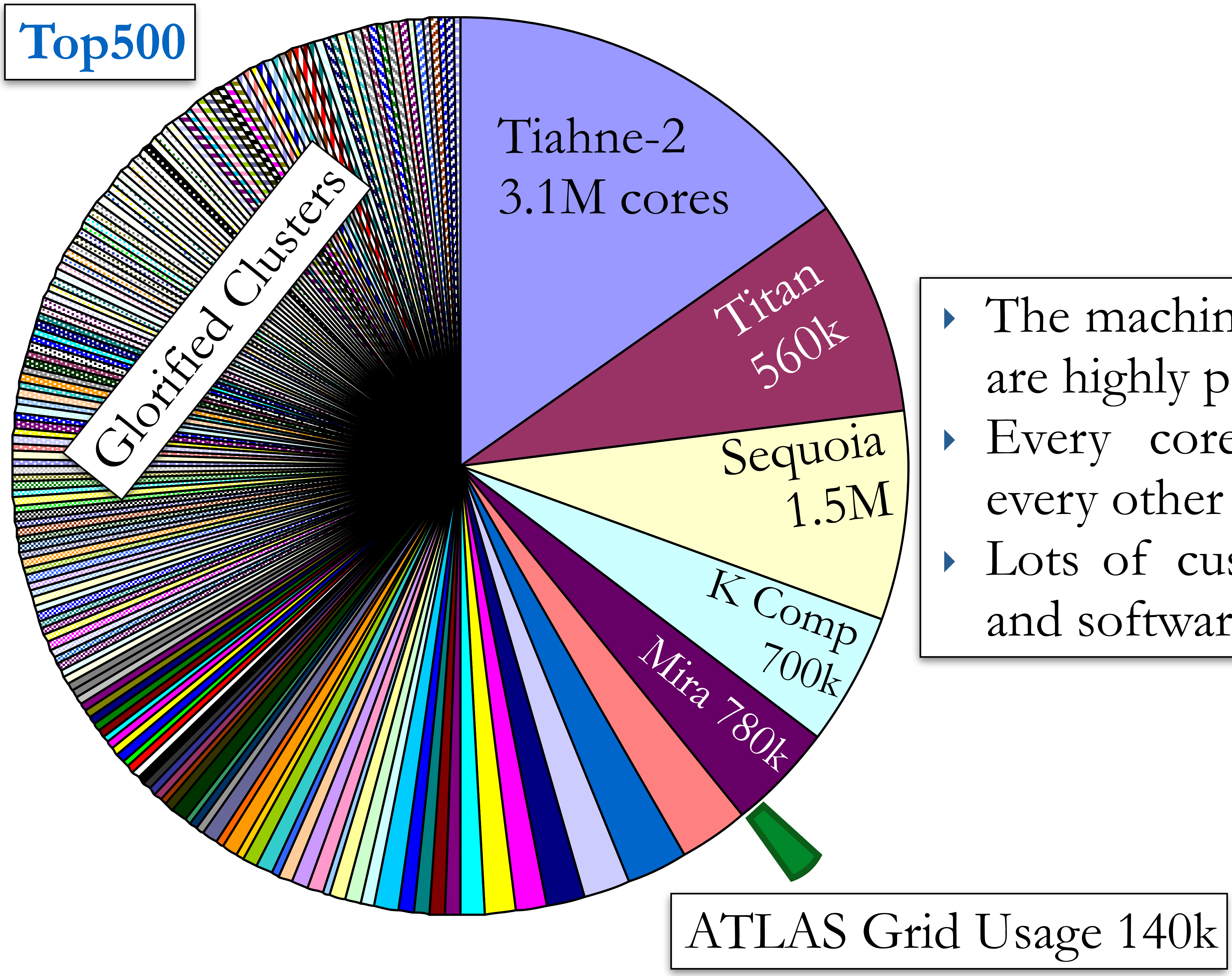
What can HPCs do for G-2?

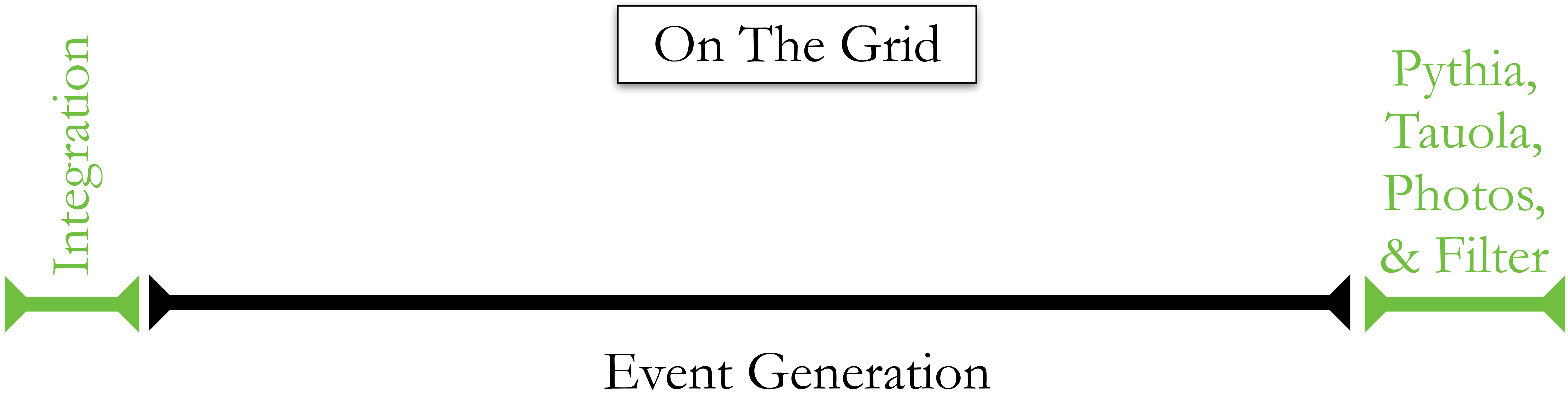
Taylor Childers (Argonne)

High Performance Computers



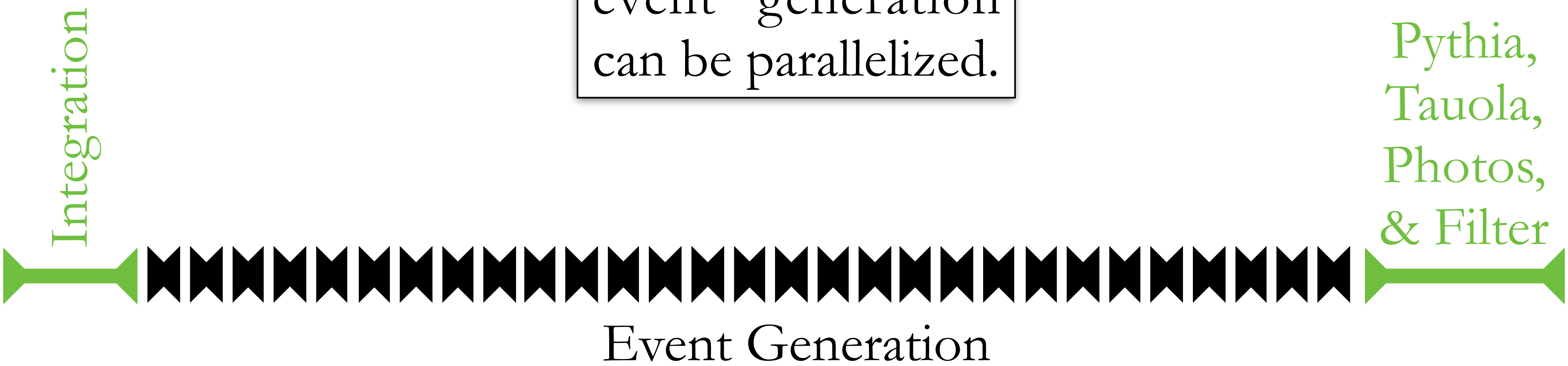
- The machines on the right are highly parallel.
- Every core can talk to every other core quickly.
- Lots of custom hardware and software

Example ATLAS Event Generation

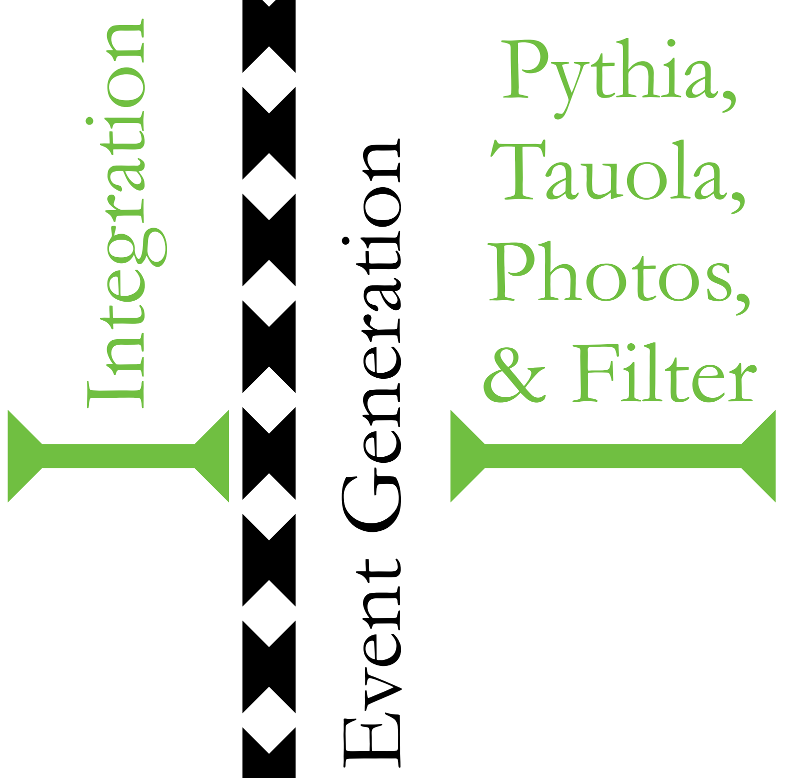


Example ATLAS Event Generation

On an HPC the event generation can be parallelized.

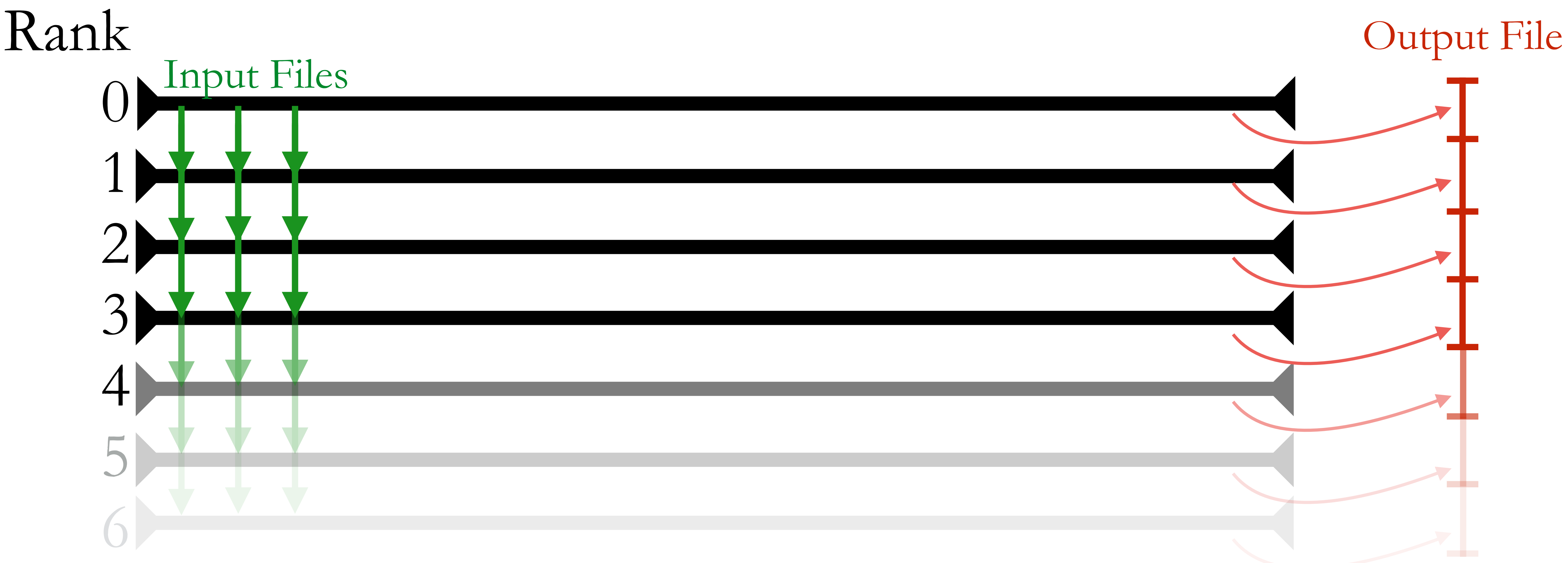


Example ATLAS Event Generation



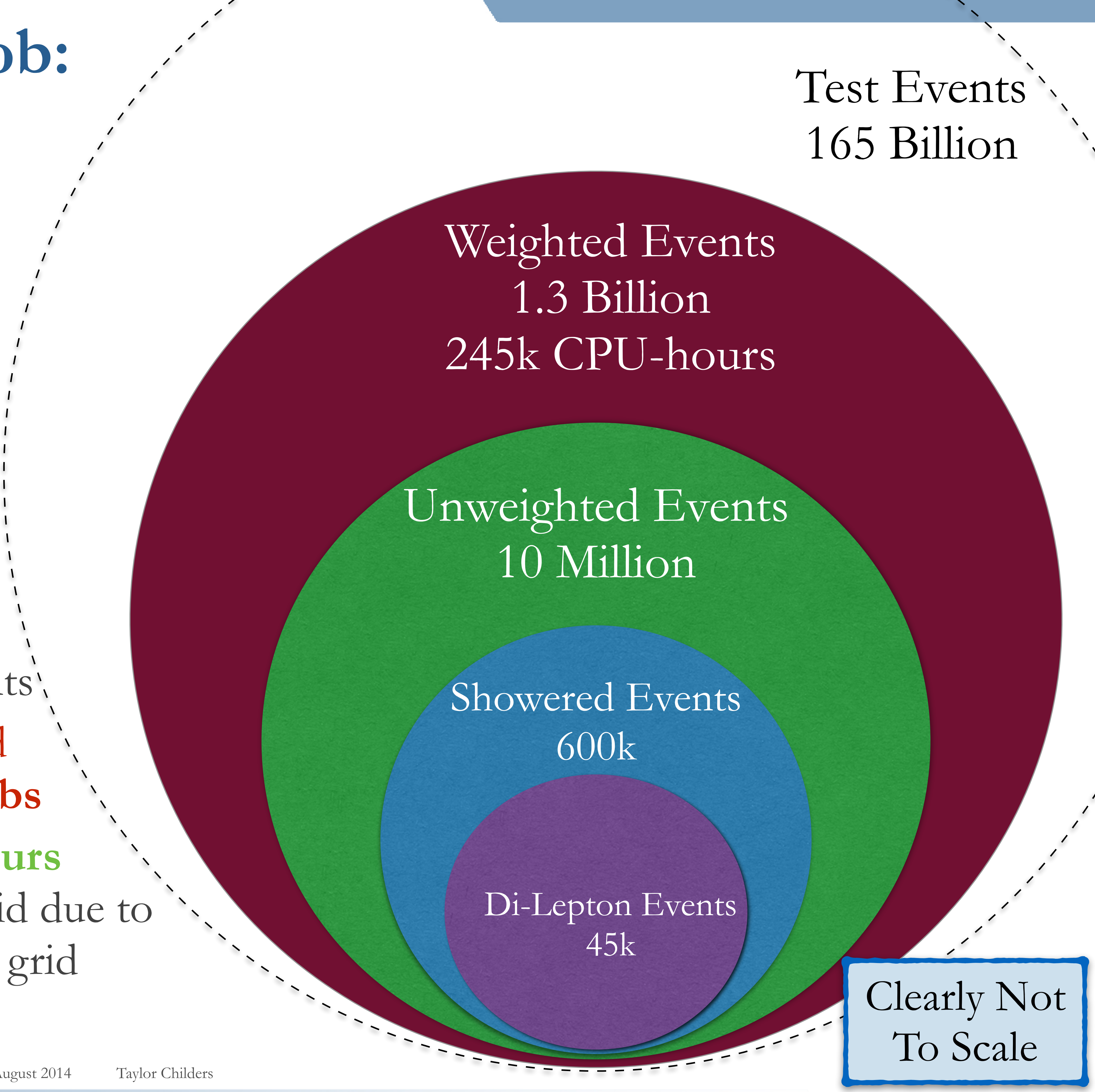
- Minimum Mira job uses 512 nodes.
- Currently running 32 threads per node (16,384 parallel instances of Alpgen using MPI)
- At 64 threads/node goes to 32,768.
- We've run at 8192 nodes (265k threads)

What does this look like in detail?



- First thing Alpgen does is read input files
- All 250,000 ranks would read the same 4 files
- Instead, have rank 0 read files, then broadcast that information to all other ranks. Minimal code changes.
- At the end, all ranks write to a single memory block of an output file.

Completed Job: Z+5jets



- **45,000 Z→ττ+5jet** Alpgen+Pythia events
- Would have required **12,250 24hr Grid jobs**
- **Saved 50k CPU-hours** compared to the Grid due to work duplication in grid workflow

ATLAS Usage

- We've been running ATLAS MC event generation on Mira:
  - Alpgen
  - Sherpa
  - Pythia
  - Geant 4.10
- Have spent some time modifying Alpgen to run at large scales
  - Teaches us what problems we face
  - Uses the machine as it is meant to be used
- Alpgen scales to 10,000 parallel processes with minimal changes
  - Use MPI to set random numbers based on rank number
- Alpgen scales to 250,000 parallel process with modest changes
  - Each rank reads the exact same file 3 files which wreaks havoc on the meta-data servers
  - Each rank writes 5 files which wreaks havoc on the IO nodes
  - Used MPI so that only rank 0 reads the files, but broadcasts them to the other ranks
- Have run Z/W+5 jets with 256,000 parallel processes (1/6 of Mira)
- Work recognized with an ALCC allocation of 50M CPU-hours on Mira, 2M CPU-hours at NERSC

Imagine there's a Simulation Fairy!

- This fairy can provide you with as much simulation as you want.
- What would you simulate?
- What aspects of the G-2 setup could you study and possibly improve?

Mira: The Simulation Fairy

- 786k 1.6GHz PowerPC A2 cores
- 16 cores/node
- 4 physical threads/core
- 16GB/node (250MB/thread)
- 512 nodes/job minimum
- 10 Petaflop max
- ~6.8 billion CPU-hours

Two ways to get CPU-hours

Allocation

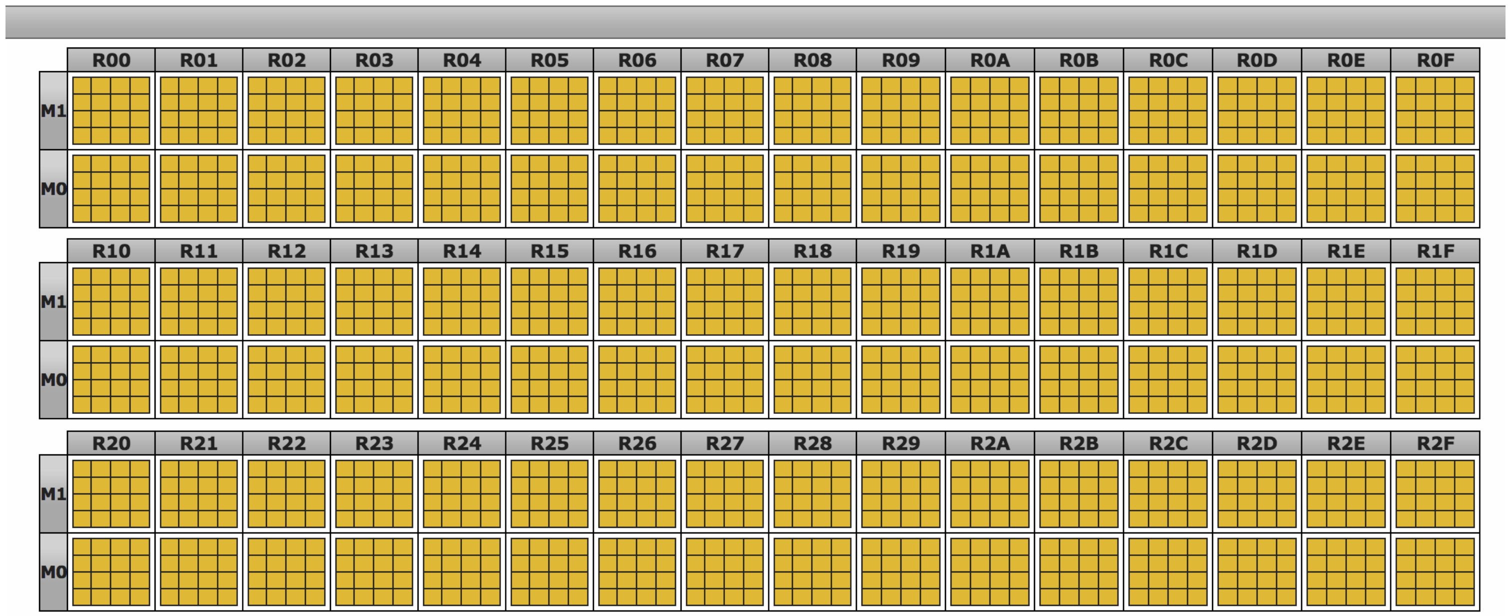
- Competitive Application Process
- Range in size from 10M-500M CPU-hours
- Enter job queue with quality priority

Back-fill

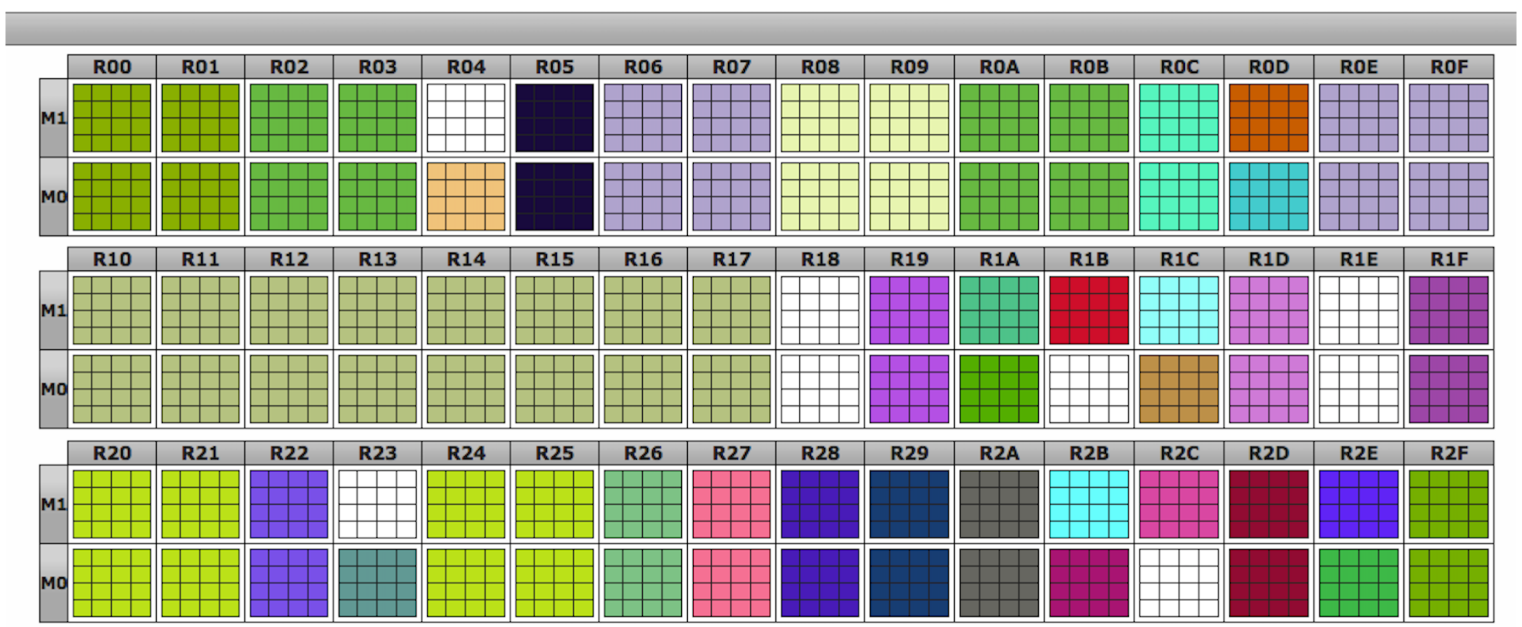
- Competitive mainly with Lattice QCD folks
- On average 5-10% of the machine is under utilized because of variation in job sizes.
- Enter job queue with no priority
- Jobs run in the case that no job with an allocation can fit into current hole

Back-fill Queuing

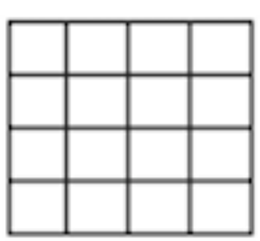
Mira Activity



What you think is Full



What an LCF considers Full

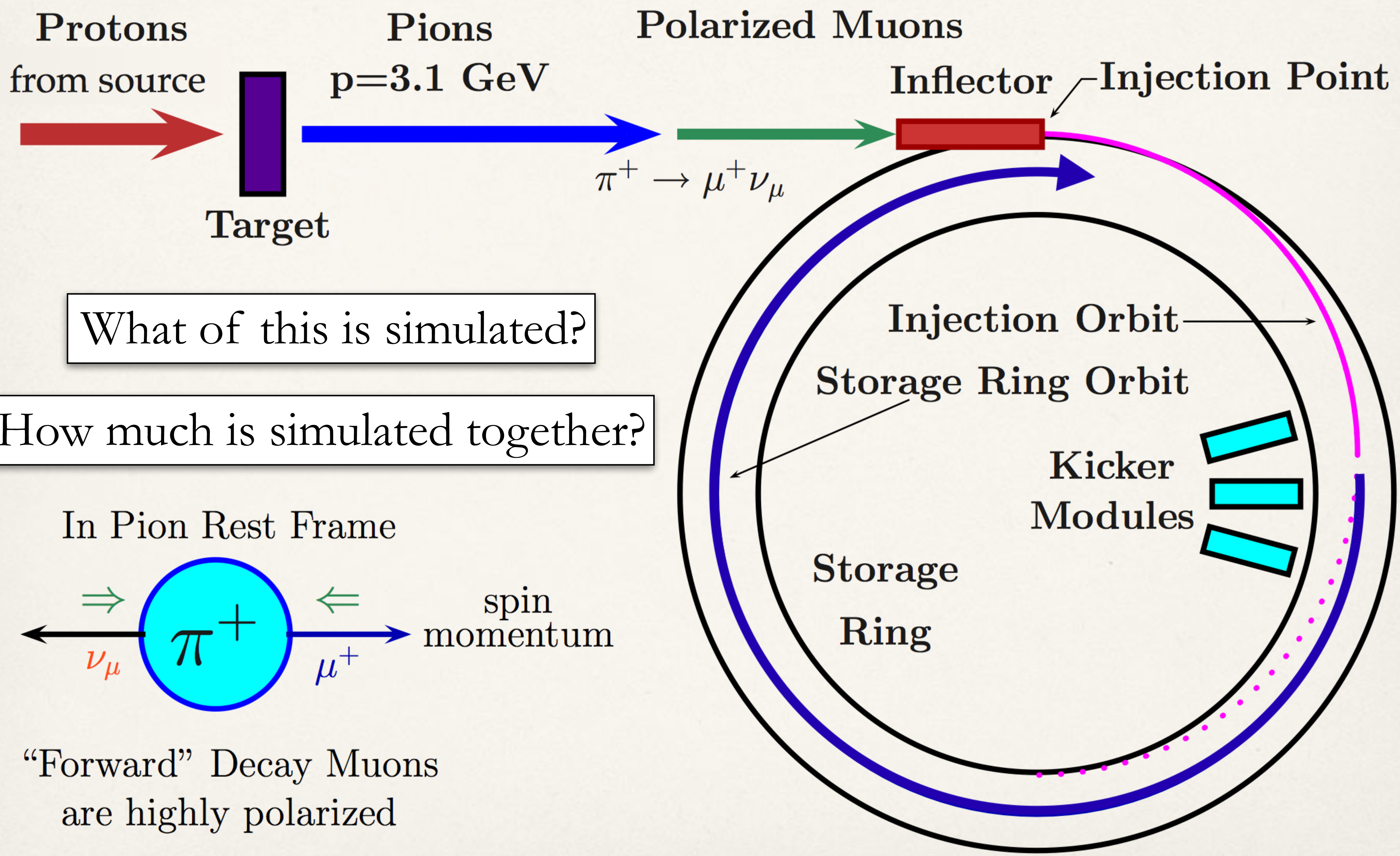


Empty 512 Nodes (minimum Mira job size)

Questions for you

- Where do you run simulations now? Will you use the Grid?
  - Would like to know how much simulation you can afford without HPCs.
- What kinds of simulations do you run?
  - Beam-line+Rings+Detectors = Geant4+ART
  - Target (p->pion) = MARS
- Are there any constraints or things that make you say, "man, I wish we could do…"?
- If you were given unlimited CPU time, what would you do with it?
- What are typical run times for your simulation on what kind of machine?
  - Want to be able to estimate for the proposal how many runs we can simulate for X amount of CPU-hours.
- If we can produce multiple G-2 datasets, are the calibrations, measurements, systematics we can optimize?
  - B-Field measurement (average vs. samples)?
  - Coherent Betatron Oscillations? Can we simulate these?
  - Beam Dynamics? Muon Momentum Distributions? Pileup? Lost Muons?

The Brookhaven Experiment



What of this is simulated?

How much is simulated together?