

Using Thrill to Process Scientific Data on HPC

Mariia Karabin
Clemson University
mariiak@clemson.edu

Supreeth Suresh
University of Wyoming
ssuresh@uwyo.edu

Xinyu Chen
University of New Mexico
xychen@cs.unm.edu

Li-Ta Lo
Los Alamos National Laboratory
ollie@lanl.gov

Pascal Grosset
Los Alamos National Laboratory
pascalgrosset@lanl.gov

ABSTRACT

With the ongoing improvement of computation power and memory capacity, the volume and velocity of scientific data keeps growing. This makes the Big Data processing and analytics tools more interesting for scientists to gain insights from huge amount of scientific data. We explored the Thrill library to help scientists at the Los Alamos National Laboratory (LANL) to post-process and analyze simulation data from plasma physics and molecular dynamics. The Thrill library provides high-level application programming interface (API) and leverages the C++ programming language and MPI communication functions to achieve high performance. The analytic results are verified by the scientists. By using the Thrill library, we implemented the required operations with less programming efforts than hand-crafting data processing programs with ad hoc tools.

CCS CONCEPTS

• **Mathematics of computing** → **Cluster analysis; Exploratory data analysis;** • **Computing methodologies** → *Massively parallel algorithms;*

KEYWORDS

Distributed Data Processing, Clustering, Big Data, Molecular Dynamics

ACM Reference format:

Mariia Karabin, Supreeth Suresh, Xinyu Chen, Li-Ta Lo, and Pascal Grosset. 2018. Using Thrill to Process Scientific Data on HPC. In *Proceedings of 47th International Conference on Parallel Processing, Eugene, OR, USA, August 13–16, 2018 (ICPP 2018)*, 2 pages. <https://doi.org/10.1145/3225058.3225149>

1 INTRODUCTION

The challenge of *Big Data* is spreading from the traditional Business Intelligence to scientific researches. To deal with huge volumes, scientists need to put more efforts in processing and analyzing data than before. On the other hand, larger and larger data sets provide

opportunities for new discoveries and insights. At the Los Alamos National Laboratory (LANL), scientists have been using many tools and languages to process and analyze their simulation data. Often times, they use bash scripts to deal with several thousands of, even up to millions of, files. It is helpful to scale up the process for them to handle many files in parallel.

Thrill[1] is a recent parallel data processing library that aims to introduce the high level programming interfaces and declarative programming style into the high performance computing community to facilitate data processing and analysis.

In this poster, we presented our exploration of using Thrill library to process and analyze simulation data to get time series from the Accelerated Molecular Dynamics (AMD) simulation[3], K-means clustering of AMD states and particle trajectories from plasma physics[2]. The results are verified by the scientists. The Thrill implementations took the advantage of distributed computing to process many files in parallel. Experiments show our implementations also have strong scalability to some extent.

2 METHODS

The Thrill library distributes data into distributed immutable arrays (DIAs). It provides scalable algorithmic primitives such as *Map*, *ReduceByKey*, *Sort* and *Join* to operate DIAs. Essentially, DIAs are equivalent to tables in relational databases and algorithmic primitives are equivalent to SQL operations. We used the operation of querying time series and K-means clustering to illustrate the methods to build more complex data processing operations from the above mentioned DIAs and algorithmic primitives.

2.1 Query Time Series

Scientists got up to millions of state-files from the Accelerated Molecular Dynamics (AMD) simulations[3]. Each state file contains the coordinates of hundreds of atoms. AMD simulations also produced trajectory files to describe the sequences and durations of each state. The time series of states and their potential energies are important to find out the points where transitions are happening.

We first *Mapped* a function that uses the LAMMPS library to calculate the total potential energy (PE) to each state files to get a DIA of {state_id, potential_energy}. Similarly we used *Map* and *PrefixSum* to create a DIA of {state_id, time_stamp}. Then we used *Join* to get the final time series of potential energies (PEs). Figure 1 illustrated the entire flow of operations to query time series.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICPP 2018, August 13–16, 2018, Eugene, OR, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-6510-9/18/08...\$15.00

<https://doi.org/10.1145/3225058.3225149>

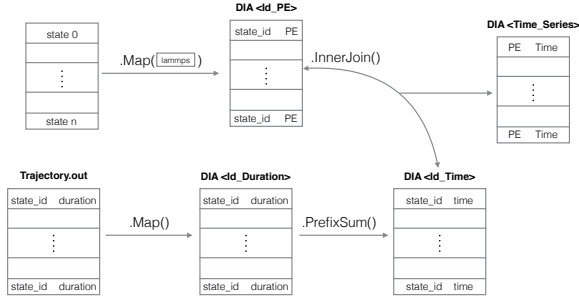


Figure 1: Querying time series of total potential energy.

2.2 K-means clustering of AMD states

To find out clusters of similar AMD states, we *Mapped* a function that uses LAMMPS to calculate 6 common neighbor analysis(CNA) features to each state. This operation created a DIA of states characterized by a 6-dimensional CNA-feature vector{state_id, cna₁,cna₂,cna₃,cna₄,cna₅,cna₆}. By sampling k random centroids, we applied *Map* and *ReduceByKey* to the above DIA of CNA-feature vectors to update the centroids.

2.3 Find particle trajectories

The Vector Particle in Cell(VPIC) simulation studies how particles get accelerated in magnetic fields in plasma physics[2]. To get trajectories of the top m energetic particles, we used extra *Sort* by energy to the last time step file to get *particle_ids*. Then we used *Filter* to all time steps to extract those particles of interests. In order to accelerate the filtering operation, we used *Sort* by *particle_ids* before the filtering. The trajectory plots are omitted because of space limits.

3 RESULTS

We showed both correctness and scalability of our Thrill implementations. Figure2 and figure3 illustrated the time series and K-means clustering of the AMD states. These results were verified by scientists.

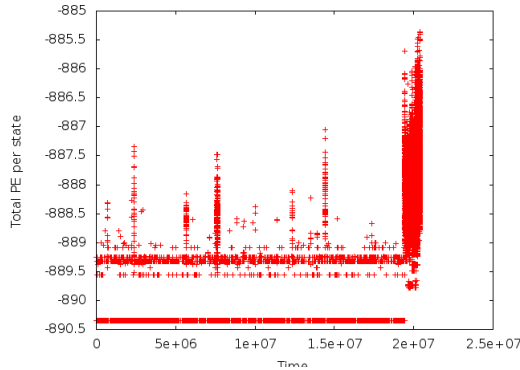


Figure 2: The time series of total potential energy shows an interesting transition starts around time 2×10^7 time-units.

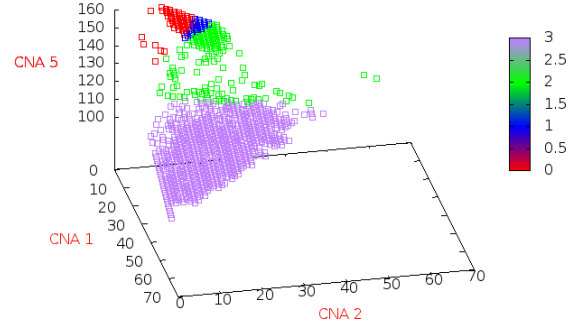


Figure 3: Four clusters found by the K-means clustering operation.

Figure4 illustrated that the Thrill implementation had strong scalability to some extent.

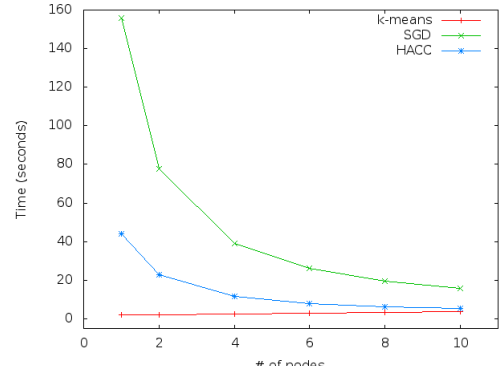


Figure 4: The Thrill implementations showed strong scalability up to 100 MPI ranks.

4 CONCLUSIONS

We explored the Thrill library to process and analyze simulation data. This helped scientists to deal with many files in parallel. The results were verified and displayed strong scalability to some extent.

ACKNOWLEDGMENTS

The authors would like to thank the Data Science at Scale summer school in the Los Alamos National Laboratory.

REFERENCES

- [1] Timo Bingmann, Michael Axtmann, Emanuel Jöbstl, Sebastian Lamm, Huyen Chau Nguyen, Alexander Noe, Sebastian Schlag, Matthias Stumpp, Tobias Sturm, and Peter Sanders. 2016. Thrill: High-performance algorithmic distributed batch data processing with c++. In *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 172–183.
- [2] Fan Guo, Yi-Hsin Liu, William Daughton, and Hui Li. 2015. Particle acceleration and plasma dynamics during magnetic reconnection in the magnetically dominated regime. *The Astrophysical Journal* 806, 2 (2015), 167.
- [3] Rao Huang, Li-Ta Lo, Yuhua Wen, Arthur F Voter, and Danny Perez. 2017. Cluster analysis of accelerated molecular dynamics simulations: A case study of the decahedron to icosahedron transition in Pt nanoparticles. *The Journal of chemical physics* 147, 15 (2017), 152717.