Ontology-based Knowledge Representation for PORPLE

Feifei Wang
fwang12

Introduction

- **PORPLE** [Chen'14]
  - a portable data placement engine
    - *memory system*
    - *data access pattern*
- **Ontology** [Gruber'95]
  - a general-purpose model for knowledge

Memory System

- die = 16 tpc; tpc = 1 sm; sm = 32 cores; membus = 48 bytes;
- globalMem 8 Y rw na 5375M 128B 32 ? 600clk <L2 L1> <> die 1 <0.1 0.5> warp{address1/blockSize!=address2/blockSize};
- L1 9 N rw na 16K 128B 32 ? 80clk <> <L2 globalMem> sm 1 ? warp{address1/blockSize!= address2/blockSize};
- L2 7 N rw na 768K 32B <32|4> ? 390clk om om die 2 ? warp{ thread1/<32|4>!=thread2/<32|4> || address1/blockSize != address2/blockSize }; //address1 and address2 are the transformed addresses in L2
- constantMem 1 Y r na 64K ? 32 ? 360clk <cL2 cL1> <> die 1 ? warp{address1 != address2};
- cL1 3 N r na 4K 64B 32 ? 48clk <> <cL2 constantMem> sm 1 ? warp{address1/blockSize!= address2/blockSize};
- cL2 2 N r na 32K 256B 32 ? 140clk <cL1> <constantMem> die 1 ? warp{address1/blockSize!= address2/blockSize};
- readonlyMem 11 Y r na 5375M 32B 32 ? 617clk <L2 tL1> <> die 1 <0.1 0.5> warp{address1/blockSize!= address2/blockSize};
- sharedMem 4 Y rw na 48K ? 32 32 48clk <> <> sm 1 ? block{word1!=word2&&word1%banks ==word2%banks};
- tL1 6 N r na 12K <32B 4> 4 ? 208clk <> <L2 textureMem> sm 1 ? warp{ thread1/4!=thread2/4 || address1/blockSize.x!= address2/blockSize.x}; //address1 and address2 are the transformed addresses in tL1
- textureMem 5 Y r na 5375M na 4 ? 617clk <L2 tL1> <> die 1 <0.1 0.5> ?;
- textureMem 5 om om 1 128ME 32B om ? ? om om om om om warp{thread1/4!= thread2/4 || address1/blockSize != address2/blockSize};
- textureMem 5 om om 2 <64KE 64KE> <16B 2> om ? ? om om om om warp{thread1/4!= thread2/4 || address1.x/blockSize.x!=

Data Access Pattern

- constant 99999999 0 384 99999999 0 0 99999999 0 0
- global 64 0 0 64 0 0 4 0 0
- readonly 32 16 0 32 0 0 99999999 0 0
- texture1D 32 16 0 32 0 0 99999999 0 0
- texture2D 64 0 0 64 0 0 99999999 0 0
- Shared: 512 512 32
- - global 64 0 0 64 0 0 4 0 0
- - readonly 16 0 0 32 0 0 99999999 0 0
- - texture1D 16 0 0 32 0 0 99999999 0 0
- - texture2D 64 0 0 64 0 0 99999999 0 0
- shared 32 16 1

Objective

- Use ontology to systematically and formally represent and reuse knowledge about GPU memory systems and data access patterns.
  - generality
  - extensibility
  - reusability

Motivation

- **OpenK** [Liao]
  - ontology-based program analysis and optimizations in High Performance Computing (HPC)
  - software, hardware, and optimizations
  - open, extensible, and reusable

Challenge

- Understanding ontology, e.g.,
  - classAssertion(‘GlobalMemory’,’M2075globalMem’)
  - propertyAssertion(‘hasUpperLevel’,’globalMem’,’L2’)
  - propertyAssertion(‘hasLowerLevel’,’L2’,’globalMem’)

Solution

- Learn examples by using Protege

Lessons

- Think comprehensively
  - functional property
  - inverse functional property
  - transitive property
  - symmetric property
  - asymmetric property
  - and etc.

Results

- Demo
  - MSL https://github.com/FeifeiWang7/CSC766/blob/master/Ontology/MSL_To_Ontology/output.txt
  - Data Access Pattern https://github.com/FeifeiWang7/CSC766/blob/master/Ontology/PORPLE_To_Ontology/output.txt

Future Plan

- Fix two bugs of the code
  - differentiate different scenario usage of memory
  - add OWL prefix in the code
- Transform more format to ontology

References

- Chen, G., Wu, B., Li, D., & Shen, X. (2014, December). PORPLE: An Extensible Optimizer for Portable Data Placement on GPU. In Microarchitecture (MICRO), 2014 47th Annual IEEE/ ACM International Symposium on (pp. 88-100). IEEE.

- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing?. International journal of human-computer studies, 43(5), 907-928.

- Liao, C., Shen, X., & Zhao, Y. OpenK: Towards Open and Resuable Ontology-Based Program Analysis and Optimizations in HPC.

Thank you!