GMetis - Xeon Phi

David Pereira Rui Brito

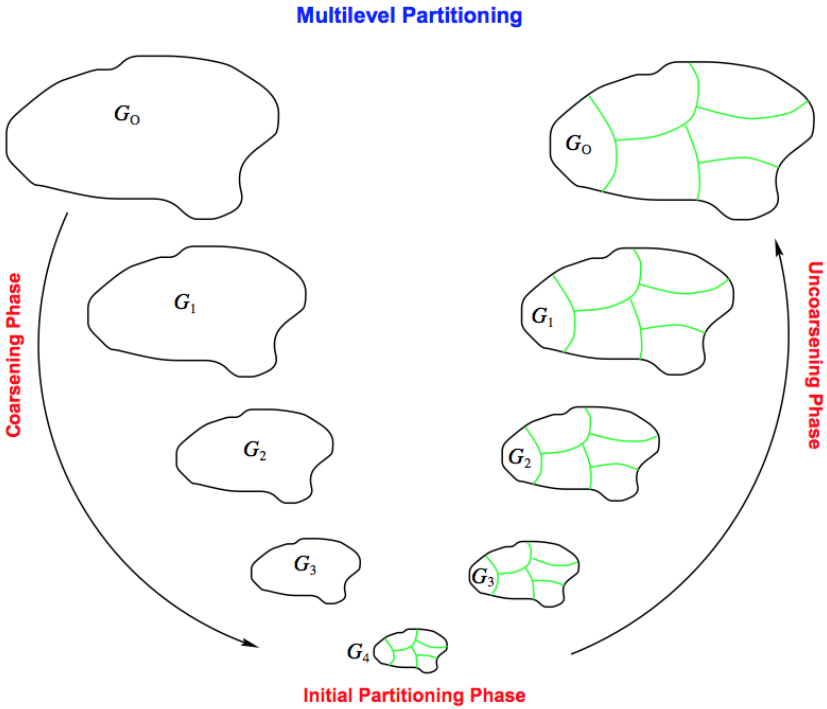
August 8, 2013

Outline

Introduction

- GMetis is a graph partitioning application which uses the Galois framework
- Consists of three major phases
  - Coarsening
    - ★ Find matching nodes
    - ★ Coarsen Graph
  - Initial Partitioning (Clustering)
  - Refinement

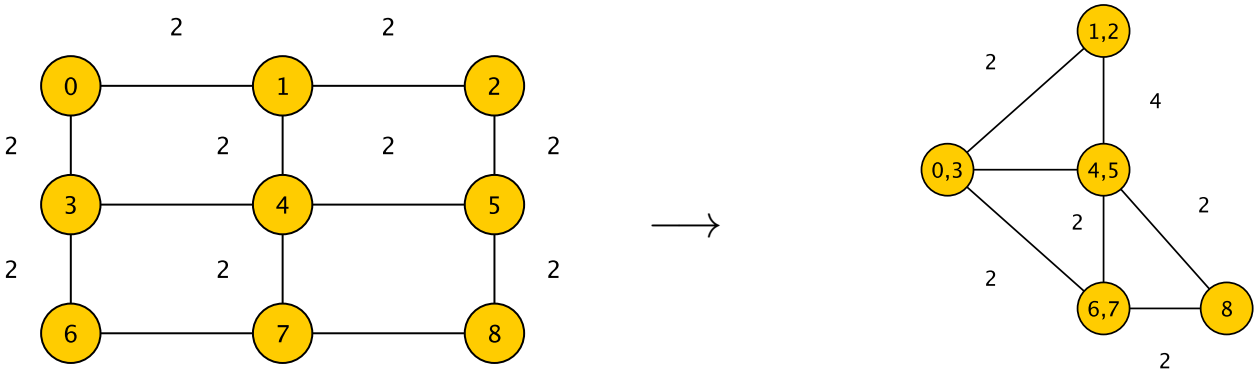
Algorithm Description



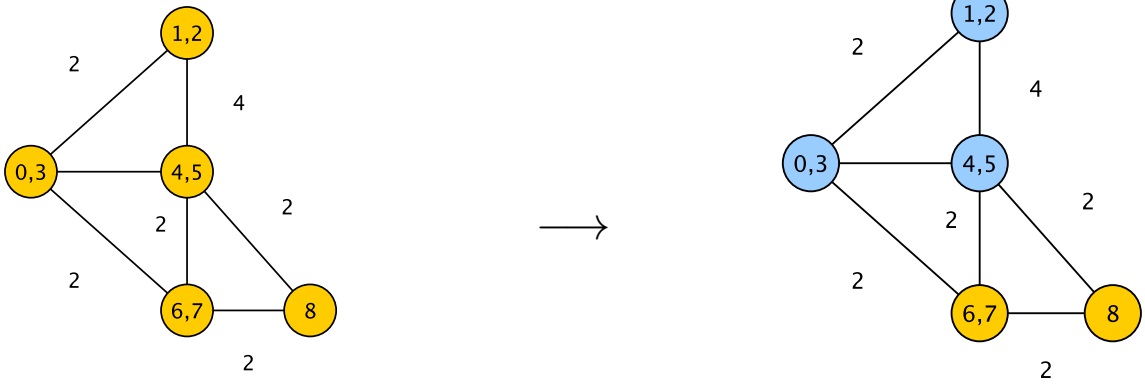
Formal Description

- Given a graph $G_0 = (V_0, E_0)$:
  - Coarsening
    - ★ G_0 is transformed into a sequence of smaller graphs $G_1, G_2, \cdots, G_m$ such that $|V_0| > |V_1| > |V_2| > \cdots > |V_m|$
  - Partitioning
    - ★ A 2-way partition P_m of the graph $G_m = (V_m, E_m)$ is computed that partitions V_m into two parts, each containing half the vertices of G_0
  - Refinement
    - ★ The partition P_m of G_m is projected back to G_0 by going through intermediate partitions $P_{m-1}, P_{m-2}, \cdots, P_1, P_0$

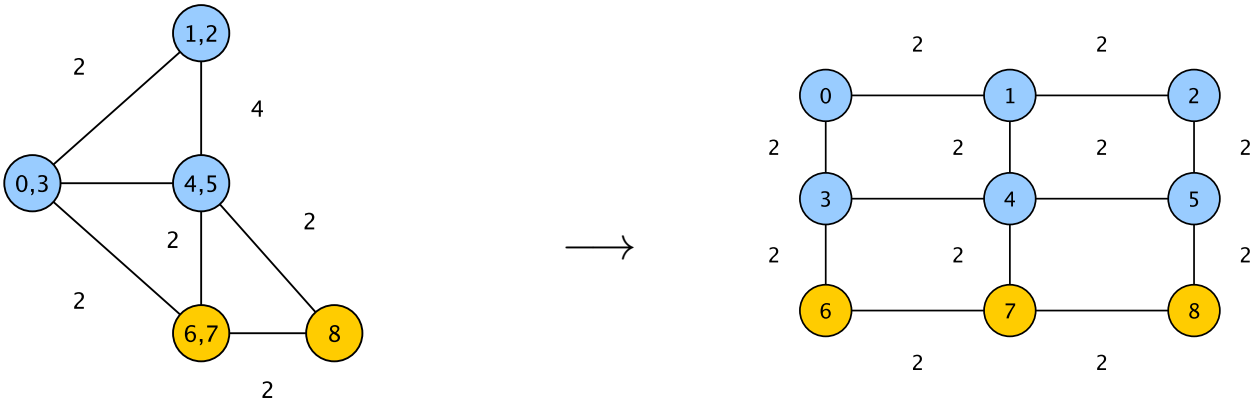
Coarsening



Partitioning



Refinement



Stampede Host

Manufacturer	Intel
Model	Xeon E5-2680
μArch	Sandy Bridge
Clock freq	2.70 GHz
#CPUs (sockets)	2
#Cores/CPU	8
#Thread/Core	1
L1 cache size/core	32 KB
L2 cache size/core	256 KB
L3 shared cache size/CPU	20 MB
Main Memory/CPU	16 GB
Vector width	256 bits (AVX)

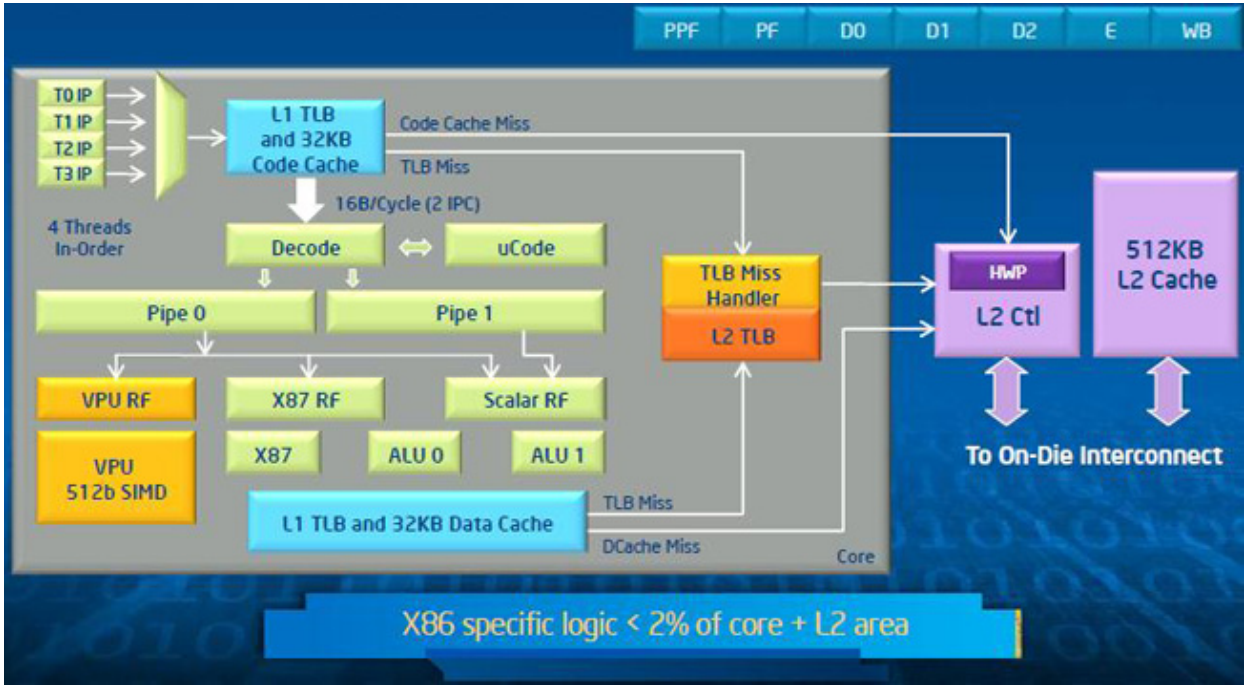
Table 1 : Intel Xeon E5-2680

Stampede Co-processor - Xeon Phi

Manufacturer	Intel
Model	Xeon Phi SE10P
μArch	Many Integrated Cores - MIC
Clock freq	1.1 GHz
#CPUs (sockets)	1
#Cores/CPU	61
#Thread/Core	4
L1 cache size/core	32KB
L2 cache size/core	512 KB
Main Memory/CPU	8 GB
Vector width	512 bits

Table 2 : Intel Xeon Phi

Xeon Phi Coprocessor Core[1]

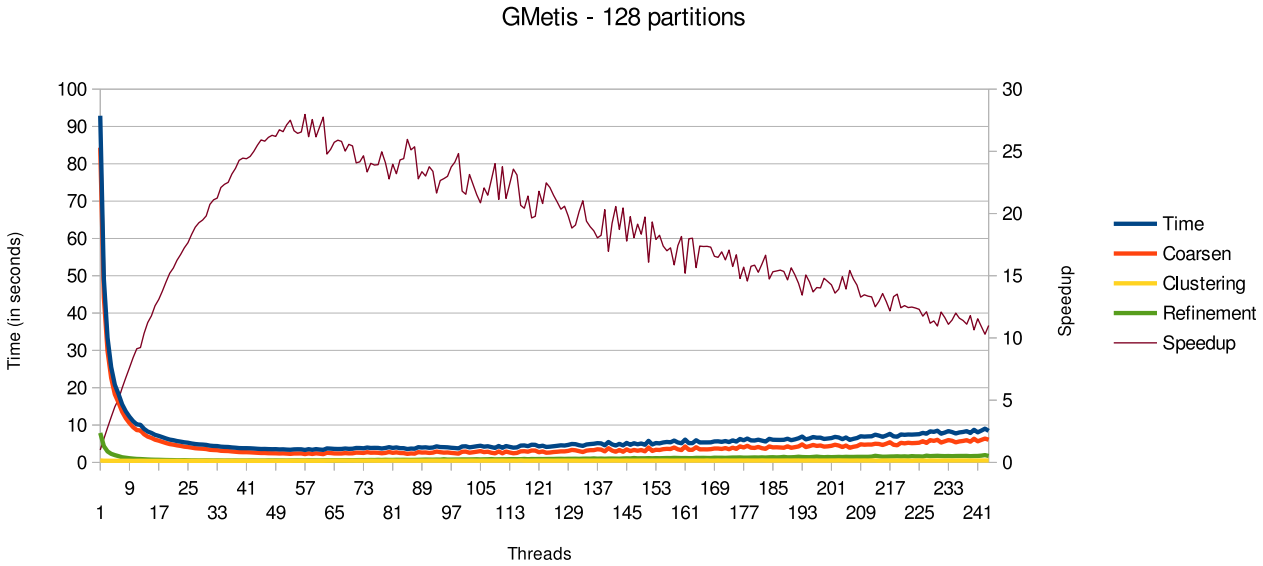


[1] http://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-codename-knights-corner

Important characteristics

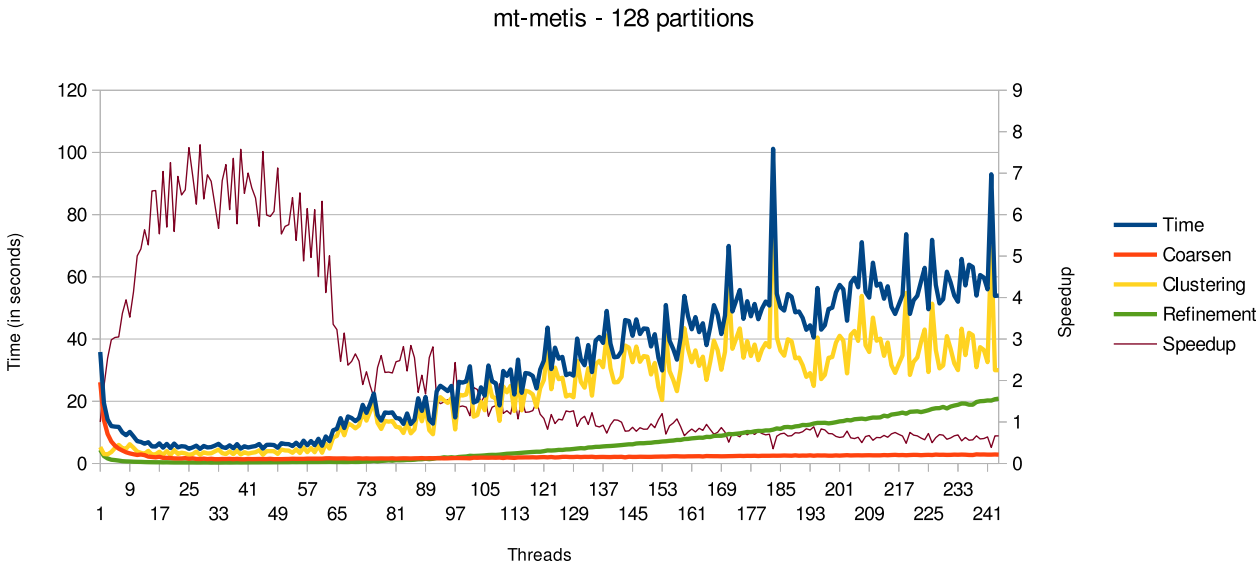
- Four hardware threads per core
- In-order dual issue pipeline
- Pipeline does not issue instructions from the same hardware context for two consecutive clock cycles
- Maximum issue rate only attainable with at least 2 threads per core

GMetis[2]

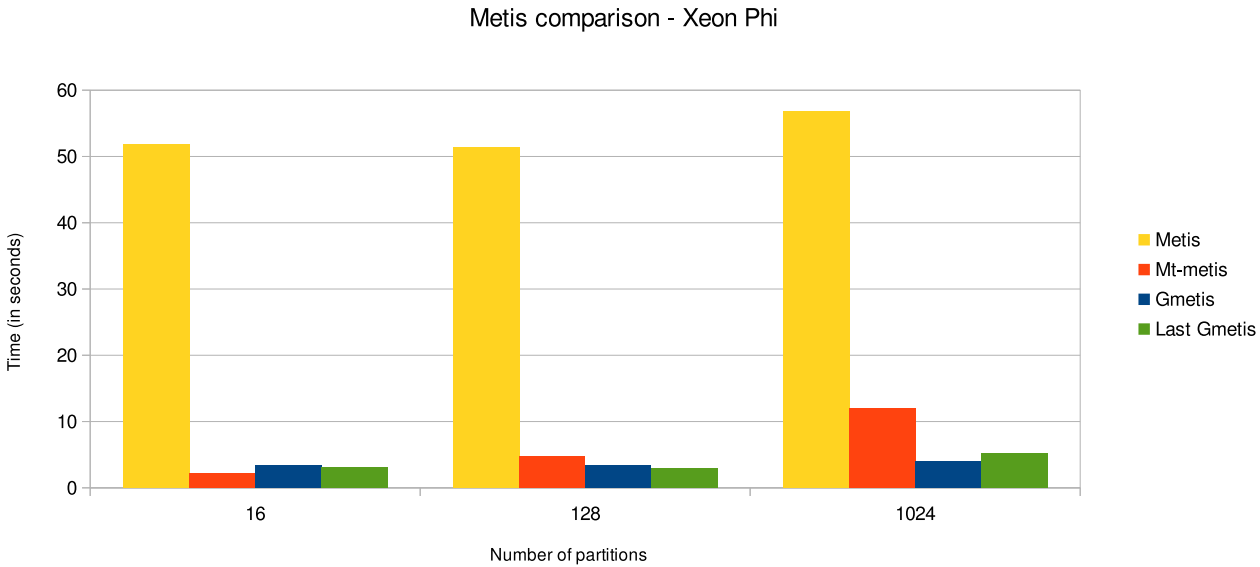


[2]USA-road-d.W.gr with 6262104 nodes and 15248146 edges

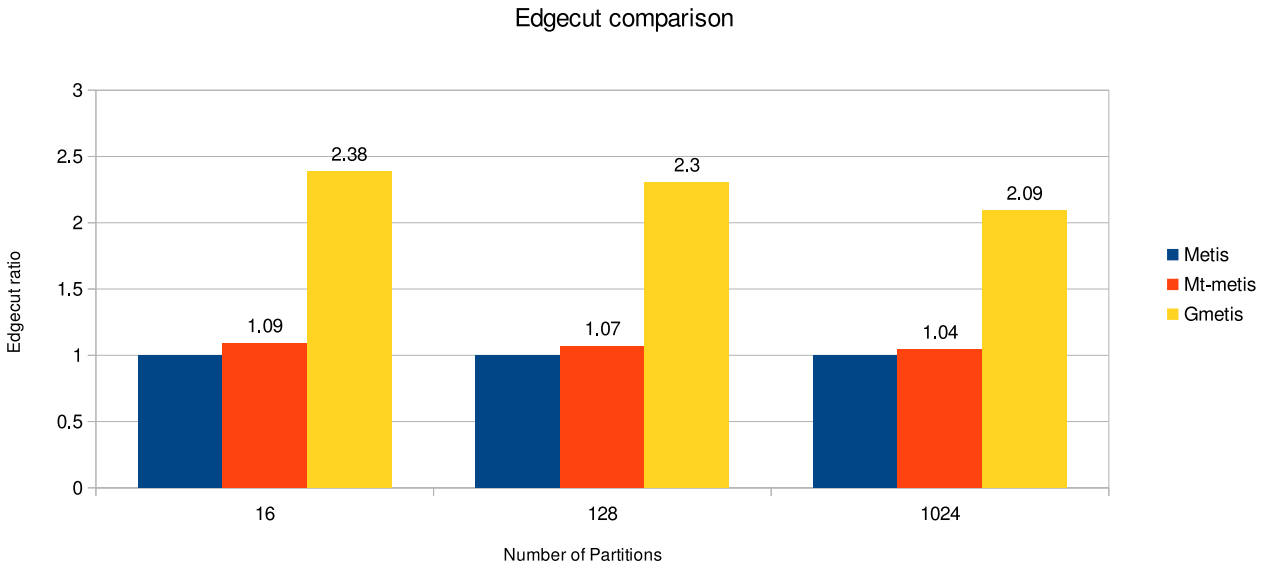
Mt-metis


mt-metis - 128 partitions
Time (in seconds)
Speedup
Threads
120
100
80
60
40
20
0
9
8
7
6
5
4
3
2
1
0
1 9 17 25 33 41 49 57 65 73 81 89 97 105 113 121 129 137 145 153 161 169 177 185 193 201 209 217 225 233 241
Time
Coarsen
Clustering
Refinement
Speedup

GMetis


Metis comparison - Xeon Phi
Time (in seconds)
60
50
40
30
20
10
0
16
128
1024
Number of partitions
Metis
Mt-metis
Gmetis
Last Gmetis

GMetis


Edgecut comparison
Edgecut ratio
3
2.5
2
1.5
1
0.5
0
2.38
2.3
2.09
1.09
1.07
1.04
16
128
1024
Number of Partitions
Metis
Mt-metis
Gmetis

Enhancements

- Package Mapping (HW Topology)
  - Default Mapping
  - Load Balance
  - Dense Package
- Use of Random Match and Heavy Weight Match
- WorkList schedulers
- Software prefetching

Conclusion

- Metis and mt-metis have better edgecut;
- Metis and mt-metis have lower runtimes for a smaller number of partitions;
- GMetis is faster for a high number of partitions;
- Metis graph partitioning algorithm is not suitable to run on MIC as it does not use SIMD extensions;
- Metis and mt-metis are written in C whereas GMetis is written in C++ and uses various high level abstractions (e.g. Templates). This may explain differences in performance;
- Software availability problems made progress difficult;

GMetis - Xeon Phi

David Pereira Rui Brito

August 8, 2013

?