

Distributed Computing Intro

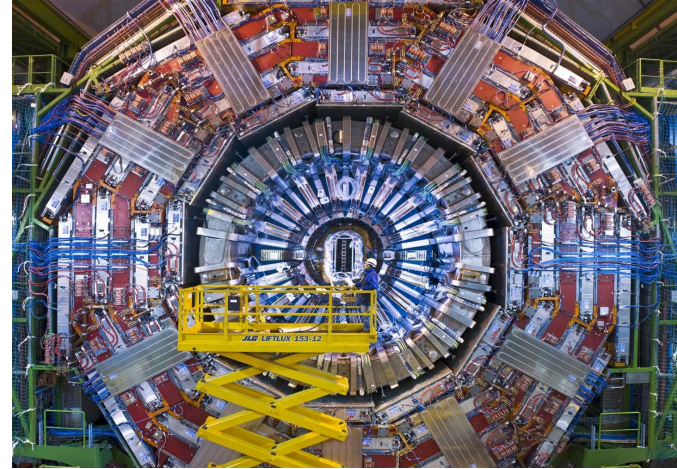
28 June 2017

Analysis in high energy physics today

In physics, we often need to process some detector data, stored in file(s), run some algorithm(s), and get a result. Of course we use computers for that.

Modern experiments have billions of events, and petabytes of data (1 PB = 1,000 TB = 1,000,000 GB)

It's impossible for a single person or group to analyze all of this data



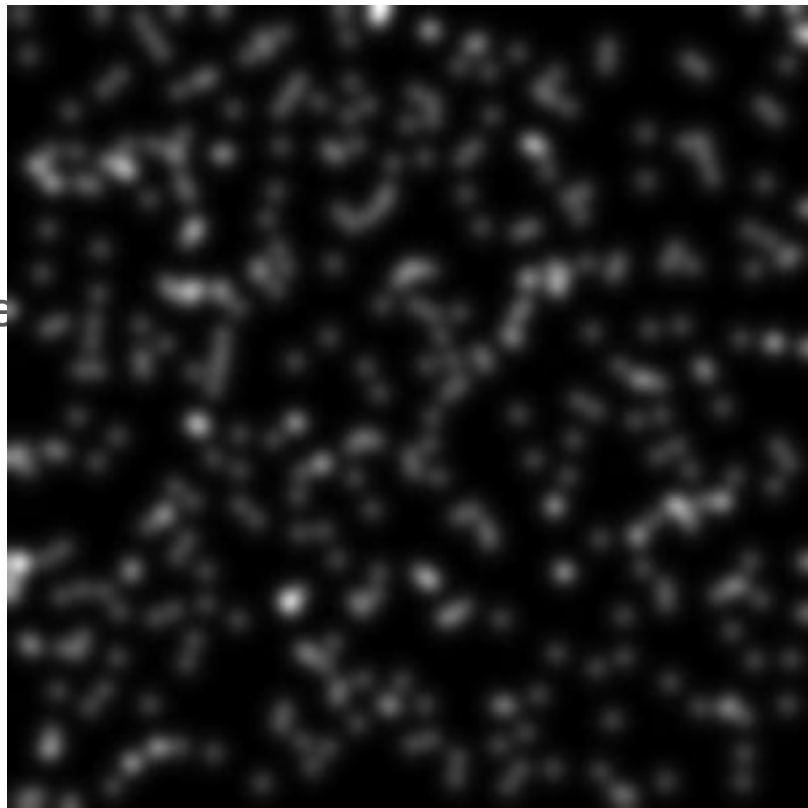
A practical example

Consider this image: how many dots are in it?

You might also ask, what counts as a dot?

Doing this by eye takes a while, but algorithms are great at this kind of thing!

How many images could you analyze in, say, one hour? Three hours? 24 hours?



Getting Things Done in a Reasonable Time

How can we solve our problem? We need hundreds of thousands (millions?) of inputs processed within a short time.

Distributed Computing!

What Is Distributed Computing?

Collections of computers around the world linked together that allow users to run jobs remotely, and can be used to work on the same problem or analysis. Sometimes also called “grids” or “the grid.”

Fermilab has about 25,000 CPUs available to its experiments, plus another ~20K primarily intended for the CMS experiment at the Large Hadron Collider

Additional resources are available around the world via the Open Science Grid, experiment allocations on supercomputers, or commercial clouds (Amazon EC2, Google CE, Microsoft Azure, etc.)

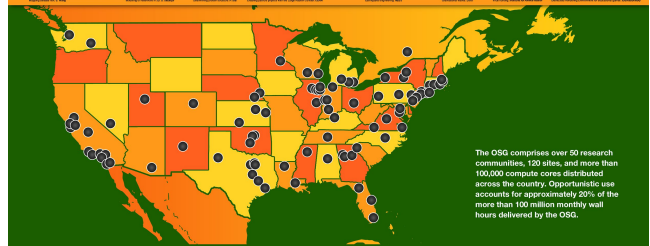


Open Science Grid

Delivering the Power of High Throughput Computing

What is Open Science Grid?

The OSG is a national, distributed computing grid designed to meet the needs of research and academic communities at all scales. In addition to providing governance and a common dependable infrastructure, the OSG distributes proven software solutions, offers support and consulting services, facilitates computer resource sharing, and organizes educational events.



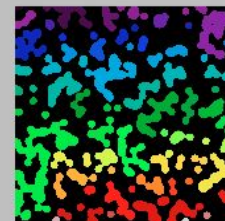
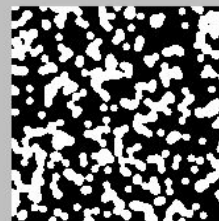
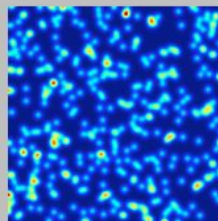
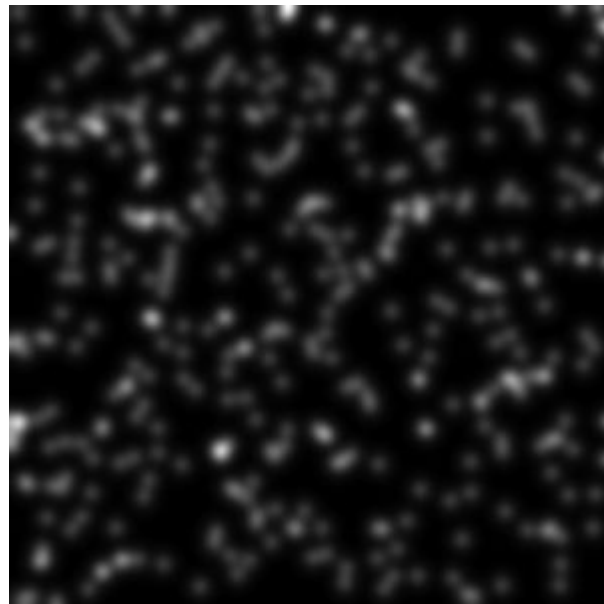
Today: Image processing

We are going to send some jobs that use distributed computing resources. We're going to count dots in images like these.

Note that not every white spot is a "dot." We have a threshold defined already for what will count.

This is in fact pretty much what happens in a physics analysis:

A bunch of information that isn't human-readable gets processed by some algorithms. There's also a lot of random noise (stuff we don't want) that has to be filtered.



First: a single image

Let's look at what we will run on the grid.

In the terminal window that is open for you, type *python count_clusters.py*

The output will will show you an image, and the dot count.

Basically the same thing will happen in your grid run, only you won't see the picture itself and you'll process more images.

Job Submission

We have a set of tools that most experiments here use to submit (shields the user from the complexity of getting to various sites around the world)

On your computers you will find a script called `submit_jobs.sh`

Execute it by typing `./submit_jobs.sh` in the open terminal window.

It will ask you to type your first name, and an image set to process (there are 10 sets, numbered from 0 to 9).

One of the outputs of `submit_jobs.sh` will be a job ID number of the form nnnnnnnn.n@fifebatchN.fnal.gov Keep track of the job ID number; you'll need it for fetching the job output files.

Today's Task

Make sure you submit at least one job. Feel free to submit additional jobs with different image sets (run the script again and put a different number)

Answer the following questions about each job:

What was the job ID number?

Where did your job run (site and hostname; they are in the .out file)

What was the name of the image file set that you processed? How many dots are in each image?

How long did your job take (the start and end times are in the .out file)?

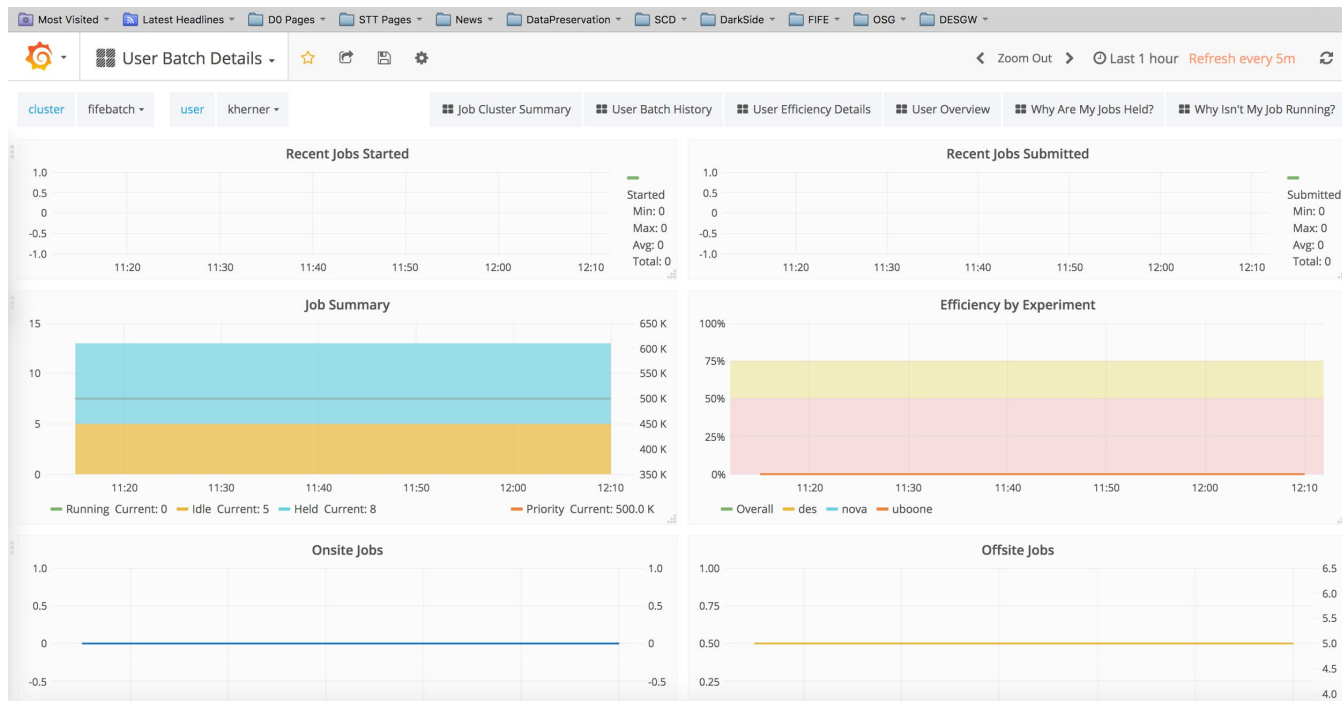
The answers to all questions are in the file ending with **.out**.

To obtain the output file type `./fetch_logs.sh your_job_id some_dir` To read the file: `cat some_dir/*.out`

Job Monitoring

It's very important to have a robust, fast system to monitor job progress

Lots of people are trying to use the system, so our jobs may have to wait in line (a queue) for a bit



Creating this infrastructure requires dedicated work-- there's room to contribute to physics even if you are not a physicist!

To see your job(s)

<https://fifemon.fnal.gov/monitor/dashboard/db/user-batch-details?var-cluster=fifebatch&var-user=kherner&from=now-1h&to=now>

Scroll to the bottom and find your job ID (click on it for more details)

After it's finished you can find it at

<https://fifemon.fnal.gov/monitor/dashboard/db/user-batch-history?from=now-1h&to=now&var-cluster=fifebatch&var-user=kherner>

While you're waiting...

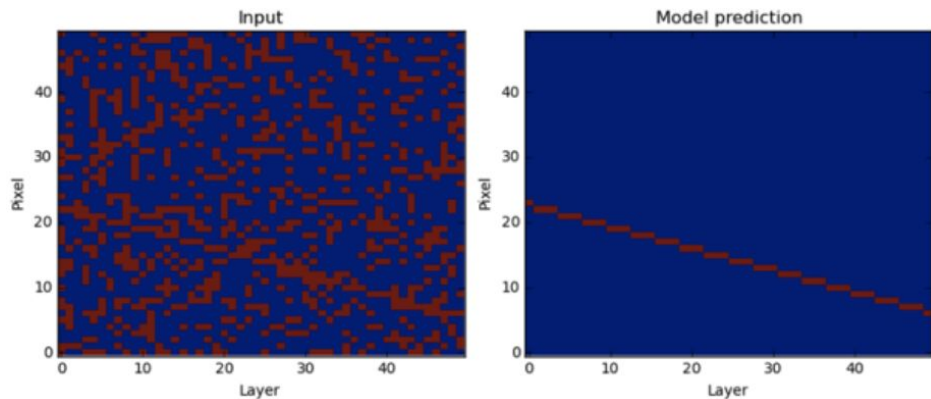
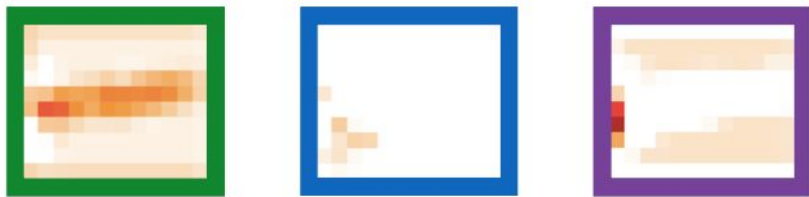
Jobs usually start fairly quickly, but while you're waiting, you can:

Send additional jobs with the other image sets! You do not need to wait for the first job to finish.

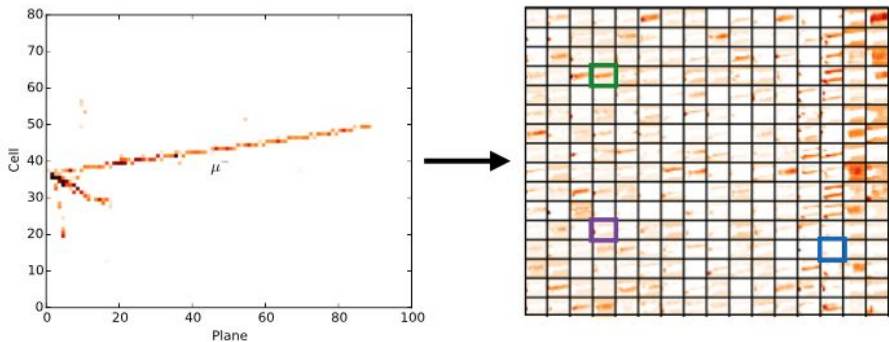
Look at the `run_HEHA_yourname.sh` script and try to understand the detailed steps. Feel free to ask questions!

What others are doing

Current physics experiments are using similar techniques to analyze their detector data and find objects of interest in noisy environments:



Steve Farrell, HEP.TrkX Project



NOvA Collaboration