmann, Ph.D. Thesis, Univ. of Zurich (2017).

Image-based Cell Profiling

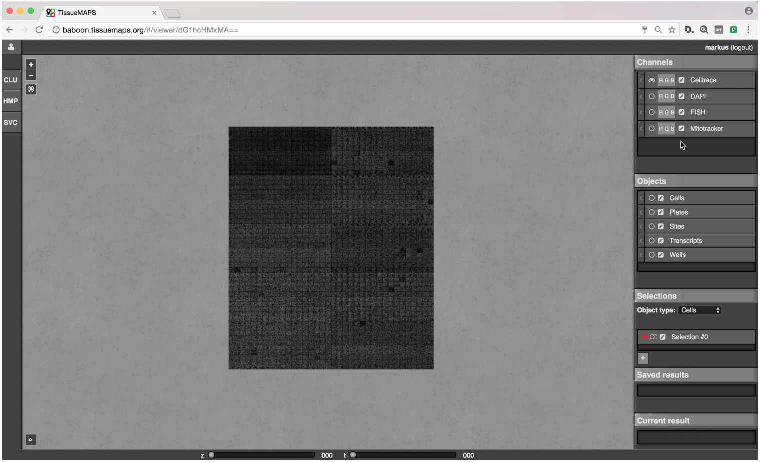




Reference: “Single-cell and multivariate approaches in genetic perturbation screens”

— P. Liberali, B. Snijder, L. Pelkmans, Nat. Rev. Genet., 16:18–32 (2015)

TissueMAPS: Demo of “Transcriptomics” data



https://youtu.be/Qmqf0ysDrx0

TissueMAPS: computational features

Pure “embarassingly parallel” workload:

- many short-lived independent jobs
- *very many* small files
- Sharded DB used to store and process real-time visualization data

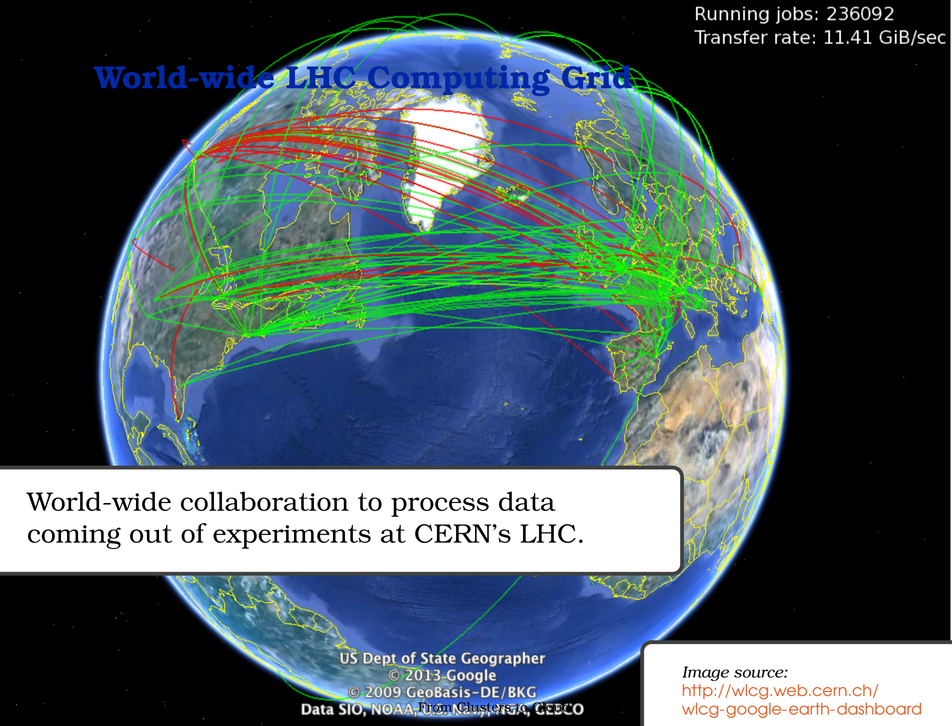
For instance, in the “transcriptomics” data set:

- input microscope images: 352′800 images, a few MBs each
- pyramid tiles: 41′231′720, a few kB each
- DB table for object features: 650M rows

Conflicting requirements!

PKDGRAV3	**TissueMAPS**
Single large MPI job.	Huge swarm of short-lived jobs
Low-latency communication.	No communication across tasks.
Relatively small number of large files.	Huge number of small to tiny files.
Adapted to (high-end) standard cluster computing environment.	Requires setup of custom DB and web-service endpoints.

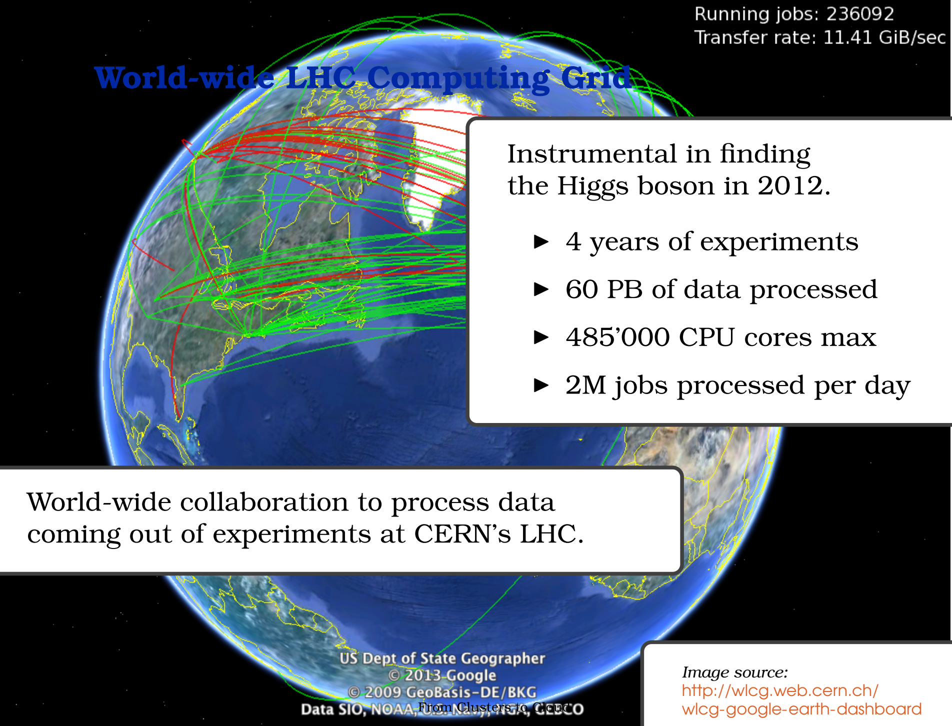
World-wide LHC Computing Grid



World-wide collaboration to process data coming out of experiments at CERN's LHC.

Image source:
http://wlcg.web.cern.ch/wlcg-google-earth-dashboard

World-wide LHC Computing Grid



Instrumental in finding the Higgs boson in 2012.

- 4 years of experiments
- 60 PB of data processed
- 485'000 CPU cores max
- 2M jobs processed per day

World-wide collaboration to process data coming out of experiments at CERN's LHC.

Image source:
http://wlcg.web.cern.ch/wlcg-google-earth-dashboard

WLCG Software Stack

World-wide federation of computing clusters, used to process data coming out of the LHC.

Large software stack to distribute data and process it.

- Mostly custom-built
- Requires installation and maintenance of specific Linux distribution, which is guaranteed to work with the entire software stack.

WLCG Software Stack

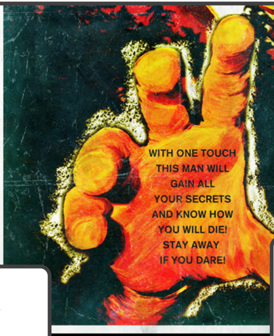
World-wide federation of computing clusters, used to process data coming out of the LHC.

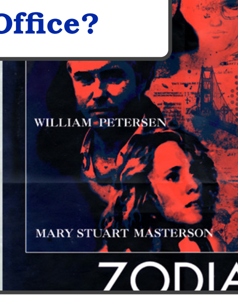
Large software stack to distribute data and process it.

- Mostly custom-built
- Requires installation and maintenance of specific Linux distribution, which is guaranteed to work with the entire software stack.

Force entire infrastructure to run SL6.*x*?

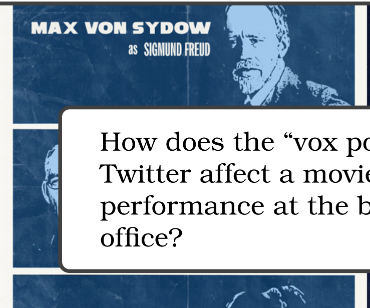
How do Tweets affect the Movie Box Office?

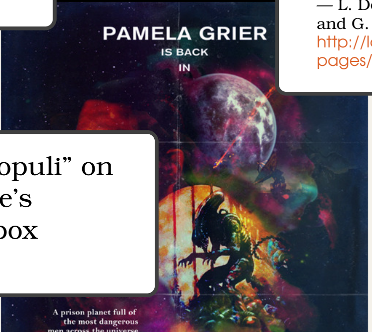




Images Copyright © 2015 Peter Stults
https://www.behance.net/gallery/25965817/What-If-Movie-Posters-Vol-V

"Tweeting to Success? Word of Mouth with Consumer and Firm Learning in the Motion Picture Industry" *(work in progress)*
— L. Deer, P. Chintagunta, and G. S. Crawford,
http://lachlandeer.github.io/pages/research.html





How does the "vox populi" on Twitter affect a movie's performance at the box office?

How do Tweets affect the Movie Box Office?

Try and isolate mechanisms by which Twitter is influencing demand — a computational experiment.

- Get the data:
  - Twitter stream dump
    - 300 movies
    - ± 6 months from release date
  - Box Office performance
- Analyze & Model
  - 85% of Tweets are in the English language — **Filter** out the rest!
  - **Categorize** each Tweet
    - advertisement, buzz, review — each category may affect the dynamics differently
  - **Compute** sentiment score of tweets
  - **Correlate** to Box Office timeseries data

How do Tweets affect the Movie Box Office?

Try and isolate mechanisms by which Twitter is influencing demand — a computational experiment.

- Get the data:
  - Twitter stream dump
    - 300 movies
    - ± 6 months from release date
  - Box Office perfor
- Analyze & Model
  - 85% of Tweets ar
    — **Filter** out the
  - **Categorize** each
    - advertisement, buzz, review
      — each category may affect the dynamics differently
  - **Compute** sentiment score of tweets
  - **Correlate** to Box Office timeseries data

> Classical data science workflow!
>
> Spark/Hadoop are the go-to tools.

How do Tweets affect the Movie Box Office?

Try and isolate mechanisms by which Twitter is influencing demand — a computational experiment.

- Get the data:
  - Twitter stream dump
    - 300 movies
    - ± 6 months from release date
  - Box Office perfor
- Analyze & Model
  - 85% of Tweets ar
    — **Filter** out the
  - **Categorize** each
    - advertisemen
      — each categ
      differently
  - **Compute** sentiment score of tweets
  - **Correlate** to Box Office timeseries data

Classical data science workflow!

Spark/Hadoop are the go-to tools.

Oh, wait. . . Do we have a Hadoop cluster here?

"There's no problem we cannot solve by adding one more layer of indirection."
— *Fundamental Theorem of Software Engineering*

Abstract away the Infrastructure Layer!

Use *Infrastructure-as-a-Service* as a base for providing compute infrastructure.

We can create and setup ad-hoc computing infrastructures:

- *dedicated*: no sharing, exactly the software and policies you want
- *ephemeral*: create when idea comes, dispose when experiment is over

IaaS cloud computing

1. Provision *virtual* resources:
   - virtual machines (VM)
   - block and object storage
   - software-defined networking
2. Network-accessible API for control
   - allows *scripting* the set-up and tear-down of infrastructure
   - "infrastructure as code"

Advantages of "Infrastructure as Code"

1. Reproducibility
   - You can re-create the exact same infrastructure at a later time.
2. Version Control
3. Easy to clone/adapt
4. Readability

Advantages of “Infrastructure as Code”

1. Reproducibility
2. Version Control
   - can easily roll back changes!
   - precise log of how the infrastructure evolved over time
   - ... plus all niceties that we have from coding environments
3. Easy to clone/adapt
4. Readability

Advantages of "Infrastructure as Code"

1. Reproducibility
2. Version Control
3. Easy to clone/adapt
   - It's just text files!
   - Good configuration/deployment tools have a programming languages: functions allow defining *"parametric infrastructure"*
4. Readability

Advantages of “Infrastructure as Code”

1. Reproducibility
2. Version Control
3. Easy to clone/adapt
4. Readability
   - Well-written code counts as documentation of the infrastructure setup

“Software-defined Sysadmin”

However, there are infrastructure setup chores:

- e.g., software installation and configuration
- now you must do these yourself!

ElastiCluster is our solution for automation of basic sysadmin tasks: provisioning and initial setup of a computing infrastructure.

What is ElastiCluster

ElastiCluster provides a **command line tool** and a Python API to **create, set up and resize** computing clusters hosted on IaaS cloud infrastructures.

Main function is to get a compute cluster up and running with a single command.

Effectively, a wrapper around **Ansible** which provides:

- idempotent configuration playbooks
- no-bootstrap remote actions via SSH

ElastiCluster features (1)

Computational clusters supported:

- Batch-queuing systems:
  - SLURM
  - GridEngine
  - Torque+MAUI
  - HTCondor
- Spark / Hadoop 2.x
- Mesos

Distributed storage:

- CephFS
- GlusterFS
- HDFS
- OrangeFS/PVFS

Optional add-ons:

- Ganglia
- JupyterHub
- EasyBuild

(Grayed out items have not been tested in a while...)

ElastiCluster features (2)

Run on multiple clouds:

- Amazon EC2
- Google Compute Engine
- OpenStack
- MS Azure
- … and anything supported by LibCloud

Supports several distros as base OS:

- Debian 7.x (*wheezy*), 8.x *(jessie)*
- Ubuntu 14.04 *(trusty)*, 16.04 *(xenial)*
- CentOS 6.x, 7.x
- Scientific Linux 6.x

ElastiCluster Demo

https://youtu.be/DDm6-QEnNsU

Typical use cases

- **On demand provisioning**
  of computational clusters
- Clusters/servers for **Teaching**
- **Testing** new software or configurations
- **Scaling** a permanent computing infrastructure

On-demand provisioning of compute clusters

TissueMAPS

- Deploy on cloud: compute cluster
  + parallel DB + web front-end

WLCG

- Deploy compute cluster with SL6.*x*

"Twitter Effect on Movies" experiment

- Deploy Spark + JupyterHub

PKDGRAV3

- Still need a real HPC cluster!

On-demand provisioning of compute clusters

:-) TissueMAPS
- Deploy on cloud: compute cluster + parallel DB + web front-end

WLCG
- Deploy compute cluster with SL6.*x*

"Twitter Effect on Movies" experiment
- Deploy Spark + JupyterHub

PKDGRAV3
- Still need a real HPC cluster!

On-demand provisioning of compute clusters

:-) TissueMAPS
- Deploy on cloud: compute cluster + parallel DB + web front-end

:-) WLCG
- Deploy compute cluster with SL6.*x*

"Twitter Effect on Movies" experiment
- Deploy Spark + JupyterHub

PKDGRAV3
- Still need a real HPC cluster!

On-demand provisioning of compute clusters

- :-) TissueMAPS
  - Deploy on cloud: compute cluster + parallel DB + web front-end
- :-) WLCG
  - Deploy compute cluster with SL6.*x*
- :-) "Twitter Effect on Movies" experiment
  - Deploy Spark + JupyterHub
- PKDGRAV3
  - Still need a real HPC cluster!

On-demand provisioning of compute clusters

:-) TissueMAPS

- Deploy on cloud: compute cluster + parallel DB + web front-end

:-) WLCG

- Deploy compute cluster with SL6.*x*

:-) "Twitter Effect on Movies" experiment

- Deploy Spark + JupyterHub

:-(PKDGRAV3

- Still need a real HPC cluster!

Clusters for teaching

Example: JupyterHub+Spark clusters

- for teaching courses (e.g., data science), or
- for short-lived events (e.g., workshops).

Key ingredient is the ability to apply custom Ansible playbooks on top of the standard ones, to make per-event customizations.

Scaling permanent clusters

Example: additional WLCG cluster for ATLAS analysis hosted on SWITCHengines

Processes: Grid Local

Country	Site	CPUs	Load (processes: Grid+local)	Queueing
Switzerland	ATLAS BOINC	98139	7894+6083	1571+4063
	ATLAS BOINC 3	98139	5815+8163	1253+4371
	ATLAS BOINC TEST	644	0+0	0+0
	Bern ce01 (UNIBE-LHEP)	1513	1048+0	156+0
	Bern ce02 (UNIBE-LHEP)	770	624+0	159+0
	Bern ce04 (UNIBE-LHEP>	304	304+0	192+0
	Bern UBELIX T3	4472	385+2822	208+2450
	CSCS BRISI Cray XC40	1500	576+0	154+0
	Geneva (UNIGE-DPNC)	720	168+349	169+0
	Lugano PHOENIX T2 arc>	1920	1526+4040	411+14
	Lugano PHOENIX T2 arc>	2240	2065+3504	391+4
	Lugano PHOENIX T2 arc>	2048	1864+3704	407+1
TOTAL	*12 sites*	*212409*	*22269 + 28665*	*5071 + 10903*

Reference: S. Haug and G. F. Sciacca,

"ATLAS computing on Swiss Cloud SWITCHengines", CHEP 2016

Scaling permanent clusters

Example: additional WLCG cluster for ATLAS analysis hosted on SWITCHengines

> *"A 304 virtual CPU core Slurm cluster was then started with one command on the command line. This process took about one hour. A few post-launch steps were needed before the cluster was production ready. However, a skilled system administrator can setup a 1000 core elastic Slurm cluster on the SWITCHengines within half a day.* ***As a result the cluster becomes a transient or non-critical component. In case of failure one can just start a new one, within the time it would take to get a hard disk exchanged.****"*

Reference: S. Haug and G. F. Sciacca,

"ATLAS computing on Swiss Cloud SWITCHengines", CHEP 2016

An Outlook into the Future (1)

Cloud is a game-changer

- Reliable managed infrastructure
- IT support work moves to software layer
- Levels the playground: small and large centers can all use the same infrastructure

An Outlook into the Future (2)

HPC batch clusters are here to stay

- Docker is the new packaging format (e.g., BIDS)
- Batch systems evolve to support containers (Singularity, Shifter)
- Next step: AWS batch?

An Outlook into the Future (3)

Access to data is *the* problem

- Experimental science *does* data processing
- Instruments are on-site, how do you move data to a (distant) cloud?
- How do you make it accessible to multiple workers?

An Outlook into the Future (4)

Computation moving to frameworks that abstract away infrastructure details.

- We already see this happening, e.g., with Spark, Snakemake, Tensorflow, ...
- More frameworks coming!

An Outlook into the Future (5)

But in the end, it's a people's thing

- IT support moving closer to researchers and away from infrastructure
- Interdisciplinary teams will be key

It's teamwork!

Special thanks go to ...

... to the ElastiCluster fellow devs:
Antonio Messina, Nicolas Bär

... to my colleagues at GC3/S3IT:
Sergio Maffioletti, Tyanko Aleksiev

... to the Scientists who contributed:
Lachlan Deer, David Dreher, Markus D. Herrmann,
Lucas Pelkmans, Doug Potter, Joachim Stadel

Thanks!

(Any questions?)