University of Texas at Austin

Institute for Computational Engineering and Sciences (ICES)

GMetis - Xeon Phi

David Pereira
pg22821@alunos.uminho.pt

Rui Brito
pg22781@alunos.uminho.pt

Austin, August 2013

Abstract

1 Introduction

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

2 The Metis Algorithm

Formally, the metis algorithm consists of three phases. They are as follows:

- 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 $P_{m-1}, P_{m-2}, \cdots, P_1, P_0$

Visually, this translates into the following scenario:

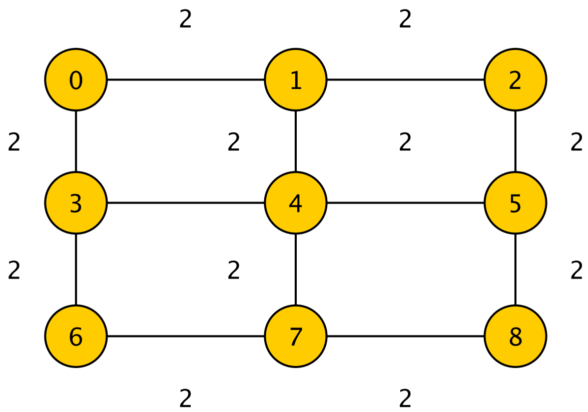

Figure 1: Initial graph

Figures 1 and 2 illustrate the coarsening phase. During this phase, a sequence of coarser graphs is constructed.[2] A coarser graph is constructed by matching neighbour vertices and then contracting the edges.

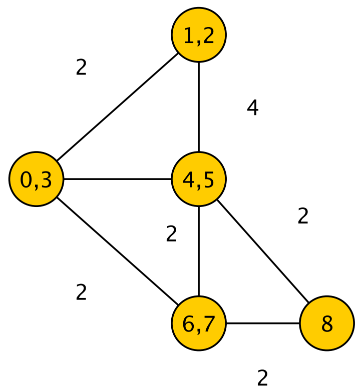

Figure 2: Coarsened graph

Figure 3 displays the partitioned graph, this is the next step in the algorithm. To do this, a Greedy Graph Growing (GGGP) algorithm is used.[3] The goal of this phase, is to compute a high quality bisection (e.g., small edge-cut) of the coarsened graph such that each part contains roughly half of the vertices and edges of the original graph.

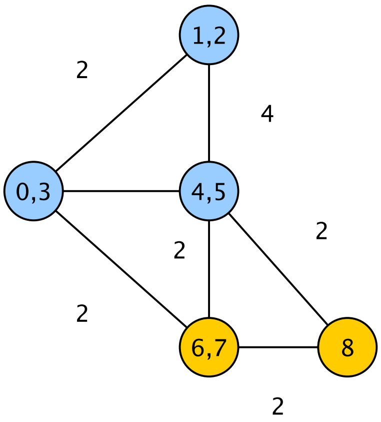

Figure 3: Partitioned graph

Figure 4 shows the results of the refinement phase. During this stage, the partition of the coarser graph is projected back to the original graph by going through the graphs.[3] Once again, the goal here, is to minimize the edge-cut, however, a good balance in the number of vertices assigned to each partition is also very important. Hence, in this final phase, some algorithms use special heuristics to further improve on the balancing achieved.

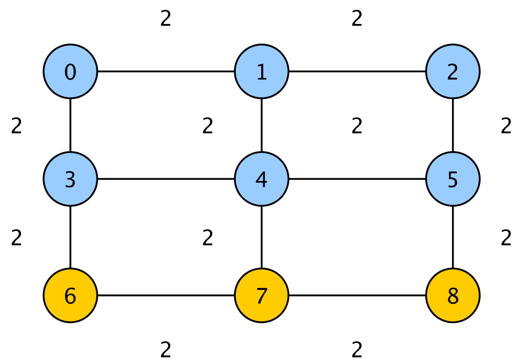

Figure 4: Refined graph

3 System characteristics

The measurements were performed in both Stampede's hosts and coprocessors. The hosts are comprised of dual Intel Xeon E5-2680, while the coprocessors are the new Intel Xeon Phi with 61 cores. Their characteristics are presented in the following tables.

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

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

Apart from the characteristics showed in table **??**, there are others that should be mentioned. Each core contains a in-order dual pipeline which can issue two instructions from the same hardware thread per clock cycle. However, the front-end of the pipeline does not issue instructions from the same hardware thread in consecutive cycles.[1]

rThis means that the maximum issue rate is only attainable with at least 2 threads per core while the other threads have the purpose of hiding pipeline stalls due to memory latency.

The fact that the pipeline issue instructions in-order increases memory related problems.

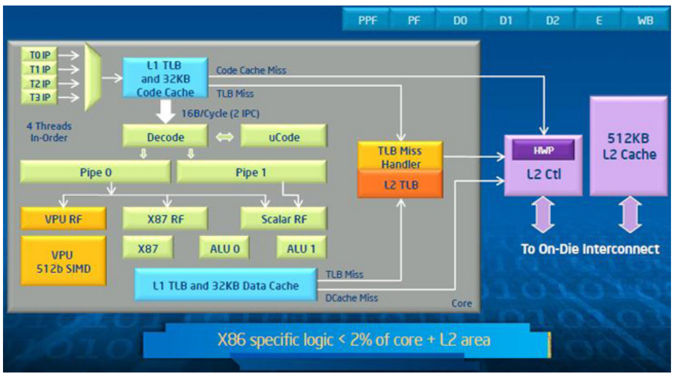

Figure 5: Xeon Phi μArch

4 Metis

5 Mt-metis

6 GMetis and the Galois Framework

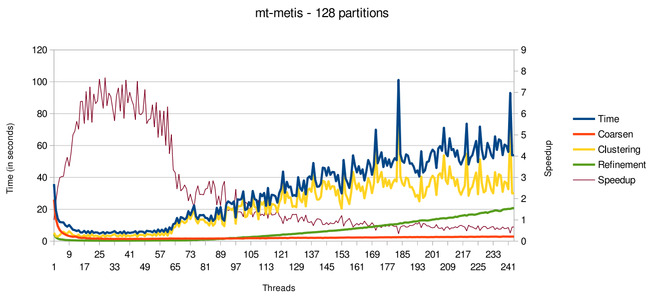

Figure 6: Mt-metis - 128 partitions

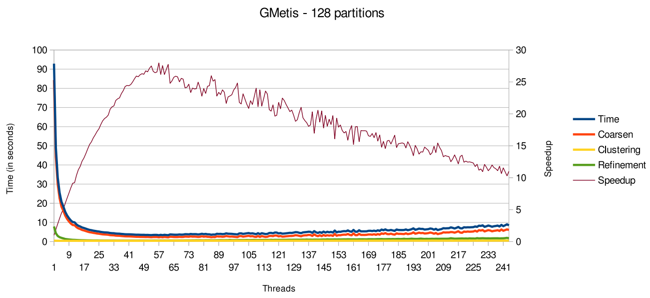

Figure 7: GMetis - 128 partitions

This figure shows the scalability of gmetis on Xeon phi over the runtime with on thread. This example is for 128 partition, but we did measurements for different number of partitions, such as 16 and 1024. All of them have a similar behaviour.

7 Conclusion

Results showed that both *Metis* and *Mt-metis* have better edgecut than *Gmetis*. However, Gmetis's runtime is lower for a high number of partitions.

Xeon Phi provides a theoretical performance of 2112 GFlop/sec for double precision arithmetic and 1056 GFlops/sec for single precision arithmetic. This values comprises the use of 60 cores since one core is necessary to perform operating system operations.

References

[1] Shannon Cepeda. Optimization and performance tuning for intel® xeon phi™ coprocessors, part 2: Understanding and using hardware events. `http://software.intel.com/en-us/articles/optimization-and-performance-tuning-for-intel-xeon-phi-coprocessors-part-2-understanding`, November 2012.

[2] George Karypis and Vipin Kumar. Parallel multilevel graph partitioning. Technical report, University of Minnesota, 1995.

[3] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, December 1998.