

Research Report

Monday 03/24/2014

Wei Wang

What I needed to do

- Needed to study the community detection code and conjugate code from PNNL
- Make them pass ROSE and/or PoCC
- Figure out ROSE transformation capability
- Others
 - Extending the workshop paper
 - PNNL benchmark (five sets of benchmarks)
 - Lulesh (using INRIA PPCG)

Progress & Problems

- Got Energy Consumption of the Community Detection code
 - Involved many code change, e.g. replace every occurrence of sync_fetch_and_add with its equivalent. Because ROSE does not support the functionality
 - Graph given too small, the energy info is not accurate (related to sampling frequency)
 - One relatively large 1D parallel loop contains indirect access, and pointers involving `std::map` type, very hard for PoCC to process
- Conjugate: `shared_ptr` class, `std::map<int,int>`
 - Did not know how to handle `shared_ptr` class in ROSE, no progress on this code

Progress & Problems

- ROSE Transformation/Auto-tuning Capability
 - Loop Transformations do not check legitimacy (tiling,unrolling,interchange) reference:
 - Recent literatures on Auto-tuning with ROSE exist, need to look into them in detail.

The plan

- Get to know the existing auto-tuning work with Rose and also the paper sent by David for PNNL meeting
- Looking for ways to generate code variants for community detection PNNL code.
- Looking into how to solve `shared_ptr` and `std::map` issue brought by PNNL code