Counting pairs using C++ and openmp

Martin Kerscher

Fakultät für Physik
Ludwig–Maximilians–Universität München

code coffee - January 26, 2021

Why $DD(r)$?

- Galaxy distribution $\{\mathbf{x}_i\}_{i=1}^N$
- Correlation function - the simplest estimator

$$\hat{\xi}(r) = \frac{DD(r)\, N_R^2}{RR(r)\ N^2} - 1,$$

and also better estimators, use the paircount

$$DD(r) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \mathbb{1}_{[r_k, r_{k+1}]}(|\mathbf{x}_i - \mathbf{x}_j|).$$

- Unfortunately thats $O(N^2)$ and we wont get rid of that scaling ...
  ... but we can throw hardware at it.

Parallelizing with openmp

- `openmp` — a simple framework for parallelization on shared memory machines: `https://www.openmp.org/`
- splitting the calculation

$$\begin{aligned}\sum_{i=1}^{n} i &= \sum_{i=1}^{n/2} i \quad + \sum_{i=n/2+1}^{n} i \\ &= \ldots + \ldots \quad + \quad \ldots + \ldots\end{aligned}$$

- parallelizing on several threads

```
sum = 0;

#pragma omp parallel for reduction(+:sum)
for (int i=1; i<=n; ++i) {
  sum += i;
}
```

Parallelizing $DD(r)$

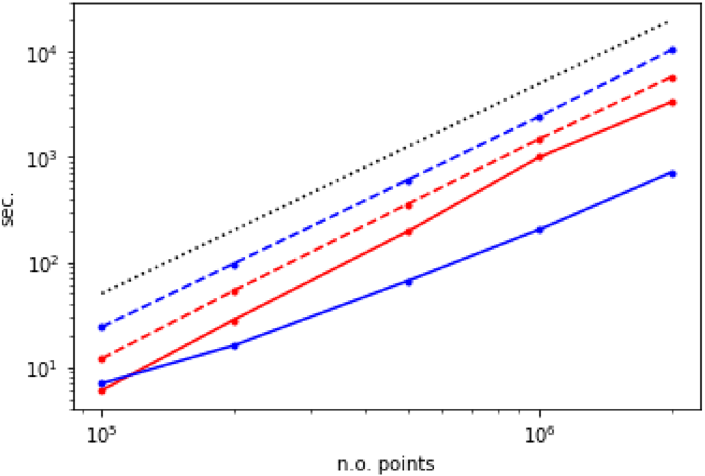
split the sum, run seperate threads, accumulate in the end.

$$DD(r) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \mathbb{1}_{[r_k, r_{k+1}]}(|\mathbf{x}_i - \mathbf{x}_j|)$$

$$= \sum_{i=1}^{N/2} \sum_{j=i+1}^{N} \mathbb{1}_{[r_k, r_{k+1}]}(|\mathbf{x}_i - \mathbf{x}_j|) + \sum_{i=N/2+1}^{N} \sum_{j=i+1}^{N} \mathbb{1}_{[r_k, r_{k+1}]}(|\mathbf{x}_i - \mathbf{x}_j|)$$

$$= \ldots + \ldots \quad + \quad \ldots + \ldots$$

Timing results



Gain:

1.7 x current small PC (2 cores 4 threads 2.50GHz 3MB cache)

14 x old compute server (8 cores 40 threads 2.30GHz 20MB cache)