Thesis Title

Joe Bloggs

January 26, 2016

Contents

Abstract

Abstract goes here

Keywords:

keywords, go, here.

1

Introduction

1.1 Introduction

The proliferation of streaming data sources such as optical sensors, health tracking sensors, and social networks has increased greatly in the previous 10 years, and with it the the desire to has the desire to understand and analyse this streaming data. The increase in computing power has similarly allowed computing simulations to become increasingly useful as a method of predicting and understanding various processes occurring, whether they be physical, social or purely theoretical.

This thesis seeks to address the problem of having multiple, potentially interrelated, heterogenous computational tasks that are long running or streaming tasks. The problem is broken up into the following parts:

1. Load balancing heterogenous long-running jobs between heterogenous computing hardware
2. Providing an api for job dependency graph
3. Providing an inter-job communication api
4. Providing an intra-job communication api

The system is built using Extempore, a high performance lisp environment. This system was originally built for use in real-time cyThe benefits of being built in Extempore include:

- Providing a direct access to system devices via its C interoperability
- Extempore Just-In-Time (JIT) compiles all code via the LLVM compiler to x86 bitcode, providing high performance program execuition
- Dynamic code redefinition, allowing extempore processes to be retargeted with new functionality as is required

2

Literature Review

2.1 Load Balancing

Sees load balancing as a task done once by a job router, does not include the idea of redistributing the jobs once they are scheduled. **?**

"A load balancing algorithm consists of three components.

1) The information policy specifies the amount of load and job information made available to the job placement decision maker(s), and the way by which the information is distributed. 2) The transfer policy determines the eligibility of a job for load balancing based on the job and the loads of the hosts. 3) The placement policy decides, for eligible jobs, the hosts to which the jobs should be transferred." **?**

2.1.1 Dynamic vs Static

?

Dynamic vs static definition **?**

?

Queuing system M/M/n

?

2.1.2 Centralised vs Decentralised

"In a centralized load balancing system, the global load information is collected at a single node, called the central scheduler. The other nodes, called

local nodes, send their load status messages to the central scheduler. All load balancing decisions are made at the central scheduler based on the collected messages.

In a decentralized load balancing system, each node broadcasts its load information periodically to the other nodes to update their locally maintained load tables. Every node performs its node selection action. This can be done by having every node keep track of the global system load status continually. " Lan and Yu [1995]

"The scalability of the centralized algorithms, GLOBAL and CENTRAL, however, is also very good, at least up to 49 hosts. In constrast, DISTED scales quite badly. We can see two conflicting factors in action. On the one hand, an increase in system size makes it easier to find a lightly loaded host. On the other hand, the message overhead per host grows linearly with system size, The composite effect is a moderately rising curve for the normalized response time." **?**

Centralised **?**

2.1.3 Homogenous vs Heterogenous Jobs

2.1.4 Homogenous vs Heterogenous Hardware

uses homogenous system **?**

2.1.5 Measuring Load

Traditional High Performance Computing (HPC) tasks are typically CPU bound, requiring processing power to (for example) advance their simulations forward in time, whilst Big Data tasks are typically memory and storage bound.

Measuring multiple load sources **?**

2.2 Dependency Graph

2.3 Communication Systems

2.3.1 Inter-Job

Actor

Channel

2.3.2 Intra-Job

MPI

3

System Description

3.1 Architecture

3.2 Master Controller

3.3 Job Controller

3.4 Worker

4

System Demonstration

5

Discussion

6

Conclusion

Reference List

Youran Lan and Ting Yu. A dynamic central scheduler load balancing mechanism. In *Computers and Communications, 1995., Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on*, pages 734–740, Mar 1995. doi:10.1109/PCCC.1995.472412.

Appendices

A

Appendix

Todo list