CSC 714 Real Time Systems

Project Webpage: http://www4.ncsu.edu/~abahman/CSC714/

Analyzing the Effect of Predictability of Memory References on WCET

Amir Bahmani
Department of Computer Science
North Carolina State University
abahaman [at] ncsu [dot] edu

Vishwanathan Chandru
Department of Computer Science
North Carolina State University
vchandr6 [at] ncsu [dot] edu

Objective

Analysis of the effect of systematic mappings of tasks/jobs to tiles on WCET considering the base of memory predictability.

Problem Statement

Platforms consisting of several cores (multicores) and more than a dozen of cores (many-cores) have nowadays become the mainstream in many scientific areas, most notably high performance computing, while are the new frontier technology in others like real-time embedded systems. Along with positives of whole lot of processing power and increased system availability, this comes with multiple latencies, especially in terms of memory accesses as multiple cores try to access the memory at the same time. It becomes critical as we talk of hard real time systems where predictability is very important. This becomes even worse with a multi-processor board like Tilera which is having processors in order of 50's and above, NoC added to that unpredictability. We performed a short literature survey to understand the delay or unpredictability added due to contention of resources especially memory and found a few things. First was the contention to access the same memory when memory is not divided into banks or bank aware allocation is not performed. We found two analysis procedures in this context request driven analysis and job driven analysis. We also found about various inter-bank and intra bank interferences resulting to unpredictability [1]. We found one of the bank aware allocator called PALLOC which alleviated this problem to a certain extent[2]. But static/dynamic partitioning alone cannot solve this problem as when we have a board like Tilera with large number of cores and NoC packets being sent across for memory request and communication predicting becomes tricky [3]. We need to consider worst case latency in this scenario using per pattern analysis. Also if we consider involvement of multiple memory controllers and live process migration for load balancing situation becomes even trickier. So we concluded that we need a multi faced solution which involves distributing memory requests across controllers to reduce latency, static/dynamic banking of memory and optimal placement of processes among cores to ensure minimum n/w traffic and minimum latency, thus a more predictable and reliable WCET.

In this project, we propose to analyze the effect of mapping related tasks/jobs in close proximity on memory predictability considering bandwidth of memory controller, thus on WCET. As part of this framework we will explore the possibility of implementing space separation for memory access request by way of coloring the MCs and distributing access across MCs of different colors or by using available bandwidth in each controller. One done, this can help in adding substantial predictability for WCET. We will also analyze the effect of network contention depending on proximity of memory controller. For mapping of the tasks(if time permits), it can be

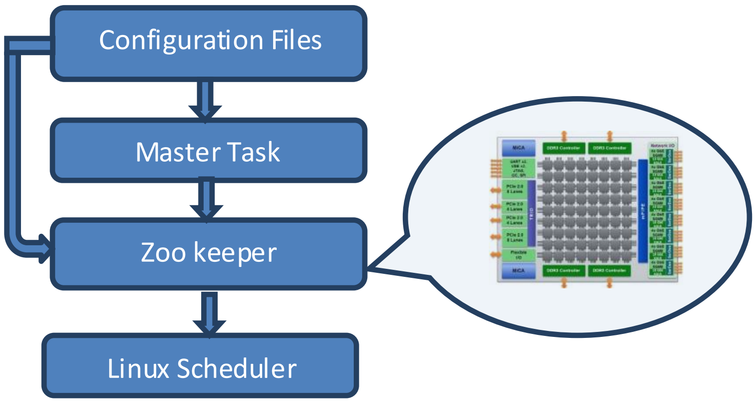

Fig 1 : Design of proposed framework

done via static/dynamic mapping function known to all the processes and all the processes compute there affinity to a certain core/processor based on mapping function. Figure 1 shows our proposed framework. In this framework, we assumed that the user provides characteristics of real-time tasks (e.g. deadline, period, resource dependency, etc.) to ***MasterTask*** which is responsible to create jobs and mapping function that we called it ***Zoo Keeper***. After implementing the remapping of memory accesses and mapping the tasks, we can judge the efficiency of mapping based on the percentage and degree of deadline misses.

Challenges:

1) Figuring out the proximity of memory controllers
2) Figuring out the targeted MC from the physical address.
3) Figuring out the physical address from the virtual address.
4) Figuring way to modify cpu frequency on the fly.
5) Figuring out way to query available bandwidth with a MC.
6) Figuring out a way to remap the memory access request to a different MC.
7) Verifying the compatibility of **LITMUS** with Linux running on Tilera.
8) Absence of memory banking could make the mapping less efficient.
9) Potential cross interference caused due to contention for memory access by multiple tasks.
10) Defining Test Scenarios keeping in mind RAW, WAW and WAR patterns to reduce caching.
11) Devising good heuristic algorithms considering maximum number of factors.
12) Requests going to controller not in proximity of tile/processor.
13) Negative impacts of the mapping algorithm.

In this project we will be primarily focusing on mapping of memory accesses to certain MCs based on factors such as proximity, memory banking etc. In case time permits, we will analyze the mapping to tasks to various tiles based on various factors.

CSC 714 Real Time Systems

Project Webpage: http://www4.ncsu.edu/~abahman/CSC714/

Milestones:

The below mentioned milestones are tentative.

Ramp up on Tilera architecture and APIs (Both)	**24tn March 2014**
Figuring out base parameters for coloured malloc(Vishwanathan)	**24th March 2014**
Figure out way to remap the memory accesses (Vishwanathan)	**31th March 2014**
Figure out a way to figure out the corresponding MC from physical address (Amir)	**31st March 2014**
Memory Mapping strategy (Amir)	**7tn April 2014**
Strategy for contention analysis (Vishwanathan)	**7tn April 2014**
Malloc implementation (Both)	**14tn April 2014**
Test Cases Design and Implementation (Both)	**14tn April 2014**
Task Mapping Algorithm and implementation (Amir)	**21st April 2014**
Task Mapping algorithm validation(Vishwanathan)	**25tn April 2014**
Presentation(Amir)	**27tn April 2014**
Final Report(Both)	**1st May 2014**

References:

[1] Hyoseung Kim , Dionisio de Niz, Bj □ orn Andersson† , Mark Klein†, Onur Mutlu, Ragunathan (Raj) Rajkumar , Bounding Memory Interference Delay in COTS-based Multi-Core Systems in COTS-based Multi-Core Systems

[2] Heechul Yun , Renato Mancuso , Zheng-Pei Wu , Rodolfo Pellizzoni, PALLOC: DRAM Bank-Aware Memory Allocator for Performance Isolation on Multicore Platforms

[3] Borislav Nikoli ́ c, Patrick Meumeu Yomsi and Stefan M. Petters, Worst-Case Memory Traffic Analysis for Many-Cores using a Limited Migrative Model