A Framework for Resource Aware Multithreading

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

Problem Formulation

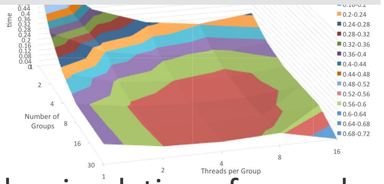
Current Data movement Strategy:

- Oblivious to the execution pattern of neighboring threads

Parallelism Vs Locality

- Accessing closed page is expensive both for performance and power
- Serialization of concurrent threads in memory banks

Parameter sweep between threads aggregated by groups, size of groups and time

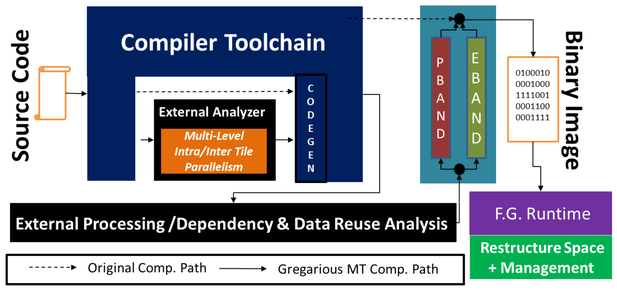


Aiming toward Exascale

- Stencil computation is at the heart of the simulation of several physical phenomena that is of scientific interest such as the Data Analysis Proxy App used in the Center for Exascale Simulation of Advanced Reactors.

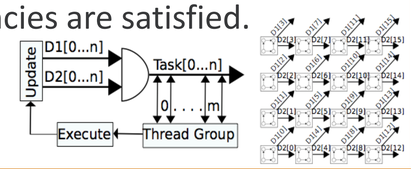
Group Locality

Threads collaborate to improve locality using static information gained form the compiler.



The Fine Grain Execution Runtime

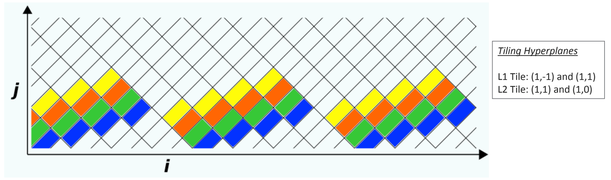
- Dependencies and tasks (i.e. tiles) are represented by bit masks.
- Dependency bit mask are AND'ed to get task mask.
- Inner tiles run in parallel as dependencies are satisfied.
- Take advantage of outer tile locality.
- Hierarchical



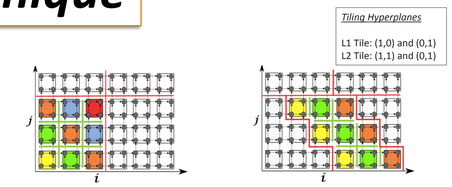
An Efficient Tiling Technique

The Jagged Tiling technique

- Designed to improve intratile parallelism.
- Uses Polyhedral formulation to create the different tiling levels



Hierarchical Parallel Start will Jagged Diamond Tiling

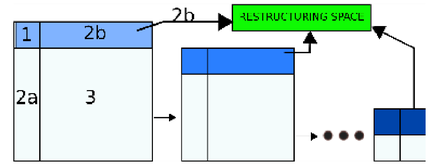


Classical Multi hierarchical tiling v.s. normal jagged tiling

- Two flavors:
  - ***Jagged Tiling*** (pipeline start codes) & ***Jagged Diamond Tiling*** (parallel start code)

The Data Restructuring Framework

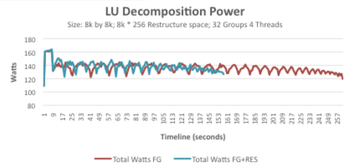
- Mem. access latency dependent on access pattern.
- Access pattern = Access Function + Mem. Mapping
- Based on Reuse of Data
- Mapping of one address space to the other using invertible transformation matrix.



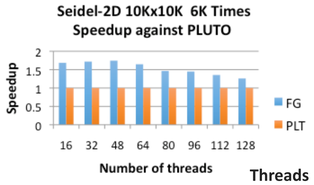
Restructuring of a LU Decomposition Example

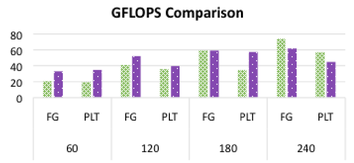
Initial Results

Experimental Platform:
Intel Xeon Phi 7110P
61 Core at 1.1 GHz
L1 cache 32KB, L2 cache 512 KB



Threads per Group: 4
FG : Fine-grained execution
FG+RES: Fine-grained execution combined with restructuring. Restructuring is based on reuse of data.





Threads per Group: 4
FG : Fine-grained execution of tile inside L2 using Jagged Tiling
PLT: PLUTO generated tiled OpenMP code.

Conclusions

- Increase performance by better utilization of resources
- Overall better energy profiles due to reduction in time and a negligible changes in power consumption
- Future work includes more in-depth characterization & memory structure mapping

U.S. DEPARTMENT OF ENERGY

Authors: *Sunil Shrestha[1] , Joseph Manzano[2] , Andres Marquez[2] , John Feo[2] and Guang R. Gao[1]*
[1]University of Delaware [2]Pacific Northwest National Laboratory

This research was supported in part by DOE ASCR XStack program under Awards DE-SC0008716, DE-SC0008717