GPE: GPU-accelerated Parameter Estimation for gravitational waves with x360 acceleration

YUN-JING HUANG
ACADEMIA SINICA, TAIWAN
2021 MAR 30
yunjinghuang14@gmail.com
Outline

- Introduction
- Motivation
- Parallelization Methods
  1) Waveform
  2) Nested Sampling
- Comparison with LALInference – GW150914
  - Parameter Estimation Result
  - Performance Test
Use Bayes' theorem to estimate the posterior distribution of the source parameters.

Stochastically sample the posterior.

Time-consuming process

- One parameter estimation run requires collecting tens of thousands of samples from the posterior.
- Each sample is found by conducting thousands of random walks.
- Each random walk involves one waveform generation.
  - One parameter estimation run requires more than millions of waveform calculations.

May take several days to several weeks to run parameter estimation on a single event.

Bayes' theorem

$$p(\theta|d, H) = \frac{p(\theta|H)p(d|\theta, H)}{p(d|H)}$$

posterior  prior  likelihood  evidence
Speedup Motivation

- Increase in detection rate
- Statistical studies
  - e.g. detector observing scenarios
  - Parameter estimation of large number of events.
- EM follow-up
  - Fast and accurate production of sky localization regions.
GPE Overview

- Adapted from the nested sampling flavor of LALInference (lalinference_nest)
- Newly written in C++ and CUDA
- Produces same output for cbcBayesPostProc
- Precision choices
  - Double precision
  - Single precision
Parallelization method #1: Waveform

- Dominant source of time consumption
- Two frequency-domain waveforms:
  - TaylorF2
  - IMRPhenomPv2
- Calculations in each frequency bin are independent from each other, therefore they are highly parallelizable.
- Calculations for each detector are also independent.
- Each GPU thread -> each frequency bin of each detector
- ~33,000 parallel calculations per waveform call (two detectors)
- ~1,000 calls per iteration
- ~16,000 iterations per run
- ~$10^{11}$ calculations per run
Parallelization method #2: Nested Sampling

- Calculates the evidence \( (Z) \) and produces the posterior as a by product
- Maps multi-dimensional parameter space into one-dimensional priormass space.
  - By defining “nested” likelihood contours and their enclosed priormass
- Algorithm:
  - Sprinkle several samples uniformly in prior.
  - At each iteration, replace lowest likelihood sample with new sample within a likelihood contour.
  - Finding a new sample => Parallelization!

Parallelization method #2: Nested Sampling

- **LALInference:**
  - Copy one existing sample
  - Perform several random walks
  - Find one new point

- **GPE:**
  - Copy *many* existing samples
  - Perform several random walks for *many* points *at the same time*
  - Find *many* new points
  - Save additional points for later use
Comparison with LALInference: GW150914

- Use GW150914 data from LIGO Open Science Center
- Number of live points: 500
- LAL: use lalinference_nest with IMRPhenomPv2 waveform
- GPE: parallelized IMRPhenomPv2 waveform
  - Double precision
  - Single precision
Parameter estimation result
<table>
<thead>
<tr>
<th>Code</th>
<th>Parallelization Method</th>
<th>Precision</th>
<th>Hardware</th>
<th>Cores</th>
<th>Max Clock rate</th>
<th>Wall time</th>
<th>Speedup w.r.t. LAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAL</td>
<td></td>
<td>Double</td>
<td>Core™ i7-8700 CPU</td>
<td>1</td>
<td>3.20 GHz</td>
<td>20h53m37.7s</td>
<td></td>
</tr>
<tr>
<td>GPE</td>
<td>Waveform</td>
<td>Double</td>
<td>GeForce GTX 1080 Ti</td>
<td>3584</td>
<td>1.58 GHz</td>
<td>26m2.0s</td>
<td>×48.2</td>
</tr>
<tr>
<td></td>
<td>Waveform + Nested</td>
<td>Single</td>
<td></td>
<td></td>
<td></td>
<td>13m45.9s</td>
<td>×91.1</td>
</tr>
<tr>
<td></td>
<td>Sampling</td>
<td>Double</td>
<td></td>
<td></td>
<td></td>
<td>10m56.0s</td>
<td>×114.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Single</td>
<td></td>
<td></td>
<td></td>
<td>3m27.4s</td>
<td>×362.7</td>
</tr>
</tbody>
</table>
Summary

- GPE can achieve a 360 times acceleration
- GPE can produce consistent results with LALInference
- Speedup is essential for
  - Parameter estimation of real events in multi-detection era
  - Large simulations
  - EM follow-up