The Landscape of Academic Research Computing

Julia Gray
julia.ann.gray@gmail.com

Original slides by:
Jaehoon Yu <jaehoonyu1@gmail.com>
Rob Quick <rquick@iu.edu>

Some Slides Contributed by the University of Wisconsin HTCondor Team and Scot Kronenfeld

Julia Gray

Senior Data Scientist at Procter & Gamble

- About Me: Croatian American living in Geneva
- Time with P&G: 4 years
- Previous Roles: DS at a microfinance NGO, experimental particle physicist at CERN
- Passions/Interests/Hobbies: Traveling, running, yoga, skiing, reading, volunteering, drumming
- Favorite Things: My husband, kids, cats, & coffee

OSG Consortium

- Founded in 2005
  - DOSAR one of the founding mebers of OSG consortium

Horst Severini

Pat Skubic

Follow Along at:

https://osg-htc.org/dosar/ASP2024/ASP2024_Materials/

DOSAR: Distributed Organization for Scientific & Academic Research

Goals for this session

- Define Local, Clustered, High Throughput Computing (HTC), High Performance Computing (HPC), and Cloud Computing (XaaS)
- Shared, Allocated, and Purchased
- What is HTCondor? And why are we using it in this Summer School?

Overview of day

- Lectures alternating with exercises
  - Emphasis on lots of exercises
  - Hopefully overcome PowerPoint fatigue

Open Science Grid

RESEARCH DATA ALLIANCE U.S.

Some thoughts on the exercises

- It's okay to move ahead on exercises if you have time
- It's okay to take longer on them if you need to
- If you move along quickly, try the "On Your Own" sections and "Challenges"

Most important!

- Please ask questions!

  …during the lectures

  …during the exercises

  …during the breaks

  …during the meals

  …over dinner

  …via email after we depart

  If I don't know, we'll find the right person to answer your question.

The setup: You have a problem

- Your science computing is complex!
  - Monte Carlo, image analysis, genetic algorithm, simulation…
- It will take a year to get the results on your laptop, but the conference is in a month.
- What do you do?

Open Science Grid

RESEARCH DATA ALLIANCE U.S.

Option 1: Wait a year

Option 2: Local Clustered Computing

- Easy access to additional nodes
- Local support for porting to environment (maybe)
- Often a single type of resource
- Often running at capacity

Option 3: Use a "supercomputer" aka High Performance Computing (HPC)

- "Clearly, I need the best, fastest computer to help me out"
- Maybe you do…
  - Do you have a highly parallel program?
    - i.e. individual modules must communicate
  - Do you require the fastest network/disk/memory?
- Are you willing to:
  - Port your code to a special environment?
  - Request and wait for an allocation?

Open Science Grid

RESEARCH DATA ALLIANCE U.S.

Option 4: Use lots of commodity computers

- Instead of the fastest computer, lots of individual computers
- May not be fastest network/disk/memory, but you have a lot of them
- Job can be broken down into separate, independent pieces
  - If I give you more computers, you run more jobs
  - You care more about total quantity of results than instantaneous speed of computation
- This is **high-throughput computing**

Option 5: Buy (or Borrow) some computing from a Cloud Provider

- Unlimited resources (if you can afford them)
- Full administrative access to OS of the resources you ‘buy’
- Specialized VM images reducing effort in porting
- XaaS Business Model

These are All Valid Options

- Remember the problem you have one month to publish results for your conference
  - Option 1: You will miss your deadline
  - Option 2: You might miss your deadline – But if your lucky you'll make it (or if you know the admin)
  - Option 3: If you have parallelized code and can get an allocation you have a good chance
  - Option 4: If you can serialize your workflow you have a good chance
  - Option 5: You can meet your deadline for a price. Though some efforts are underway to enable academic clouds

Computing Infrastructures

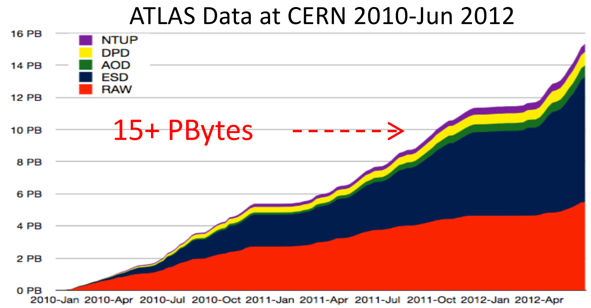
- Local Laptop/Desktop – Short jobs with small data
- Local Cluster – Larger jobs and larger data but subject to availability
- HPC – Prime performance with parallelized code
- HTC – Sustained computing over a long period for serialized
- Cloud – Need deeper permission on an OS and have deeper pockets

Why focus on high-throughput computing? (HTC)

- An approach to distributed computing that focuses on long-term throughput, not instantaneous computing power
  - We don't care about operations per second
  - We care about operations per year
- Implications:
  - Focus on reliability
  - Use all available resources
    - Any Linux based machine can participate

The Problem

- Detectors are complicated and large ➔ Need a large number of collaborators
  - They are scattered all over the world!
  - How do we get them communicate quickly and efficiently?
  - How do we leverage collaborators' capabilities?
  - How do we efficiently utilize all the computing resources?
- Data size is large >>10 PB per year for raw data only
  - Entire data set 15+PB on discovery
  - Where and how to store the large amount of data?
  - How do we allow collaborators scattered all over the world to access data in an efficient fashion?

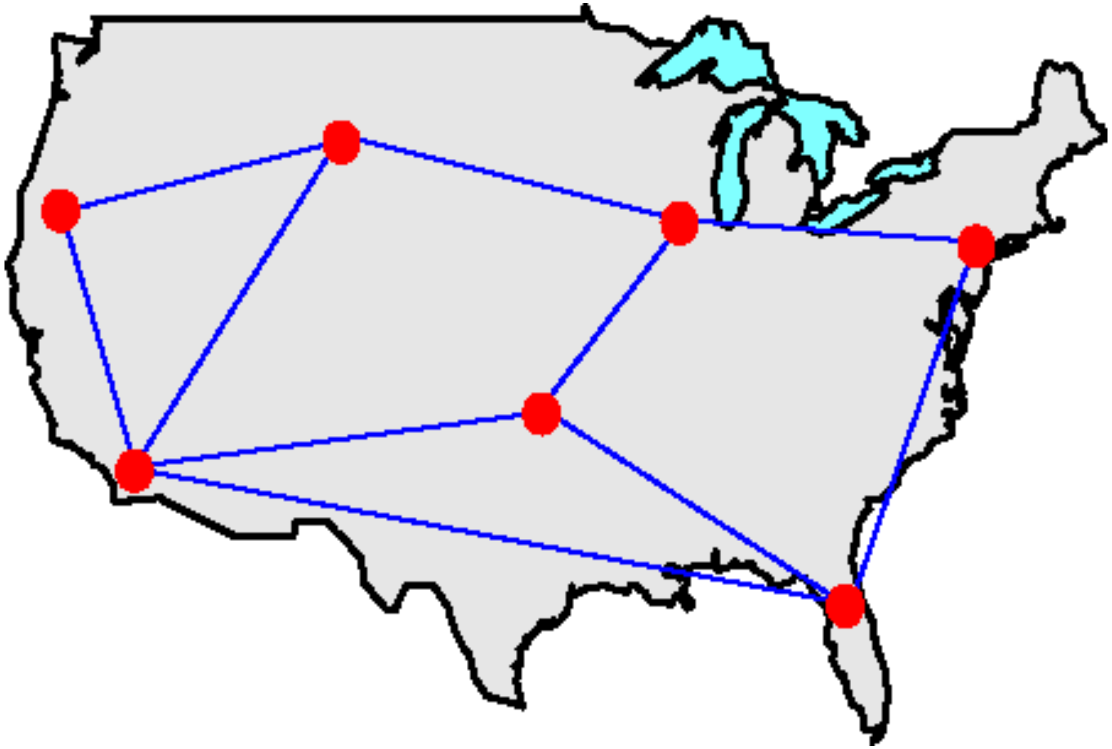


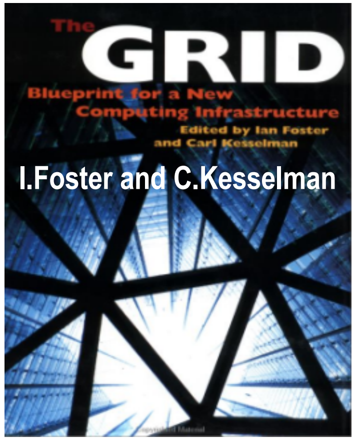
The Problem, cont'd

- How do we allow people's analysis jobs to access data and make progress rapidly and securely?
  - What is the most efficient way to get jobs' requirements matched with resources?
  - Should jobs go to data or data go to jobs?
  - What level of security should there be?
- How do we allow experiments to reconstruct data and generate the large amount of simulated events quickly?
  - How do we garner the necessary compute and storage resources effectively and efficiently?
  - What network capabilities do we need in the world?
- How do we get people to analyze at their desktops?

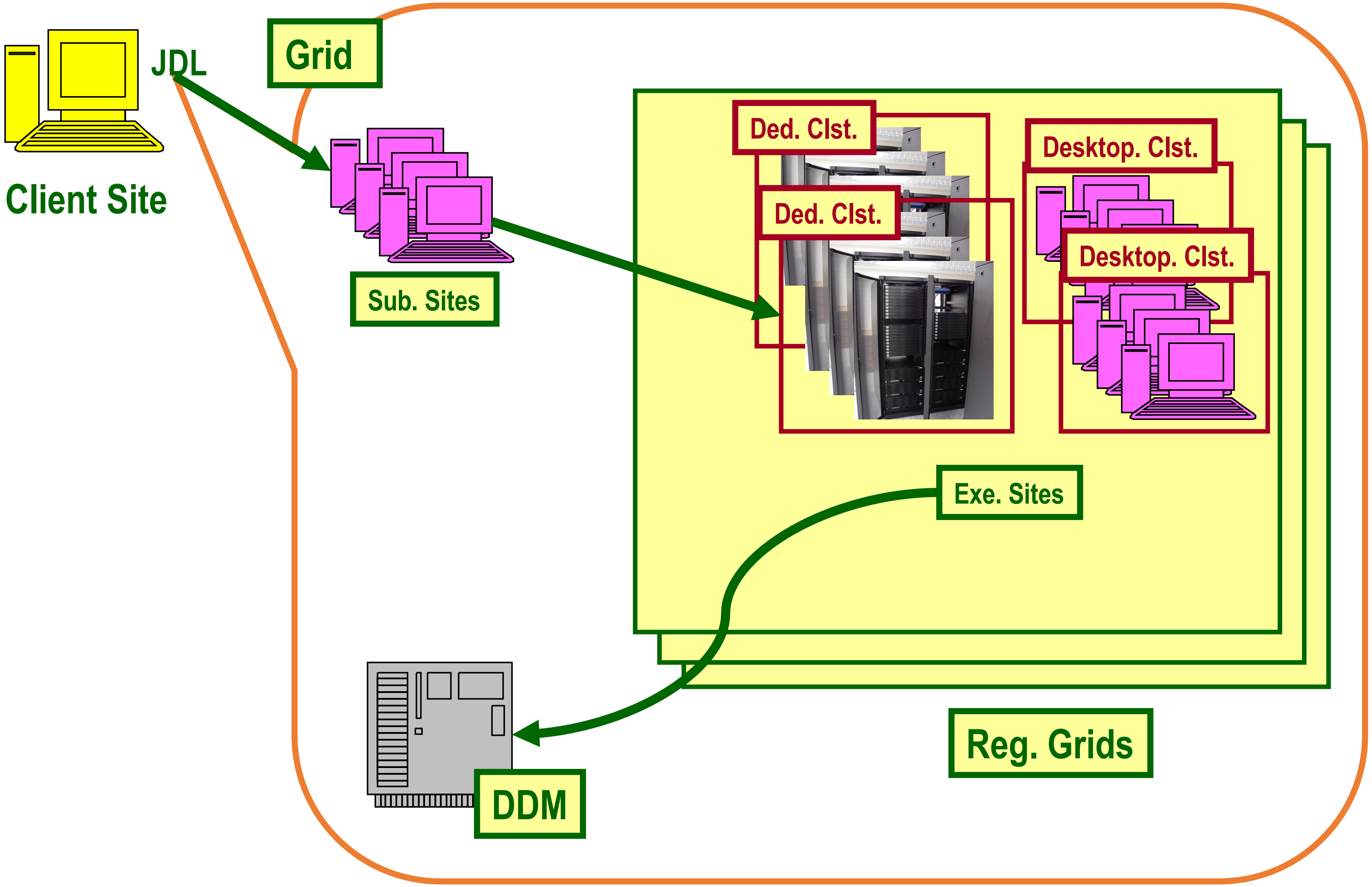
What is a Computing Grid?

- Grid, the definition: Geographically distributed computing resources configured for a coordinated use
- Physical resources & good network provide hardware capability
- The "Middleware" software ties them together

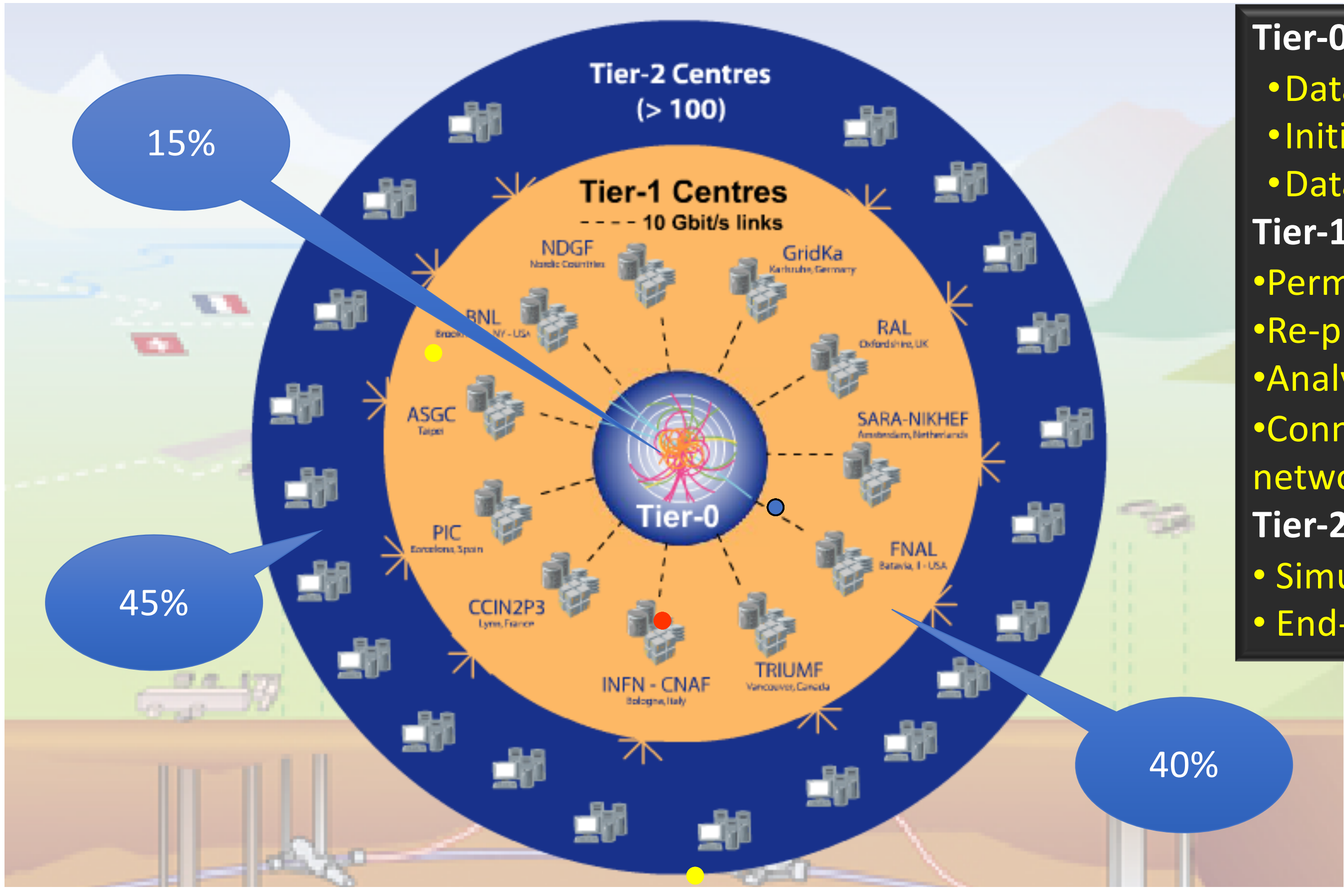




How does a computing Grid work?



Implemented ATLAS Grid Structure



Tier-0 (CERN): (15%)
- Data recording
- Initial data reconstruction
- Data distribution

Tier-1 (11 centres): (40%)
- Permanent storage
- Re-processing
- Analysis
- Connected by direct 10 Gb/s network links

Tier-2 (~200 centres): (45%)
- Simulation
- End-user analysis

How to look for rare particles?

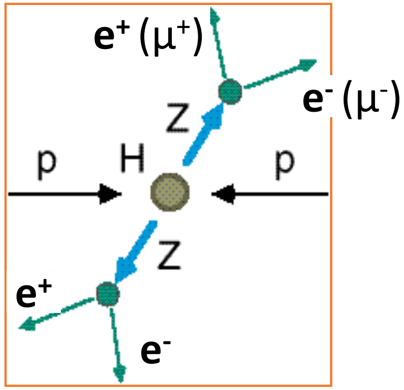
Open Science Grid

RESEARCH DATA ALLIANCE U.S.

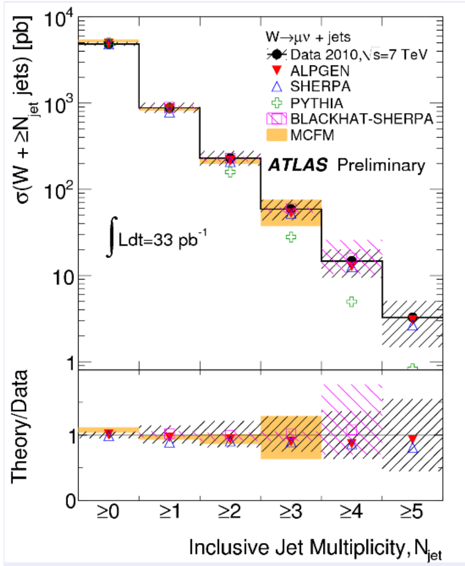
- Many of these rare particle are so heavy they decay into other lighter particles instantaneously
- When one searches for a new particle, one looks for the easiest way to get at them
- Of many signatures of the rare particle final states, some are much easier to find ➔e.g. for the Standard Model Higgs particle
    - H→ gg
    - H → ZZ* → 4e, 4μ, 2e2μ, 2e2v and 2μ2v
    - H→ WW*→2e2v and 2μ2v
    - And many more complicated signatures

How do we look for a rare particle?

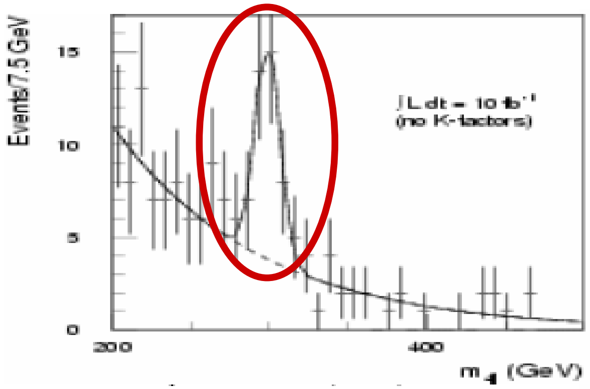
1. Identify Higgs candidate events



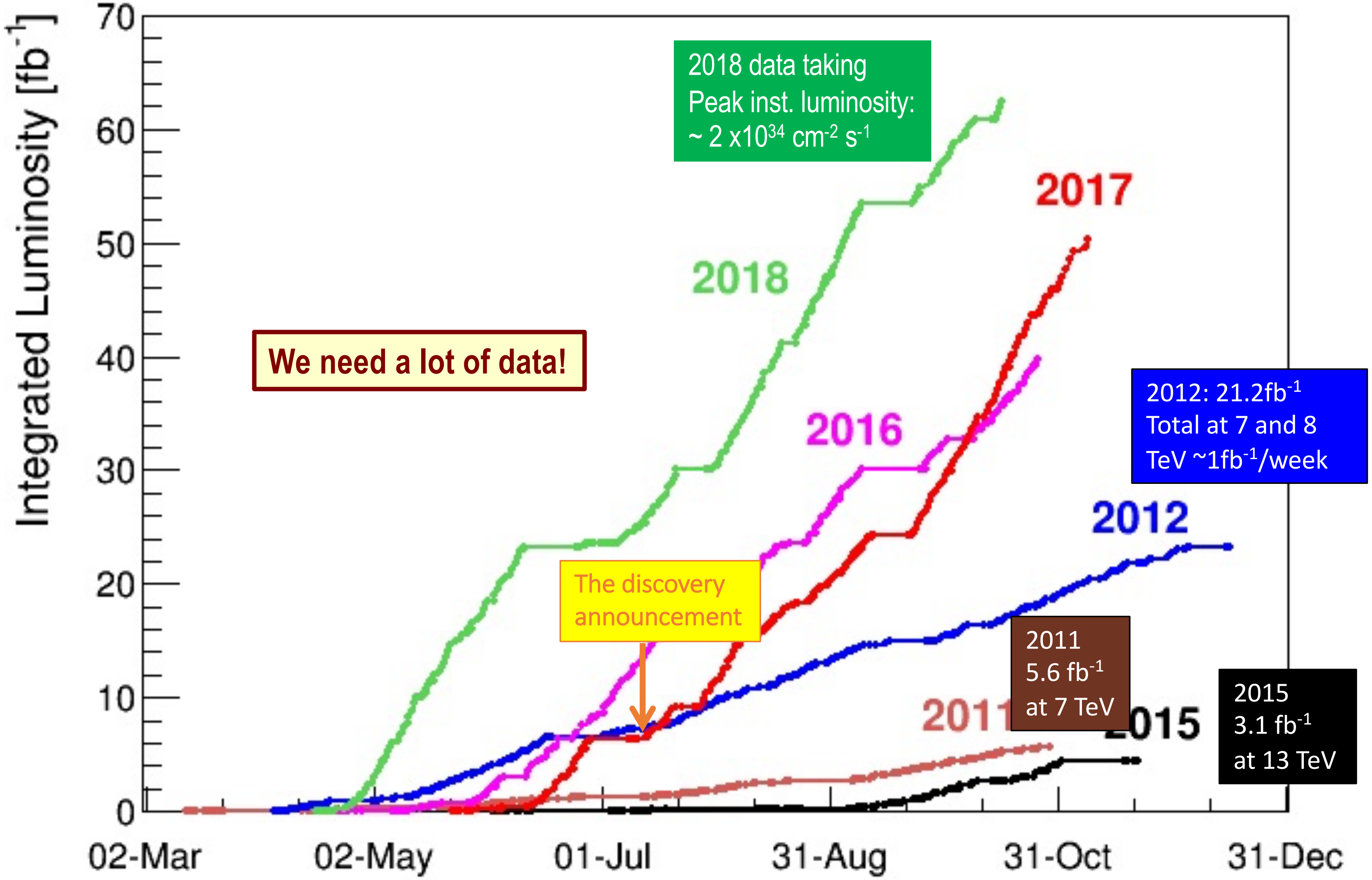
2. Understand backgrounds



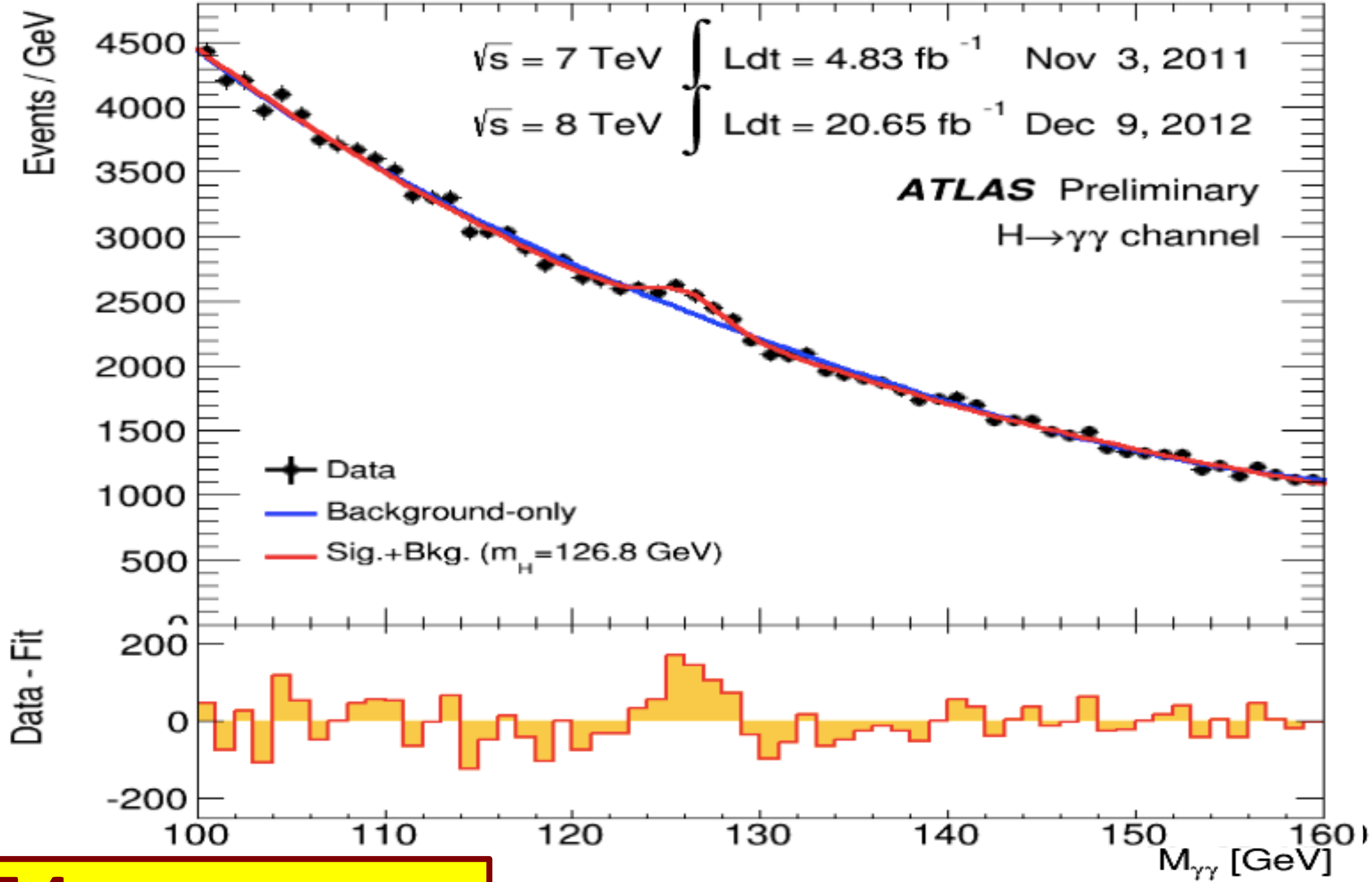
3. Look for a bump!



Amount of the LHC Data



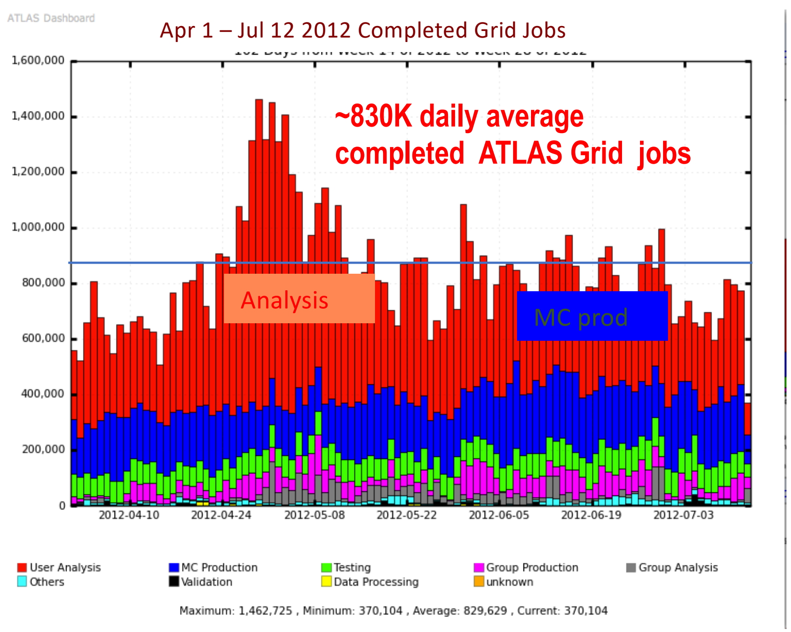
What did statistics do for Higgs→ $\gamma\gamma$?

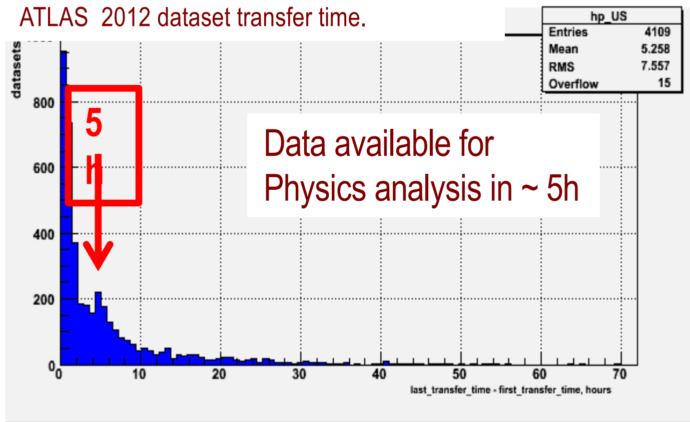


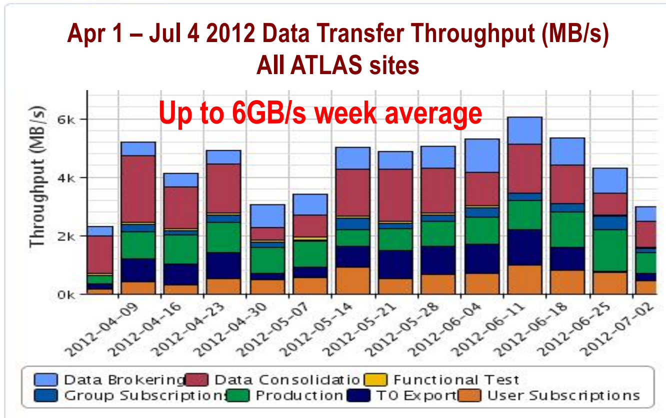
Excess of 7.4 σ at M_H=126.5GeV

Performance of the Grid for LHC

- ATLAS Distributed Computing on the Grid : 10 Tier-1s + CERN + ~70 Tier-2s +…(more than 80 Production sites)
- High volume, high throughput process through fast network!!





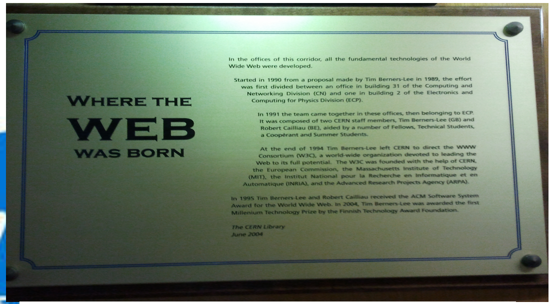


The commercial world picked up..

Open Science Grid

RESEARCH DATA ALLIANCE U.S.

Early 90's



2004

1996

1998

2006

Many private entities fully utilized the internet communication we've developed to multi-trillion dollar venture!!

The concept of cloud and the HTC turned into a new area of study, the Data Science!!

GenAI is Changing the Game Again

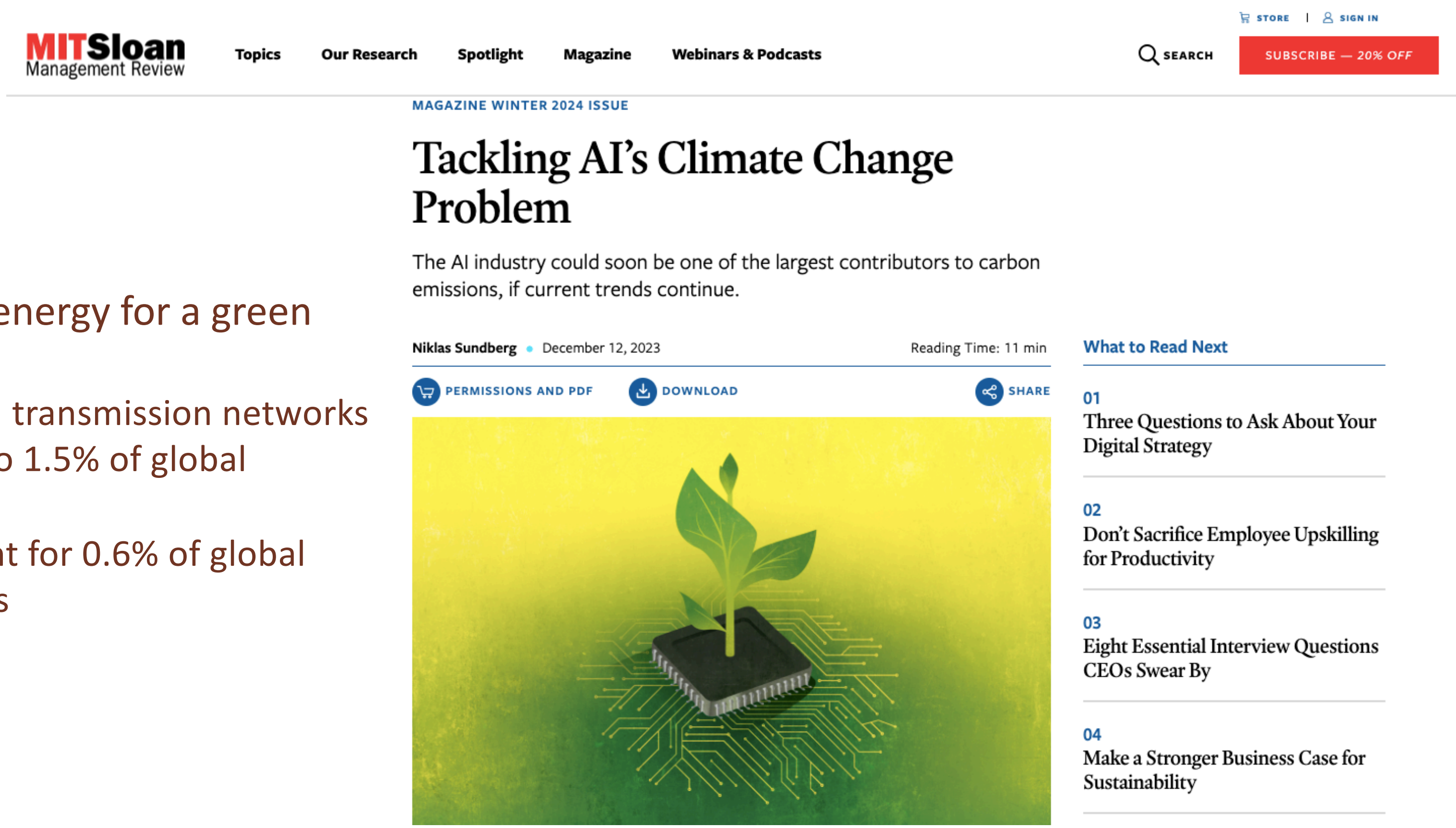
Compute power is driving innovation

- from GenAI to Quantum Computing

Climate Change and the Cloud

Need sustainable energy for a green computing future

- Data centers and transmission networks account for 1% to 1.5% of global electricity use
- They also account for 0.6% of global carbon emissions


MITSloan Management Review
Topics | Our Research | Spotlight | Magazine | Webinars & Podcasts
STORE | SIGN IN
SEARCH
SUBSCRIBE — 20% OFF

MAGAZINE WINTER 2024 ISSUE

Tackling AI's Climate Change Problem

The AI industry could soon be one of the largest contributors to carbon emissions, if current trends continue.

Niklas Sundberg • December 12, 2023 Reading Time: 11 min

PERMISSIONS AND PDF | DOWNLOAD | SHARE

What to Read Next

01 Three Questions to Ask About Your Digital Strategy

02 Don't Sacrifice Employee Upskilling for Productivity

03 Eight Essential Interview Questions CEOs Swear By

04 Make a Stronger Business Case for Sustainability

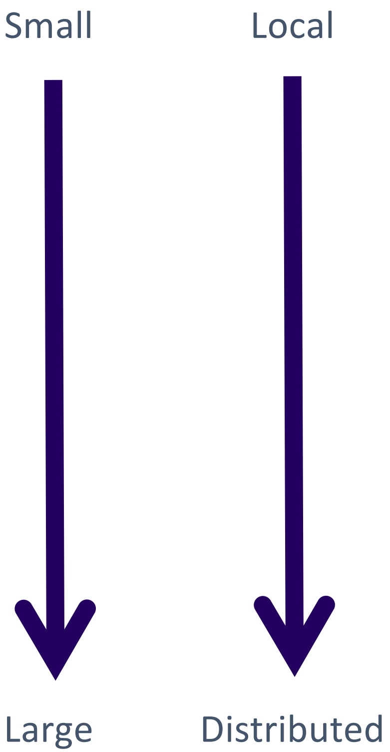
Open Science Grid

RESEARCH DATA ALLIANCE U.S.

Why is HTC hard?

- The HTC system has to keep track of:
  - Individual tasks (a.k.a. jobs) & their inputs
  - Computers that are available
- The system has to recover from failures
  - There will be failures! Distributed computers means more chances for failures.
- You have to share computers
    - Sharing can be within an organization, or between orgs
    - So you have to worry about security
    - And you have to worry about policies on how you share
- If you use a lot of computers, you have to handle variety:
  - Different kinds of computers (arch, OS, speed, etc..)
  - Different kinds of storage (access methodology, size, speed, etc…)
  - Different networks interacting (network problems are hard to debug!)

Let's take one step at a time



- Can you run one job on one computer?
- Can you run one job on another computer?
- Can you run 10 jobs on a set of computers?
- Can you run a multiple job workflow?
- How do we put this all together?

This is the path we'll take

What does the user provide?

- A “headless job”
  - Not interactive/no GUI: how could you interact with 1000 simultaneous jobs?
- A set of input files
- A set of output files
- A set of parameters (command-line arguments)
- Requirements:
  - Ex: My job requires at least 2GB of RAM
  - Ex: My job requires Linux
- Control/Policy:
  - Ex: Send me email when the job is done
  - Ex: Job 2 is more important than Job 1
  - Ex: Kill my job if it runs for more than 6 hours

What does the system provide?

- Methods to:
  - Submit/Cancel job
  - Check on state of job
  - Check on state of available computers
- Processes to:
  - Reliably track set of submitted jobs
  - Reliably track set of available computers
  - Decide which job runs on which computer
  - Manage a single computer
  - Start up a single job

Quick UNIX Refresher Before We Start

- **$** #This symbolizes the prompt.
- ssh username@training.osgconnect.net
- nano, vi, emacs, cat >, etc.
- **which**, **rpm**, **ps**, mkdir, cd, **gcc**, ls
- A varitey of condor_* commands

Questions?

- Questions? Comments?
  - Feel free to ask us questions now or later:
  - Julia Gray julia.ann.gray@gmail.com
  - Horst Severini severini@ou.edu
  - Pat Skubic pskubic@ou.edu
  - Jaehoon Yu jaehoonyu1@gmail.com

Exercises start here:
https://osg-htc.org/dosar/ASP2024/ASP2024_Materials/
Presentations are also available from this URL