Antimatter

Using High Throughput Computing to Study Very Rare Processes

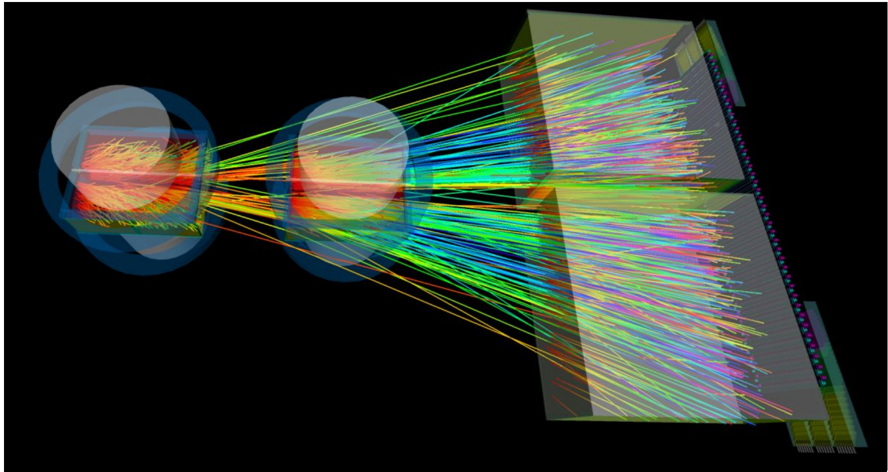
Anirvan Shukla

Department of Physics, University of Hawaii at Manoa

OSG Virtual School 2021, August 11

What is antimatter?

- Matter is made of elementary particles like protons, electrons etc.
- Each elementary particle has a corresponding antiparticle.
- I study antimatter production in particle accelerators at the European Organization for Nuclear Research (CERN), which is the largest particle physics experiment in the world.
- I also model how antimatter is produced when cosmic rays interact with the interstellar medium.



Colliding two protons at the NA61/SHINE experiment at CERN.

What can antimatter tell us about dark matter?

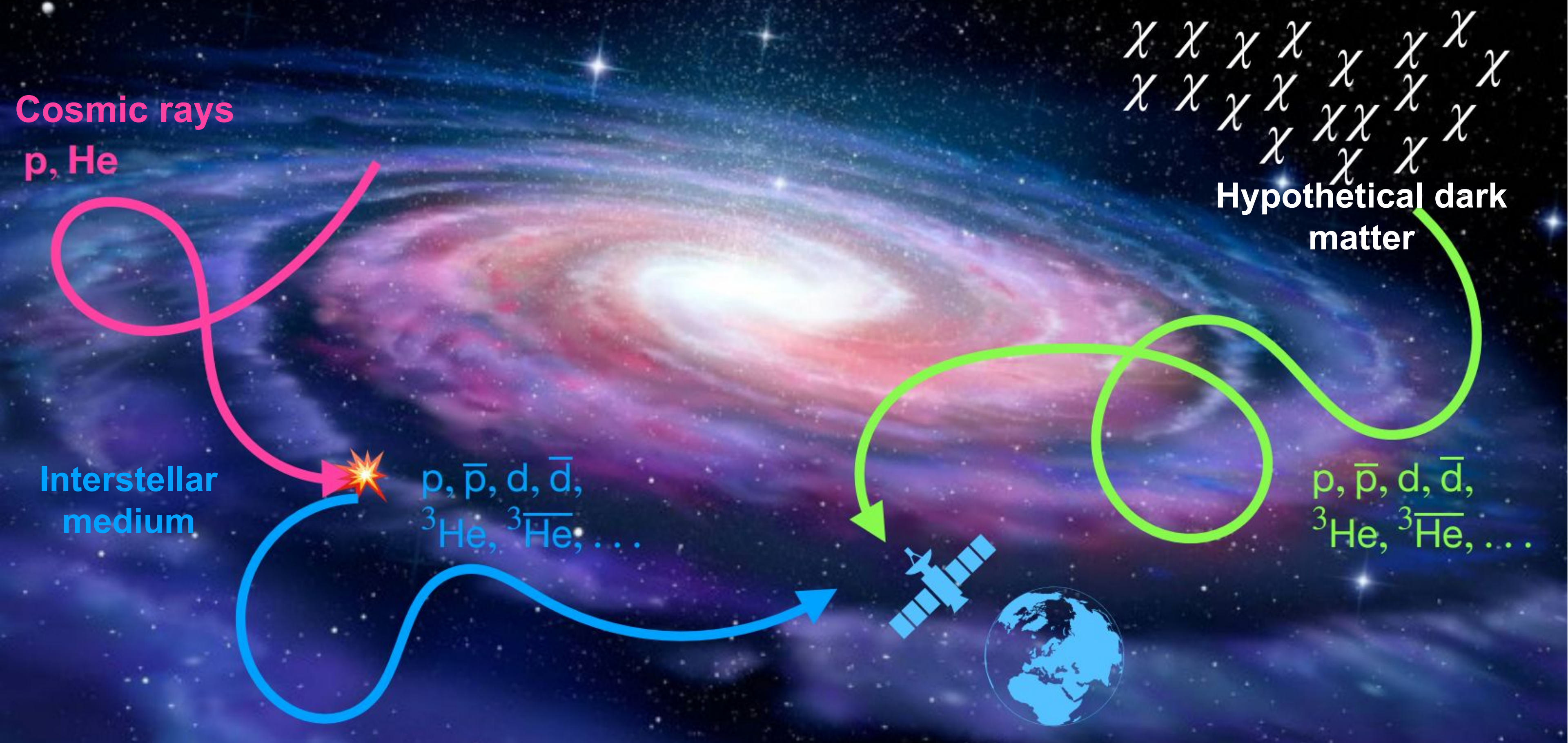


Image from S. Hornung, ICHEP 2020

Why can you not simulate this on your laptop?

Production of heavier antimatter is extremely rare!

- 1 antihelium particle is produced in every 10 billion - 1 trillion events.
- 1 million events need ~1 CPU-hour.
- Total ~100 trillion events needed!
- About 100 million CPU- hours need (~12,000 years on a single CPU).

Data storage challenges

- 10 million events stored in 10 GB output file.
- With 100 trillion events, need ~100 million GB of storage.

User School 2016 to kickstart my project

- Access to my existing computing resources at CERN and the University of Hawaii's HPC cluster were not enough for this project.
- The User School made me familiar with the OSG's capabilities, and the software environment.
- Information about preinstalled software using "modules", path to different gcc/g++ compilers, etc. was very helpful.
- Hands-on experience of running simple example jobs during the User School helped in learning the basics of HTCondor.

A simple workflow

Start with a simple HTCondor job, and iteratively build on it.

My final jobs consisted of the following:

- HTCondor submit file
  - Handle input files to transfer to compute node.
  - Launches bash wrapper script.
- Bash wrapper script
  - Load software modules.
  - Move configuration files, custom software/libraries, temporary directories.
  - Launch python script.
- Python wrapper script
  - Launches multiple C++ program in sequence
  - Transfers final histogram file to server in Hawaii
  - Performs clean up, and exits job.
- Two statically-compiled C++ executables.

These jobs were designed to run for 10 hours on the OSG.

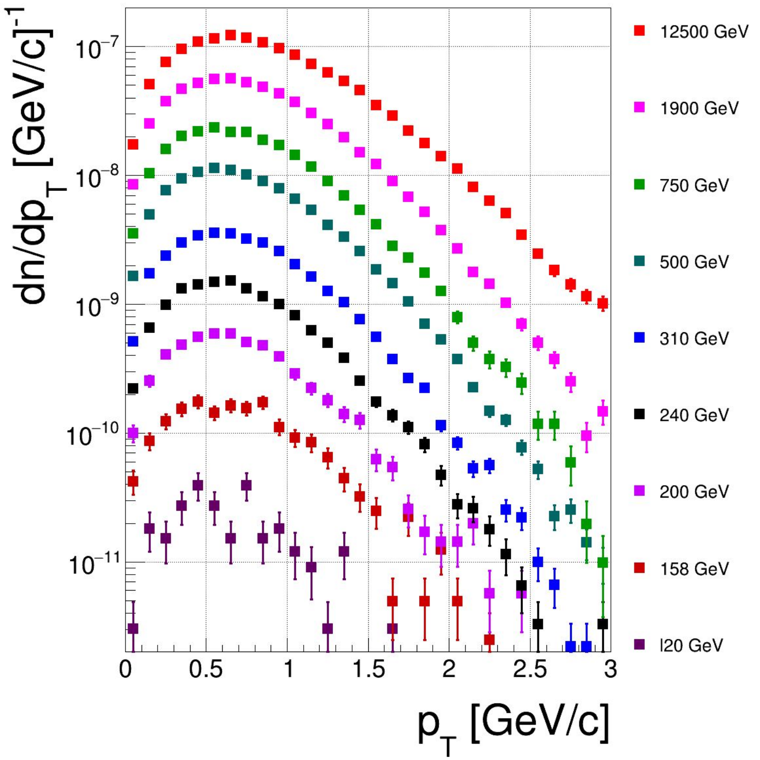
Finding solutions to the storage constraints

How to handle the large (100 million GB of output files)?

- Breaking simulations into chunks of 10-hour jobs with 10 million simulated events.
  - Output file of ~10 GB.
- Analyzing the final output file within the job.
  - Extracting all possible data of interest into hundreds of histograms.
  - Lose detailed information at the individual particle level - a compromise.
  - Drastically reducing file size: 10 GB output file -> 100 MB histogram file.
- No local storage!
  - Transfer the 100 MB histogram file directly from the compute node to our server in Hawaii.
- But 10 million files of 100 MB each i.e. total ~1000 TB!
  - However, histograms are excellent for “adding” up.
  - Final size of simulation output: ~2 GB!
  - This approach was computationally expensive, but the final storage required was small.
  - Ideal for using multiple clusters.

New antimatter predictions using the OSG

- Almost 60 trillion proton-proton collisions were simulated to calculate this spectra.
- Using more than 6000 CPU-years on the OSG.
- These spectra could be calculated for the first time using a particle physics model.
- Published in [Phys. Rev. D 102, 063004 (2020)].
- Already well received by the experimental community.



A new project, and solving new challenges with DAGMan

- To expand on my previous work, my earlier workflows needed more optimizations.
  - Shorter OSG jobs (~1 hour) run with a higher success rate, as opposed to 10-hour jobs.
  - Number of concurrent jobs is also much higher with short jobs.
  - But 10x file transfers was a bottleneck.
  - My jobs were failing at a high rate of ~50%.

- Restructuring the old workflow using a very basic DAG.
  - OSG jobs are submitted by the DAGMan job.
  - On job completion, the output file is transferred to the OSF submit node via `transfer_output_files` and `transfer_output_remaps`.
  - A DAGMan POST script runs after each job, and takes care of file transfers from the submit node to Hawaii.

- DAGMan also does the babysitting - I can list 100,000 jobs, and DAGMan throttles job submission in a way that adapts to the load on the OSG pool.

OSG usage statistics

- Typical resourced requested
  - CPU: 1
  - Memory: 1-2 GB
  - Disk space (10-15 GB).

- The compute nodes were widely distributed across the US. The table on the right shows the top 20 facilities for my jobs.

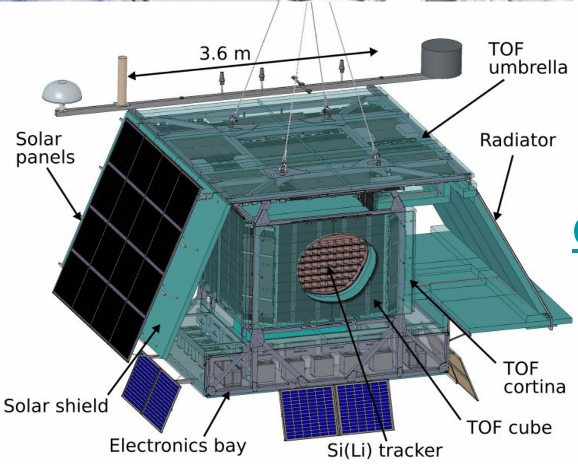
- Core hours used in last two years:
  - Total: 49 million core-hours (5600 core-years)
  - Total jobs: 8.3 million

Total core hours by facility

SU ITS	26 Mil
IRISHEP-SSL-UCHICAGO	6 Mil
IIT - Illinois Institute of Technology	4 Mil
Purdue Geddes	4 Mil
UColorado_HEP	1 Mil
UConn-HPC	886 K
OU ATLAS	668 K
Utah-SLATE-Notchpeak	563 K
BNL ATLAS Tier1	544 K
UConn-OSG	536 K
AMNH	475 K
Utah-SLATE-Lonepeak	446 K
Georgia Tech	443 K
GLOW	421 K
SLATE-K8S-UCHICAGO	410 K
MWT2 ATLAS UC	392 K
NWICG_NDCMS	365 K
Nebraska-CMS	338 K
ASU Research Computing	315 K
Texas Advanced Computing Center	280 K

Next Steps

- Computational resources are never enough!
- With new ground and spaced-based physics experiments coming online, there is always demand for a larger cluster, more storage, etc. to analyze the generated data.
- Another member of my research group has recently started using the OSG for the GAPS experiment.
- Hopefully, the OSG will continue to be a critical resource in our future projects.



GAPS