Master's Thesis Plan

Kristian Hartikainen (222956)
kristian.hartikainen@aalto.fi

May 15, 2015

1 Summary

This document is meant to cover all the general stuff that needs to be considered before and during writing my master's thesis. In addition to the technical planning of the thesis, this plan also includes the general matters that might affect the thesis writing.

The thesis is done under the Embedded Systems Group in Aalto University with connections to the ParallaX research project. The actual starting point for this thesis is defined by the recent bachelor's theses (Kiljunen, Teemaa) and master's theses (Hanhirova, Saarinen, Rasa), written for the Embedded Systems Group.

In the background section I present the outline for the larger research area where the thesis sets in. In the followed three chapters I present the goals for this thesis, the methods used to achieve those goals, and the material to be used. In the last chapters, I will go through the general matters possibly affecting the thesis writing, such as the material and resources in use, general challenges, the actual work plan, and the dissemination. In the appendix I will also present a scaffolding for the actual thesis structure, and the key literature for the thesis.

2 Background

The amount of data being transmitted and processed in the world is growing rapidly, and the restrictions, mainly the increasing energy consumption, in the microprocessor evolution has led in the development of new type of Multiprocessor System-on-Chip devices (MPSoC).

While these MPSoC devices seem promising in terms of the needed increase in computational performance, they also bring out new challenges in the software development. The traditionally used parallel programming models, that are based on thread parallelism, don't meet the needs of processing the heterogeneous stream-like data.

Several different programming models/frameworks have been proposed to allow

parallel software development. It is still unclear, however, how to efficiently allocate the computing resources in these models.

Also, the analysis of parallel programs is a tedious task, partly due to the non-deterministic behaviour of the parallel hardware, the complexity in the memory accesses and the delays from the communication.

TODO:

- Mikä on oikeastaan on se yleinen ongelma täällä? Ilmeisesti multiblade juttu itsessään on melko selkeää, samoin kun eventtipohjainen parallelismi. Onko stream-processing, yhdistettynä näihin kahteen, se joka tuo ongelman tähän hommaan?
- Miten analyysi liittyy näihin asioihin? Onko haasteet analyysin tekemisessä itsessään jo ongelma vai johtaako ne vain epäsuorasti ongelmiin esim. eri menetelmiä verrattaessa??

3 Objectives

The objectives of this thesis is to understand the allocation of computing resources in a streaming multi-blade computing setup. This objective includes an implementation(s) of efficient scheduling/load-balancing for multi-blade processing, as well as the analysis of those implementation(s). Hopefully we can answer the question, how should the multi-blade load-balancing be implemented? Should it be implemented on top of OpenMP/OpenEM/MPI?

TODO:

- Tavoitteet vielä melko epäselvät, ainakin vaikeasti muotoiltavissa.
- Kuten BG:ssä, mikä on se oikea ongelma?
- Onko dynaaminen/staattinen analyysi vain väline näiden tulosten evaluointiin, vai onko analyysissä itsessään jotain kysymysmerkkejä joihin halutaan vastauksia??
- Ilmeisesti pääkysymys on, että miten load-balancing toteutetaan multi-bladelle, toteutetaanko suoraan OpenEM:ään/OpenMP:hen vai jollain muulla tavalla?

4 Methods

The base for the methods will be somekind of event-driven parallelism solution. The processing-queues will probably play a big role when trying to answer the multi-blade load-balancing question. Most likely, either OpenEM or OpenMP will be extended.

The results of the experiments will be evaluated both qualitatively and quantitatively. The hardware used in the experiment (BCN/Cavium OCTEON II CPU) is sort of a black box and thus the focus will be heavily on the quantitative results.

Program analysis will be done both statically and dynamically. The main simulation (dynamic analysis) tool will be Performation Simulation Environment (PSE). Perf, rapl.

TODO:

- Workload generation?
- Ei-jaettu muisti??
- Tehdäänkö analyysi sekä staattisesti että dynaamisesti?
- Taas, mikä on se varsinainen haaste tai kysymys?
- Kuten edellä, onko analyysi itsessään jonkinlainen kysymysmerkki?

5 Material and Resources

Vesa Hirvisalo will work as an advisor for this thesis. The supervisor is still yet to be decided. Hardware and frameworks to be used in the thesis is listed below.

BCN/Cavium OCETON II: Mitä tästä saa sanoa?

Intel DPDK: "DPDK is a set of libraries and drivers for fast packet processing. It was designed to run on any processors knowing Intel x86 has been the first CPU to be supported. Ports for other CPUs like IBM Power 8 are under progress. It runs mostly in Linux userland. A FreeBSD port is now available for a subset of DPDK features."

OpenEM: "Open Event Machine (OpenEM or EM) is an architectural abstraction and framework of an event driven multicore optimized processing concept originally developed for networking. It offers an easy programming concept for scalable and dynamically load balanced multicore data plane applications with a very low overhead run-to-completion principle."

OpenMP: "OpenMP (Open Multi-Processing) is an API that supports multi-platform shared memory multiprocessing programming in C. OpenMP is an implementation of multithreading, a method of parallelizing whereby a master thread (a series of instructions executed consecutively) forks a specified number of slave threads and the system divides a task among them." Note the words shared memory.

TODO: Mitä muuta? PSE?

6 Challenges

The BCN/Cavium OCTEON II hardware used in the experiments is going to be hard to analyze.

I haven't officially received my Bachelor's degree, since I'm one course short from it. The official degree is needed to get the Master's thesis topic approved by the school.

However, this shouldn't be a problem, as I have discussed with the student counselor Elsa Kivi-Koskinen, and the course is under progress. I have done the course exam already, and right after I finish the course project, it's done from my part. Then the Bachelor's degree still have to be approved by the school council.

7 Work Plan

Quite traditional iterative approach will be used to move forward with the experiments and thesis writing. Getting familiar with the setup of the hardware used in the experiments has been going on for the last few weeks. After the hardware setup is all clear, I will begin constructing the expertiment. The construction follows with the measurement and analysis of the construction.

After the first prototype, the last two stages, construction and analysis, will be redone according to the results of the previous iteration. As many iterations are done as are needed, within the the time I have.

8 Schedule

- Week 19: Scaffolding for the thesis plan, figure out first iterations of the background information, and write the first iteration for all the general parts of the plan.
- Week 20: More studying about the topic. Figure out the first iterations for the methods and objectives parts for the plan. Rewrite other parts, mainly the background, as the topic get clearer.
- Week 22: Final version of the plan. Be in a situation where the actual construction of the experiment can be started full time.
- Week 26 (End of June): Be ready with the initial construction of the experiment. Before Vesa Hirvisalo and Jussi Hanhirova leave for the summer vacation, be in a situation such that the thesis can be taken forward without external help.
- Week 26 (End of July): Be ready with the analysis and measurements for the initial experiment.

- Week 35 (End of August): Be ready with the first iteration of the experiment.

9 Dissemination

This thesis is related to the ParallaX research project. Also, the thesis hopefully gives some background to Jussi Hanhirova's doctor's thesis.

Appendices

A Thesis structure

- Abstract
- Preface
- Contents
- Abbreviations

1. Introduction
   - Research Problem
   - Contributions
   - Structure
2. Parallel Programming Methods
   - task, event, etc parallelism
   - OpenEM/OpenMP/...
3. Analysis of Parallel Programs
   - The goal of the Analysis
   - Tools used
4. Load-balancing
5. Experiments
   - Setup
   - Results
   - Conclusions
6. Discussion
   - Challenge
   - Discoveries
   - Related Work
   - Future Work
7. Conclusions

B Key Literature

- Bachelor's theses

  Teemaa

  Kiljunen

- Master's theses

  Hanhirova

  Rasa

  Saarinen

- Cavium Docs

- Static and Dynamic Analysis

  Wilhelm et al. The Worst-Case Execution-Time Problem—Overview of Methods and Survey of Tools

  Banks et al. Introduction to Discrete-Event Simulation

  Gustavsson et al. Timing Analysis of Parallel Software Using Abstract Execution ??

  Fujimoto et al. Parallel Discrete-Event Simulation ??

- Allen & Kennedy Jotain yleistä load-balancingiin liittyen