PAD final project

Davide Neri

February 16, 2016

Master Degree in Computer Science and Networking A-Y 2015-16

Student:	Davide Neri
Instructor:	Marco Danelutto
Course:	Distributed Systems: Paradigms and models

Abstract

The report describes the final project addressing the **StreamCluster (SC)** application. The sequential code is provided by *C* code (PARSEC and RODINIA Benchmark Suite). The parallel version is implemented using **FastFlow** framework and *C++* programming language.

1 Design choice

The algorithm spend most of the time evaluating the gain of opening a new center (`pgain` function). For every new point, it weights the cost of making it a new center. If the heuristic determines that the change would be advantageous the algorithm reassigning some of the existing point to the new point that become a new center.
The points arrives in chuncks. For every chunk received, the application computes the centers points and put them into a data structure. When the stream of chunks are finished, the algorithm finds the center points among the centers stored.
A *task decomposition* pattern can be used to model parallel execution of chunks points of the stream.
For a single chunk can be used a *data decomposition* pattern to find the intermediate centers in the chunk. The application doesn't require any particular order of the chunks, because finding the intermediate centers points in different chunks is totally indipendent of the order in wich the chunks arrives.

The target platform is a shared memory machine (16 cores on host, 240 cores on Xeon phi machine).
The parameters can be tuned for having different possibility: parallelism degree in the task-farm patten and map parallel. With 1 worker is exploit only data decomposition, with 1 tjhread can be collapase using only taskfarm computation.
The task-parallelism is used to divide the stream of points in chunks and send the chunks to the workers. Each worker is in charge to compute a single chunk of points and find the centers, data parallel pattern is using loop parallelism.
Comparing parallel design patterns.

2 Compile,run and tests the project

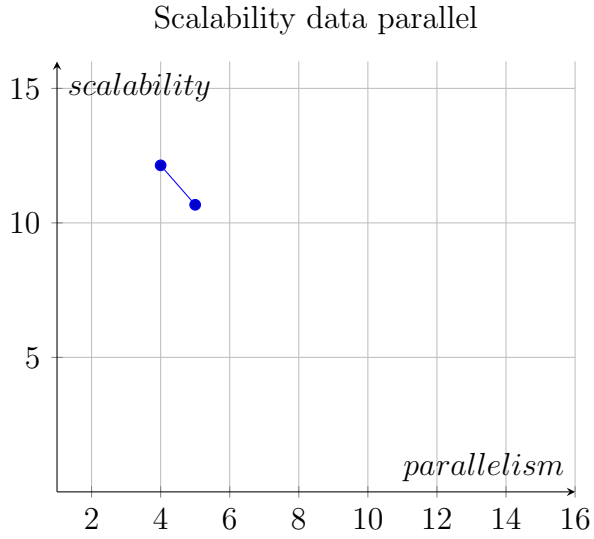
the short user manual should be detailed enough to enable me to run tests with your software (what should I compile, how, which are the parameters to use to launch experiments, how can I vary parallelism degree, which input files do I need, where are they)

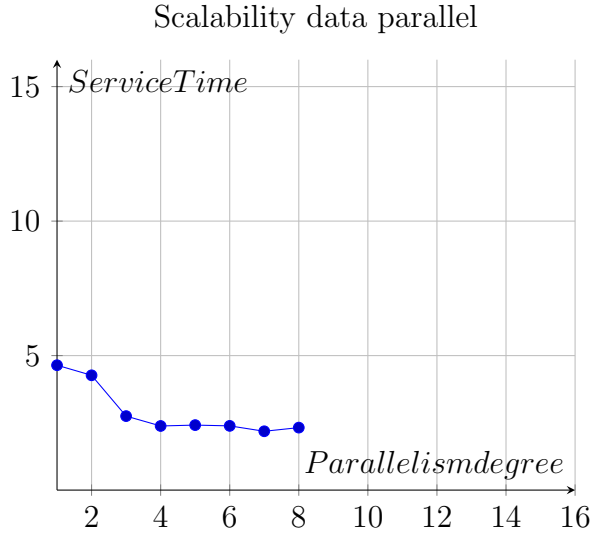
3 Experimantal results

In this section is analised three different performance measure for the two parallel solutions.

3.1 Data-parallel

scalability





References

[1] Mark Jelasity, Spyros Voulgaris, Rachid Guerraoui, Anne Marie Kermar- rec, Maarten van Steen, *Gossip-based peer sampling,* ACM Transaction on Computer Systems, October 2007.