

CS554 Project Ideas

MATRIX:BenchTEF - Benchmarking of the state-of-the-art Task Execution Frameworks for Many-Task Computing

Overview

Many-Task Computing (MTC) is a distributed paradigm that bridges the gap between High Performance Computing (HPC) and High Throughput Computing (HTC). MTC applications are structured as graphs of loosely-coupled short tasks, with explicit input and output dependencies forming the graph edges. Tasks may be uniprocessor or multiprocessor, compute-intensive or data-intensive. The number of tasks, quantity of computing, and volumes of data may be extremely large. MTC needs scalable task execution framework to handle billions of jobs/tasks.

Representatives of the task execution frameworks for MTC applications are Falkon developed by UChicago and ANL, MATRIX developed by our DataSys Lab, Sparrow developed by University of Berkeley, and Turbine developed by the Apache organization. Falkon has a centralized architecture and supports naive hierarchical scheduling. MATRIX is a distributed framework that applies work stealing to achieve distributed load balancing. Sparrow is a decentralized scheduling framework for sub-second tasks, which uses batch-sampling and virtual reservation to achieve distributed load balancing. Turbine is an excellent choice for developing applications that make use of a services-oriented architecture. This project aims to benchmarking and comparing all of these frameworks with detailed profiling about their performance and resource consumption under the same workloads. The results will help us extend MATRIX.

Relevant Systems and Reading Material

Falkon: <http://dev.globus.org/wiki/Incubator/Falkon>

MATRIX: <http://datasys.cs.iit.edu/projects/MATRIX/index.html>

Sparrow:

Paper: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-29.pdf>

Source code: <http://github.com/radlab/sparrow>

Turbine: <http://turbine.apache.org/>

Preferred/Required Skills

Required: Linux, scripting language

Preferred: MTC, C/C++, Java

Parameters

Different workloads, different scales

Metrics

Throughput, latency, efficiency, time and memory consumption

Project Mentor

Ke Wang, <http://datasys.cs.iit.edu/~kewang/>

Iman Sadooghi, <http://datasys.cs.iit.edu/~isadooghi/>