Simulation of a Volunteer Computing System

Anchit Gupta(13D100032), Anand Dhoot(130070009)
IIT Bombay

Abstract

As a part of this project, we simulate the volunteer computing system described as a part of "BOINC". The system involves several independent clients requesting for jobs from a central server and executing them. The system has to make the following important choices that determine its performance -

1. Scheduling of jobs on individual clients.
2. Selection of the job to be allotted to a client from the set of available jobs at the server.
3. Estimate the reliability of a client to assist decision making in (2).

We plan to study the effect of different policies for each of the above, using simulation and evaluate the performance of the system using basic but important metrics such as number of jobs completed, average time of completion, deadline slack etc.

1. Introduction

To quote Wikipedia, Volunteer computing is a type of distributed computing in which computer owners donate their computing resources (such as processing power and storage) to one or more "projects". The paper describes BOINC - Berkeley Open Infrastructure for Network Computing - which is "a middleware system" for volunteer computing.

In such a system, the server is a repository of jobs from one or more than one projects. The volunteer or the client requests jobs to the server and is allotted a suitable job, decided by some algorithm at the server. The client itself performs CPU scheduling, over and above the OS's scheduler.

Through this project, we plan to implement various strategies for the server and client side scheduling using a Discrete Event Simulator and evaluate them based on various metrics to present a comparative study of the same. In this document, we describe the System Model and the Assumptions in Section 2, Metrics for evaluating and comparing various algorithms in Section 3 and the need for such systems and the current status of BOINC in Section 4. Finally, this project proposal ends with a Bibliography.

2. System Model & Assumptions

We have simplified the model proposed in the paper so as to make it more viable for implementation. We have one main server in our model and an arbitrary number of clients. Every job would have the following attributes -

- Duration - CPU cycles required to complete the job
- Deadline - Time by which it must be completed

The basic parameters of intrest here are the distribution from which the job durations as well as job deadlines come from. These might effect the performance of the various metrics. The means and variances of the same are crucial here.

We also assume that server-client communication requires negligible time.

The system has to decide among various scheduling policies to determine its performance. The system has to make a choice at the server and at all the clients. The set of assumptions and available choices at the server and client are as below.

2.1. Server

We assume that the server has a very large buffer (as compared to the client buffer) of available jobs constituting a single project. As soon as a job gets allotted to a client, another incoming job fills the empty slot in the buffer.

When the server receives a request for a job, the server choses the job to be allotted using one of the scheduling schemes described below.

- FCFS - First Come First Serve
- Shortest Deadline First
- Shortest Duration First

2.2. Client

The client requests jobs from the mainframe until its own buffer is full and complete them, making a new request every time it has an empty slot in the buffer.

It has to decide upon the scheduling algorithm - Amongst the jobs in its buffer which to run and for how long. The possibilities for this scheduling policy are -

- FCFS - First Come First Serve
- Round Robin scheduling
- Weighted RR with deadlines as weight
- Fair-Share Scheduling

It must be noted that the server and the client scheduling algorithms may differ from each other. The choice of scheduling algorithms has a significant impact on the performance of the system and this would be measured using various metrics, described in detail in the subsequent section.

3. Metrics

The paper describes several metrics to measure the performance of the network. The ones applicable in our scenario, with our assumptions, are -

1. Idleness of a client CPU - Ideally, should be zero.
2. Wastage of Client CPU time - because of jobs missing deadline

Some other metrics that we plan to use to evaluate network performance -

1. Average number of jobs completed in a unit time, by all clients
2. Average Deadline Slack - Average time difference of job completion from the deadline, only for jobs that get completed within deadline. High value for this indicates that the clients were able to complete jobs well in advance of the deadline.

4. Real-life Applications & Current Status of BOINC

With the vast amount of data available today and the increasing need for computing power, volunteer computing systems such as BOINC is a way of crowdsourcing computing power. Not only does this satisfy the demand for computing power, but also enables anybody owing a computer to put it to good use when they do not need it themselves, probably earning some reward in return.

As of today, the BOINC project has a total of 295,701 active volunteers contributing 5,432,061 computers with a 24-hr average computing power of 11.841 PetaFLOPS, which is comparable to the World's largest supercomputer.

5. References

[1] D. Kondo, D. P. Anderson, and J. M. Vii. Performance evaluation of scheduling policies for volunteer computing. Dec 2007.

[2] Wikipedia. Supercomputer.

[3] Wikipedia. Volunteer computing.