LSST Processing on AWS Platform

Dino Bektesevic (Univ of Washington)
Sanjay Padhi (AWS)
Todd Miller (Univ of Wisconsin)
Hsin-Fang Chiang, Yusra AlSayyad, Meredith Rawls (LSST)

Petabytes to Science Data Inclusion Revolution
Nov 8, 2019

Background of our PoC project

- After the last Kavli workshop, a team from AWS, LSST, and HTCondor has been assembled to do a Proof of Concept.
- **Goal: Cloud deployment of LSST Data Release Production**
- In LSST Operations, ~annual data releases of image & catalog products. Data processing with all pipelines

  instrument signature removal, calibration, coadd, detection, object characterization, forced photometry, etc.
- Complex workflow with many many jobs.

  In the PoC we've scaled up to a "tract"-size workflow:
  ~7000 input raw images, 27,000 jobs → ~10% of 1 LSST night.

  DRP Y1 ~10 million jobs

HTCondor & *condor_annex*

- Easy to get lots of machines, hard to use them
- Workload (& worker) management system
  - Sometimes machines will crash…
  - Some jobs are bigger than others…
  - Some jobs depend on others…
  - … that's OK, HTCondor takes care of it.
  - Sometimes you need more machines…
  - … that's OK, use *condor_annex* to get them.
    - condor_annex dynamically expands an existing pool
    - this tutorial will use an AWS instance, but it could be your laptop or a dedicated server

HTCondor Annex

condor_annex [**-aws-region** *<region>*] **-setup** *[FROM INSTANCE|[/full/path/to/access/key/file [/full /path/to/secret/key/file]]]*

condor_annex [**-aws-on-demand**] **-annex-name** *<name of the annex>* **-count** *<integer number of instances>* [**-aws-on-demand-***] [**common options**]

condor_annex [**-aws-spot-fleet**] **-annex-name** *<name of the annex>* **-slots** *<integer weight>* [**-aws-spot-fleet-***] [**common options**]

condor_annex -annex-name *<name of the annex>* **-duration** *hours*

condor_annex [**-annex-name** *<name of the annex>*] **-status** [**-classad**]

condor_annex -check-setup

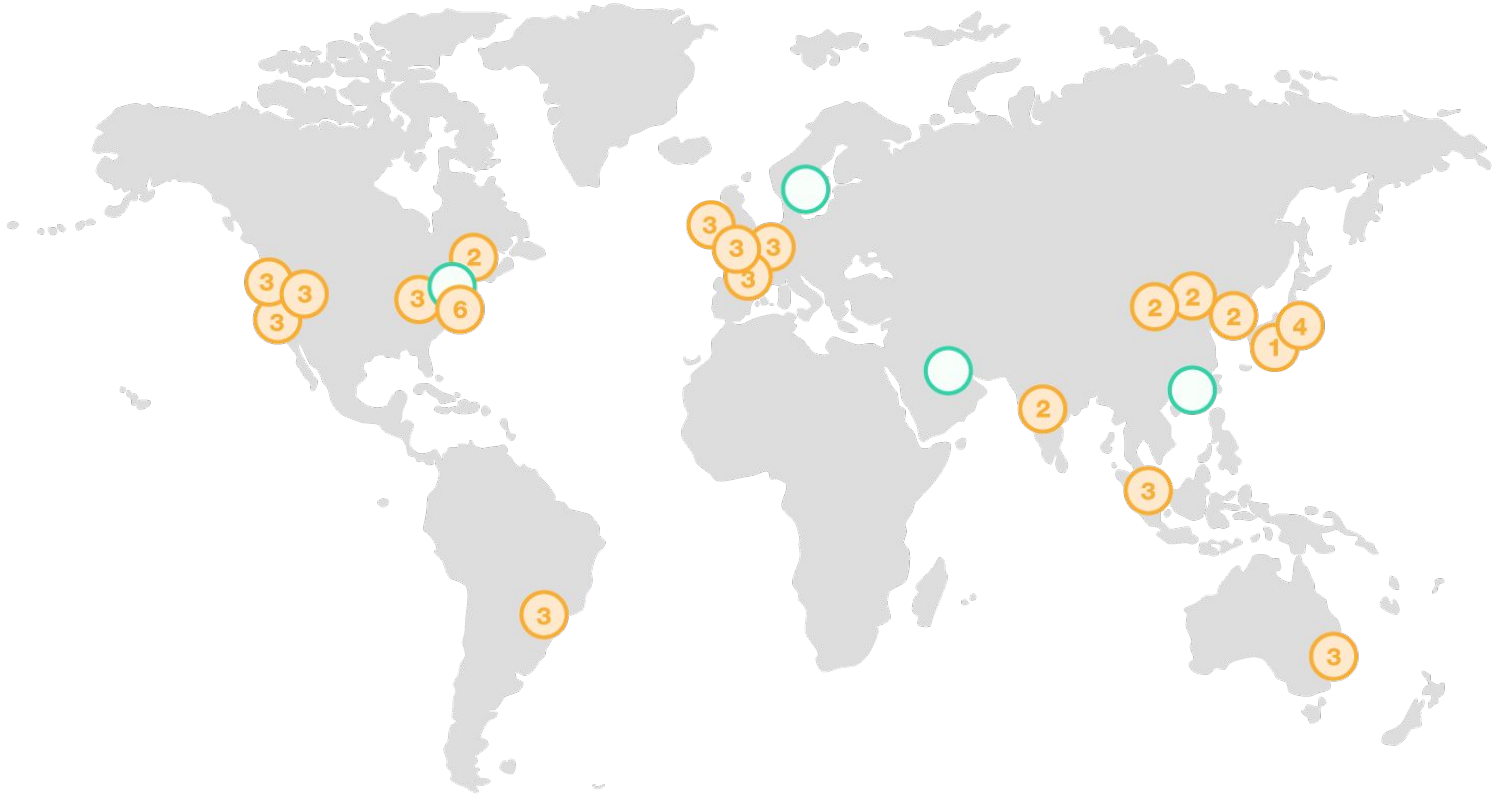
condor_annex <condor_annex options> **status** <condor_status options>

https://htcondor.readthedocs.io/en/stable/man-pages/condor_annex.html

AWS Global Infrastructure

22 Regions – **69** Availability Zones – **155** Edge Locations

11 Regional Edge Caches in **65** cities across **29** countries



The AWS Secret Region is designed and built to meet the regulatory and compliance requirements of DOD, IC, etc.

Evolution in Compute Services

Virtual Server Hosting, Container management, and Serverless Computing

Amazon EC2

Provides resizable cloud-based compute capacity in the form of EC2 instances, which are equivalent to virtual servers

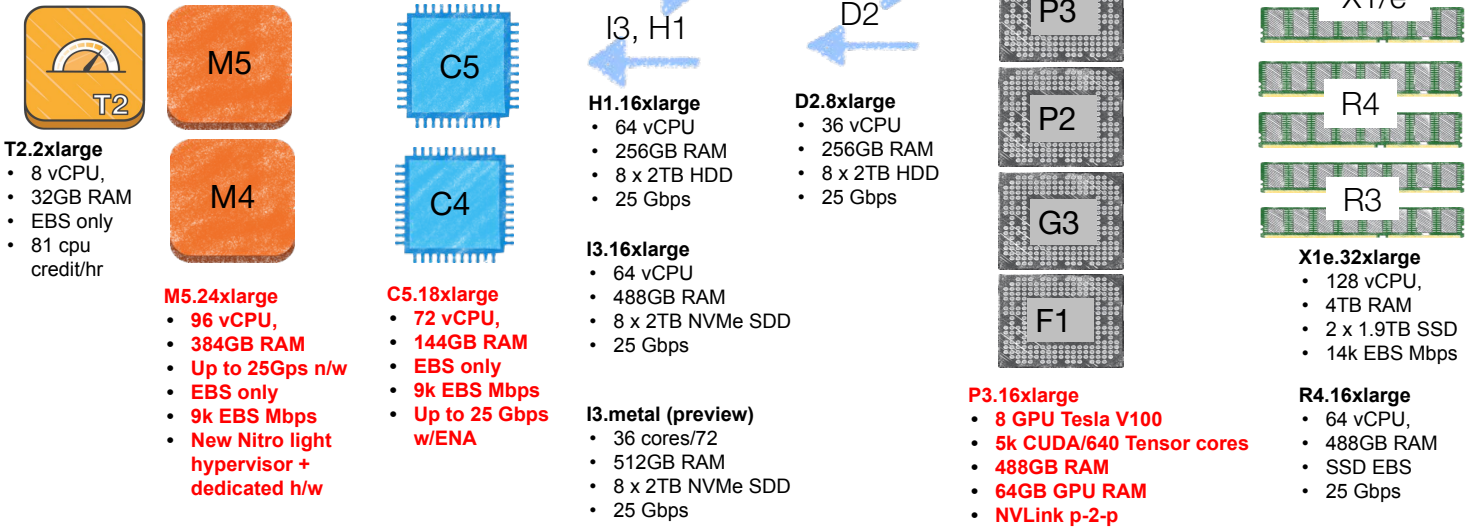
Amazon EC2 Container Service

A highly scalable, high performance container management service

AWS Lambda

Run code without thinking about servers.

Heterogeneity in Compute Resource Instance Types & CPU

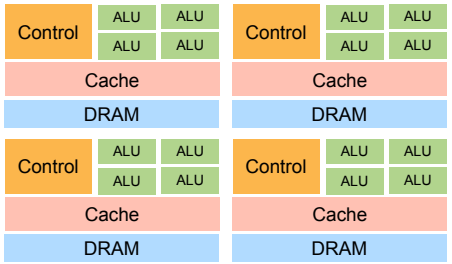


Selection of different Intel Xeon processors

- 2.3/2.4 GHz Intel Broadwell/Haswell CPUs: M4, I3, H1, D2, G3, P3/2 instance types
- 2.9 GHz Intel Haswell CPUs: C4
- 2.5 GHz Intel Platinum CPUs: w/AVX-512 instruction set: M5
- 3.0 GHz Intel Platinum CPUs: w/AVX-512 instruction &set Turbo up to 3.5Ghz: C5

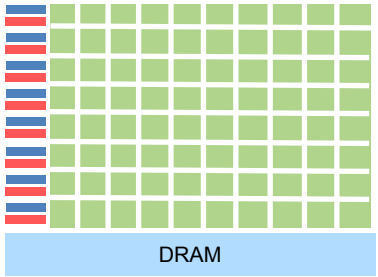
CPUs vs GPUs vs FPGA for Compute

CPU



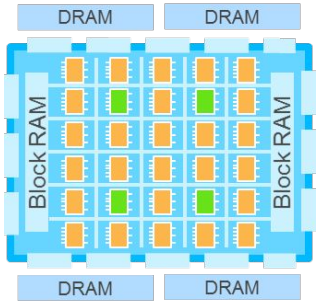
- 10s-100s of processing cores
- Pre-defined instruction set & datapath widths
- Optimized for general-purpose computing

GPU



- 1,000s of processing cores
- Pre-defined instruction set and data path widths
- Highly effective at parallel execution

FPGA



- Millions of programmable digital logic cells
- No predefined instruction set or datapath widths
- Hardware timed execution

Relational Database Service

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Amazon RDS is available on several database instance types - optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server. You can use the AWS Database Migration Service to easily migrate or replicate your existing databases to Amazon RDS.

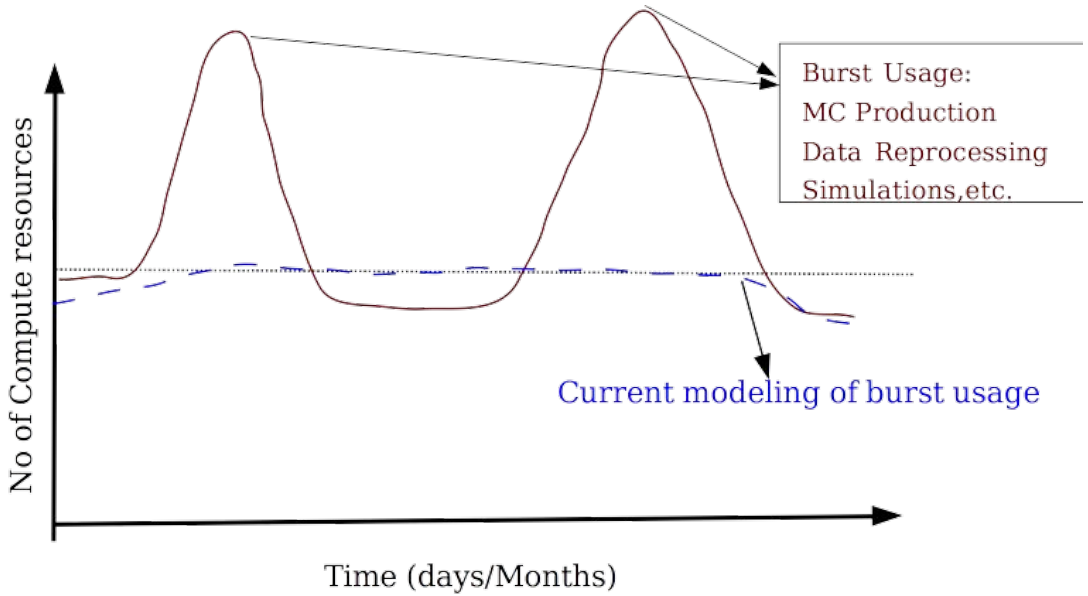
Understanding Amazon Relational Database Service (1:53)

Amazon RDS database engines

Amazon Aurora | PostgreSQL | MySQL | MariaDB | ORACLE | Microsoft SQL Server

https://aws.amazon.com/rds/

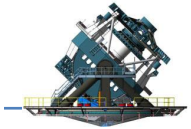
Elasticity is an essential element



Finite number of resources by the experiments (Compute as well as Storage)

"Burst" usage is modeled using delays (~months) due to (re)processing capabilities

Elasticity in the system is really essential



Use Cases: Elasticity

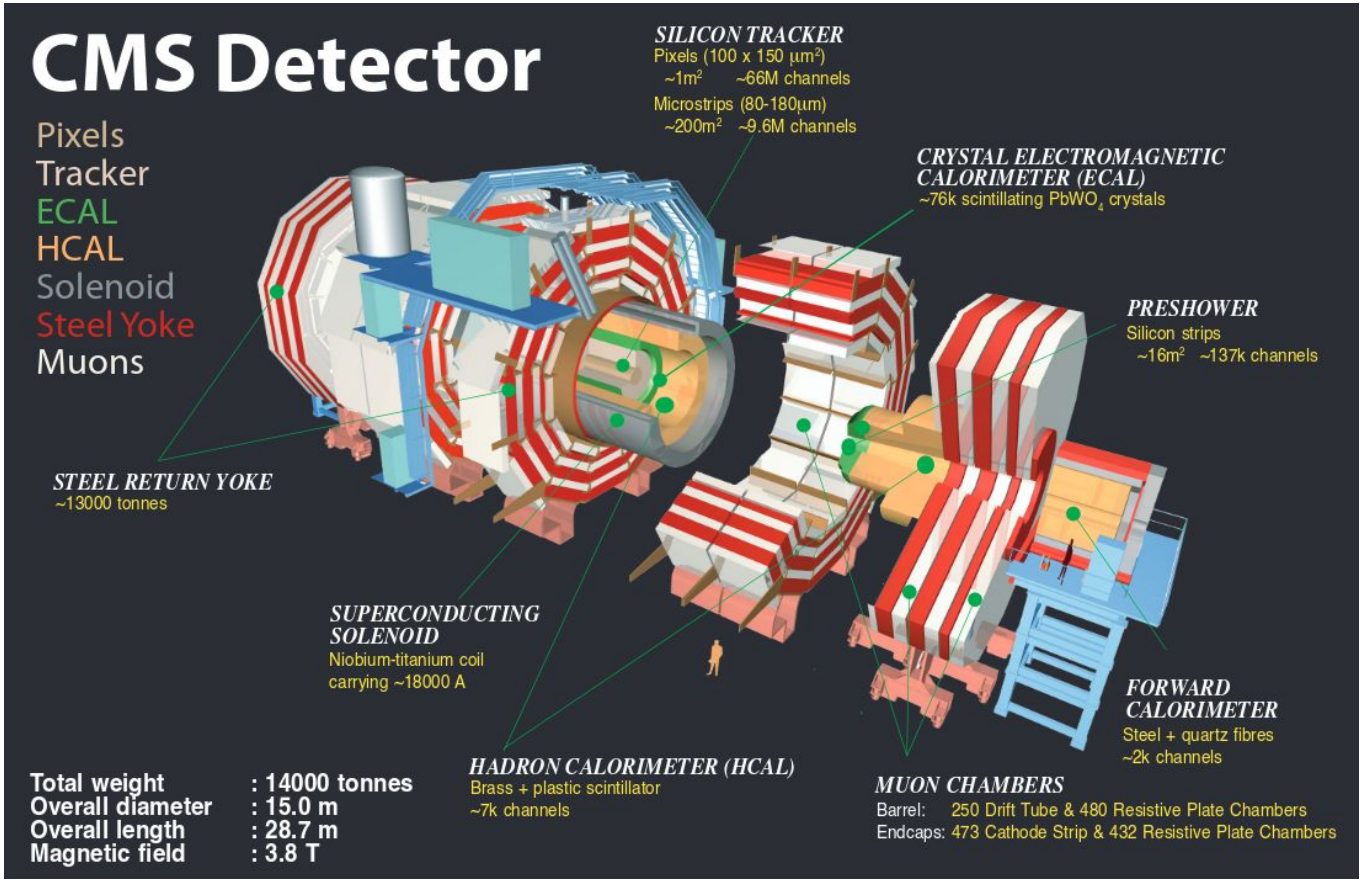
- **80 Million electronic channels**
  x 4 bytes
  x 40MHz

  ~ **10 Petabytes/sec** of information
  x 1/1000 zero-suppression
  x 1/100,000 online event filtering

  ~ 100-1000 Megabytes/sec raw data to tape
  1 to 10 Petabytes of raw data per year written to tape, not counting simulations.

- 2000 Scientists (1200 Ph.D. in physics)
  - ~ 180 Institutions
  - ~ 40 countries

- 12,500 tons, 21m long, 16m diameter

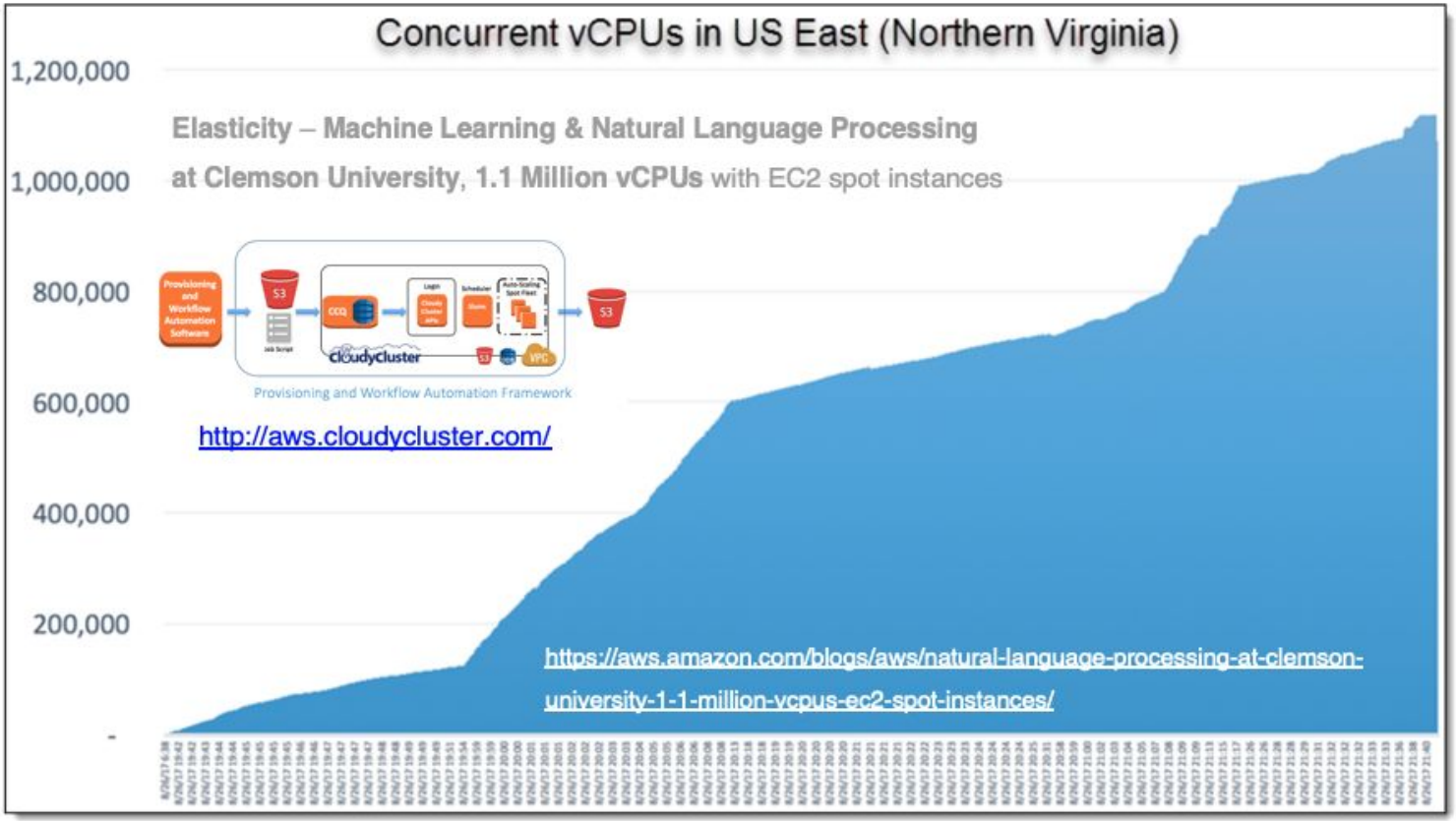


Elasticity in Computing: On demand auto-expansion to AWS

~60,000 slots using AWS spot instances. A factor of 5 larger than Fermilab capacity!

https://aws.amazon.com/blogs/aws/experiment-that-discovered-the-higgs-boson-uses-aws-to-probe-nature/

Elasticity in Computing



Children's Hospital of Philadelphia and Edico Genome

Guinness World Records title for *fastest time to analyze 1,000 human genomes*

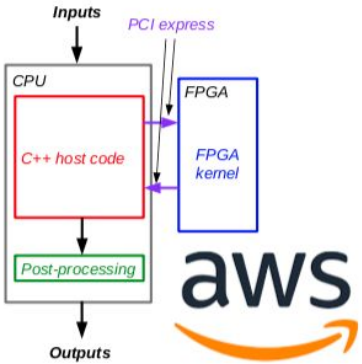
The **Amazon EC2 F1 instances**, with Xilinx Virtex UltraScale+ field programmable gate arrays (FPGAs), were used for 1,000 diverse pediatric genomes.

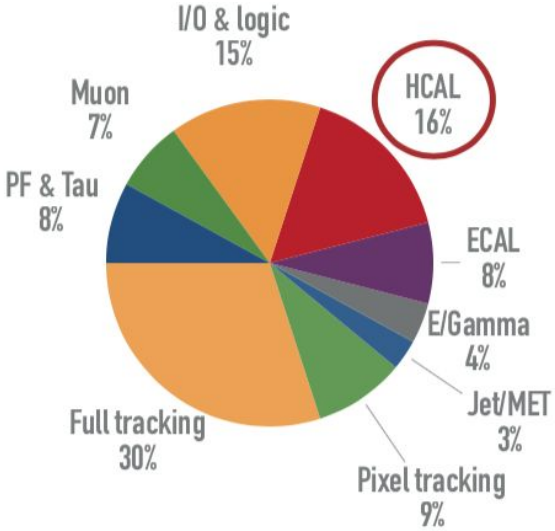
The study was completed in 2 hours and twenty-five minutes..

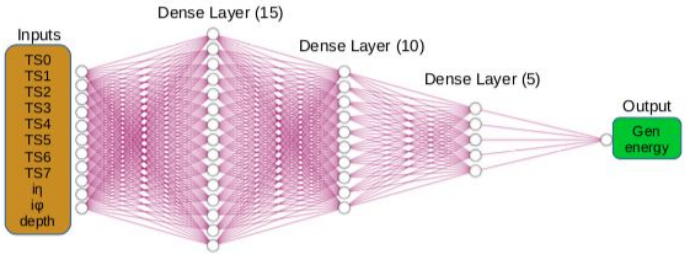
https://www.prnewswire.com/news-releases/childrens-hospital-of-philadelphia-and-edico-genome-achieve-fastest-ever-analysis-of-1000-genomes-300540026.html

Machine Learning using FPGAs

- HCAL local reconstruction contributes significantly to HLT compute time
- ML+FPGA as co-processor can reduce HCAL local reco. compute time by up to ×4
- Tested using AWS FPGAs







Workflows using HTCondor Annex

Features (examples):

Supports CPUs, GPUs and FPGAs

Checkpoint and Migration

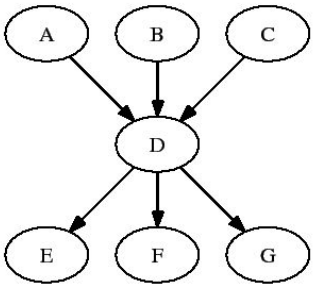
Remote System Calls

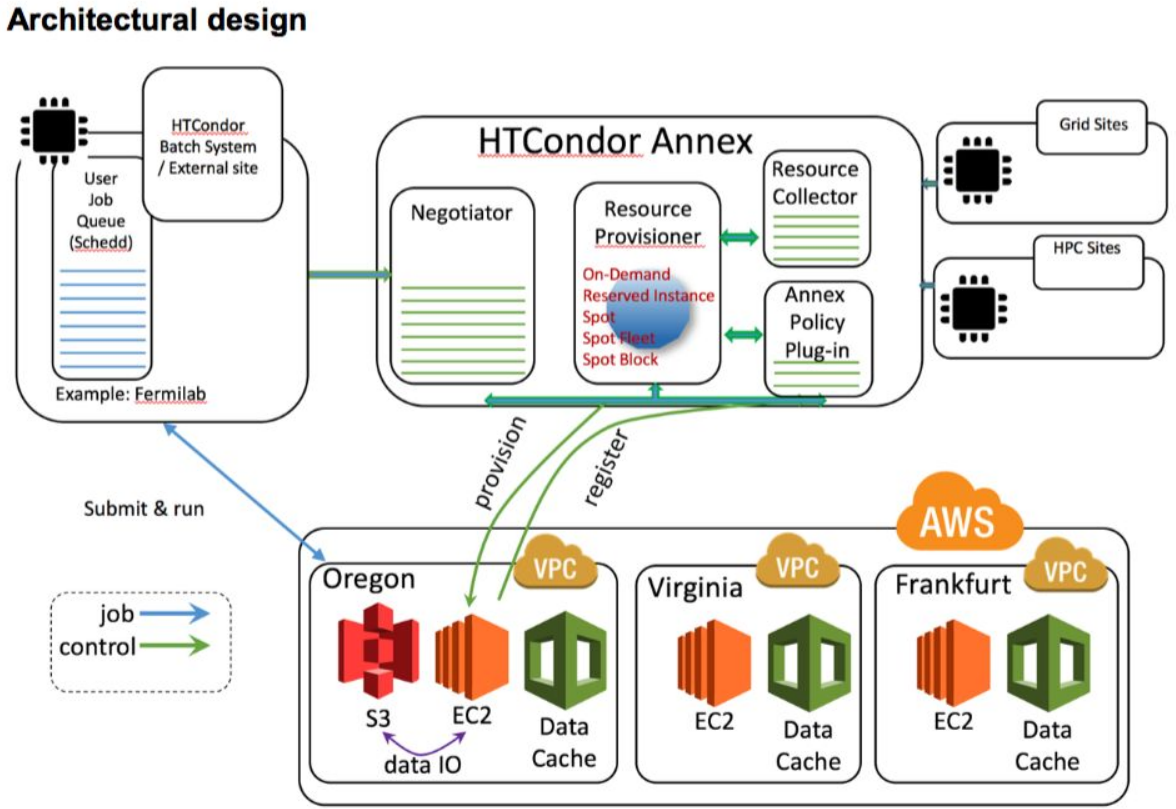
Matchmaking mechanisms

Monitoring and Controls

(Optional) File In/Out to S3

DAGs

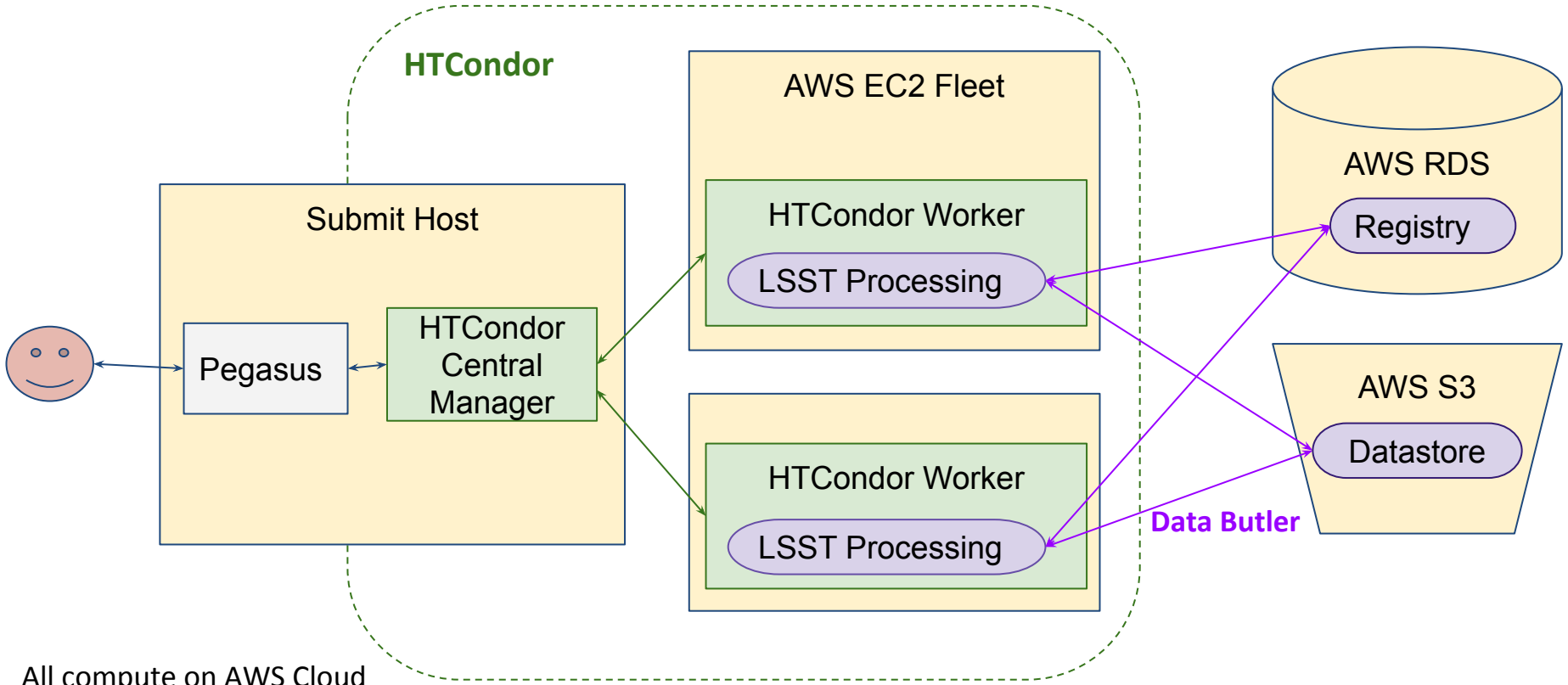




LSST Data Processing

- LSST Pipelines use "Data Butler" for all data access
- Butler hides data formats and locations from pipelines. Internal DB registry to track datasets.
- AWS backend:
  - S3 datastore -- where science files are stored
  - RDS PostgreSQL registry
- Each processing job talks to S3 & RDS

PoC Architecture View

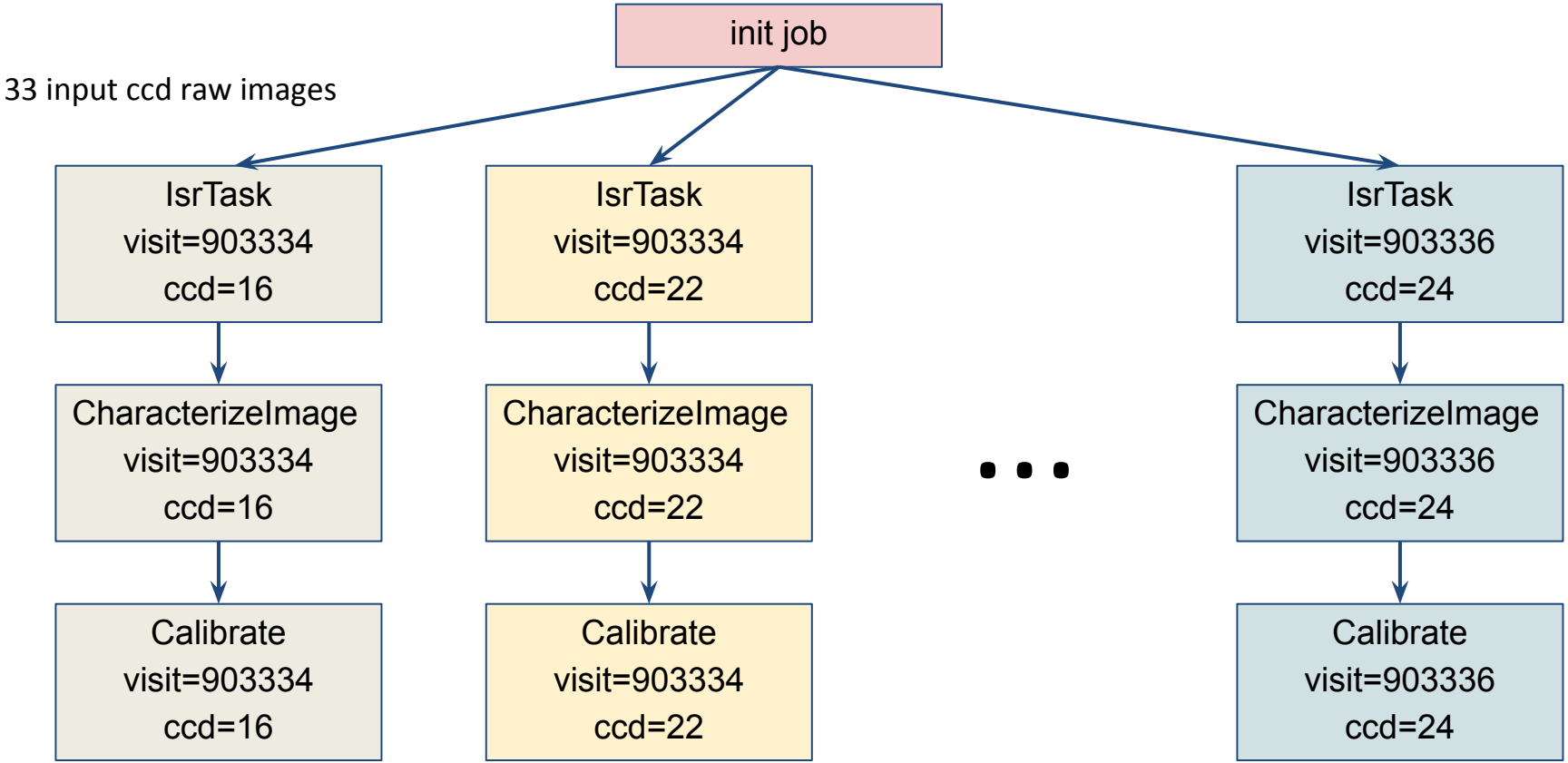


All compute on AWS Cloud
This does not represent LSST production baseline design

In this tutorial, you will...

- Launch an EC2 instance
- Setup your credentials on your instance
- Run an example instrument signature removal (ISR) task
- Find the outputs of your ISR job
- Submit a workflow of jobs
- Add Annex workers to the HTCondor pool
- See your workflow jobs run

tutorial demo workflow



AWS credentials

- On your piece of paper, you have
  - Login name: aws-lsst01
  - Password
  - AWS Access Key ID: AKI….
  - AWS Secret Access Key

Input them when you run setUpCredentials.sh

- In Step 1.9, you will create your SSH key pair and download a *yourName.pem* file

Step-by-step tutorials at

https://confluence.lsstcorp.org/display/DM/Tutorials+at+the+Kavli+workshop

&

Please talk to us in the hack sessions!