

A Task-centric...(PACT'09)

```
@inproceedings{kelm-pact09,
author = {John H. Kelm and Daniel R. Johnson, Steven S. Lumetta and Matthew I. Frank and Sanjay J.
Patel},
title = {A Task-centric Memory Model for Scalable Accelerator Architectures},
booktitle = {PACT '09: Proceedings of the 18th international conference on Parallel architectures
and compilation techniques},
year = {2009},
pages = {???-???},
location = {Raleigh, North Carolina},
}
```

1024-core で、MIMD(SIMDっぽくない、従来不向きとされていた) プログラムを
実行するためのメモリ管理の話。

visual program における共有データへのアクセスを解析

全体のキャッシュを H/W で管理するのではなく S/W で管理するためのプロトコルを規定
キューによるタスク管理で complete なタイミングでバリアを実現 ->[6]

結果は他のタスクマネジメント / メモリマネジメント手法との比較ではなかった。
命令のロードとかがってどうしてるんだろう？

以下論文より

Abstract

- task-centric memory model
 - uses a software protocol
 - working in collaboration with hardware caches
 - to maintain a coherent, single-address space view of memory w/o HW support
- for 1024-core MIMD accelerator; Rigel

Introduction

- task-centric memory model
 - hw/sw protocol for maintaining a coherent view of shared memory for accelerator
- visual computing が対象
 - a form of bulk sync. processing を使って開発される
 - barrier の間 (interval) は独立した並列処理の単位 (task) が並列に実行
 - analysis によると well-structured sharing patterns である
- DSM と似てる。違いは
 - private \$ をもった 1chip のプロセッサであるため shared-global \$ へのアクセスコストが小さい
- 1024-core の accelerator である Rigel が対象
 - a single cacheable address space
 - w/o hardware-enforced \$ coherence across all cores on the chip
- Contributions
 - data sharing patterns for class of emerging workloads の観察
 - a scalable task-centric memory model (for 1000-cores)
 - optimization
 - prefetching from DRAM is unimpeded and most beneficial to perf.
 - overhead of the task-centric model can be minimal

Motivation/Background

- data-parallel execution model だけじゃなくて irregular task-parallel computation も考えたい
- Application Characterization
 - Parallelism Structure(Programming styles)
 - bulk sync. processing
 - - the tasks exchange little or no data within an interval
 - - at the barrier, modified shared data is made globally visible
 - - mostly-data-parallel, task-based shared-memory programming model, coherence management is required to enable sharing
 - - do not depend on the HW support
 - the programmer's attempt to create scalable code(minimum sharing)
 - Sharing Patterns
 - sync. characteristics
 - benchmarks
 - - MRI benchmark(VISBench)
 - - CG, sobl edge detection, k-means clustering, DMM(Rigel kernle benchmark suite)
 - - GJK collision dtection benchmark(a freely-available seq.)
 - - Heat (Cilk)
 - Fig.1 and Fig.2 は , the freq. of non-private loads/stores
 - - the majority of non-private loads are reads to data produced before the current interval began
 - - both conflict reads and writes to data shared
 - Accelerator Workload Characteristics
 - characteristics
 - - read shared data is present within an interval
 - - sync. is coars-graind
 - - small amounts of write-shared data within an interval
 - - Fine-graind sync. (ex. atomic updates to shared data) is present but rare
 - - write sharing within an interval is rare
 - little coherence management is required
 - Cache Coherence Management
 - weakly consistent memory models
 - explicit local and global memory operations
 - task-based programming model
 - as a substitute for HW \$
- Related Work
 - bulk-sync. parallel(BSP) model \$ \$ CUDA, OpenCL
 - OpenMP, Intel's TBB
 - Workload(PARSEC, ALPBench)
 - Memory Models

Rigel Architecture and Task Model

今日のつばやき

- MacOSX の iCal 使いになってしまった . およよ (Fri Aug 21 15:55:44 2009)
- MindNode を使ってみてる . かつこいいけど , 自動で位置調整してくれないのは , 若干不便 . (Fri Aug 21 14:29:22 2009)
- さすがに , 昨日の残りのドンブリに入ったビールは飲めないです . (Fri Aug 21 14:04:57 2009)
- A Task-centric Memory Model for Scalable Accelerator Architectures を読む (Fri Aug 21 11:22

:18 2009)

- PACT'09 の論文って読めるのも多いのね . (Fri Aug 21 11:21:39 2009)