LIGO

Possible Gravitational-Wave Contributions to HEP-SCORE

Josh Willis

LIGO Laboratory, Caltech

17 February 2021

LIGO-G12100208-v2

Overview of GW Analyses

- Time series data, with many distinct analyses
  - » Searches: matched-filter transient, un-modeled transient, stochastic background, continuous wave
  - » Detector characterization and data quality
  - » Parameter estimation: detailed follow-up of significant transients
- Benchmarking has focused on most computationally intensive pipelines
- Most of our analyses are HTC. Continually porting more to run on Open Science Grid via Glidein-WMS

Modeled Transient Searches

- Compact Binary Coalesence (CBC)
- Computing figure-of-merit: templates-per-core:
  - » $TPC = \frac{\text{(time analyzed)} \times \text{(templates analyzed)}}{\text{(number of cores)} \times \text{(compute time)}}$
  - » This means that for a given template bank and amount of time series data, the expected number of core-hours is:
  - » core–hours = (detector time)×(number of templates)$/TPC$
- Several such pipelines: gstlal (time domain), MBTA (multi-banded frequency domain), SPIIR (time domain on GPU), PyCBC (frequency domain)
- All look at all acceptable detector time, and must identify significant events, down-ranking noise

LIGO

PyCBC

- Algorithmically FFT-limited, so we have benchmarked based on FFT performance
- Complication: one highly competitive FFT library (FFTW) only obtains optimal performance when allowed days to “plan” for a given architecture and configuration (MKL, CUDA do not have this issue)
- CPU version of workflow is containerized (docker→singularity via OSG) though have not attempted HEP core framework as yet

LIGO

Future: Parameter Estimation

- Has become one of our largest users of computing, due to (exciting!) high event rate
- Underlying algorithm is MCMC or nested sampling
- Dominated by generation of gravitational waveforms, which makes it an interesting target for a benchmark (highly domain-specific, not readily decomposable into common numeric library kernels; at present not well vectorized)
- Currently target of aggressive optimization, probably not yet ready for standardized benchmarking?