1 Introduction

The motivation for developing an optimized Java based jet clustering algorithm, is to provide a platform independent implementation for use in online validation scripts on HepSim[1] repository with Monte Carlo predictions for HEP experiments.

Currently, the most widely used implementation of jet clustering algorithms in high energy physics is FastJet[2]. Is it written in C++ and provides a highly optimized implementation of (anti)kt and Cambridge-Aachen jet finding algorithms with selectable $O(N^2)$ and $O(N \log N)$ complexities, which significantly reduce runtime in comparison to naive $O(N^3)$ algorithms[3].

Unfortunately, platform dependence of C++ is prohibitive for use of FastJet in client-side web application scripts. Therefore, a Java implementation was necessary.

2 Implementation

It turns out, that for reconstruction of jets obtained in pp collisions, with at most a few thousand particles per event, the most efficient algorithms are those with $O(N^2)$ complexity. The $O(N \log N)$ algorithms provide an advantage only in events with more then 10,000 particles, which only arise in experiments with heavy ion collisions[3, 4]. While implementation of $O(N \log N)$ algorithms requires use of sophisticated techniques, like Voronoi diagrams[5], $O(N^2)$ complexity can be achieved with relatively simple approaches described below.

2.1 Generalizaed kt algorithm

2.2 Geometric Factorization

This is the most significant optimization, reducing complexity from $O(N^2)$ to $O(N^3)$. It is explained by the FastJet Lemma[3].

FastJet Lemma 1: If particles i, j form the smallest d_{ij}, and p_t

Motivation:

Proof:

2.3 Tiling

FastJet Lemma 2:

Motivation:

Proof:

2.4 Linked List

3 Benchmark

Make plots to compare to FastJet performarmance.

References

[1] S. V. Chekanov, "HepSim: a repository with predictions for high-energy physics experiments," arXiv:1403.1886 [hep-ph].

[2] M. Cacciari, G. P. Salam and G. Soyez, "FastJet user manual," Eur. Phys. J. C **72** (2012) 1896 [arXiv:1111.6097 [hep-ph]].

[3] M. Cacciari and G. P. Salam, "Dispelling the N^3 myth for the k_t jet-finder," Phys. Lett. B **641** (2006) 57 [hep-ph/0512210].

[4] G. P. Salam and M. Cacciari, "Jet clustering in particle physics, via a dynamic nearest neighbour graph implemented with CGAL," LPTHE-06-02, unpublished note.

[5] F. Aurenhammer, "Voronoi diagrams: a survey of a fundamental data structure," ACM Computing Surveys 23 (1991) 345.