UCRL-TR-236625

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Evaluating Zoltan for Static Load Balancing on BlueGene Architectures

G. Kumfert

November 16, 2007

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

FY07 TechBase Final Report

EVALUATING ZOLTAN FOR STATIC LOAD BALANCING ON BLUEGENE ARCHITECTURES

Account: 4347-71

Budget: $31,000

Gary Kumfert, PI
Center for Applied Scientific Computing
Computation Directorate

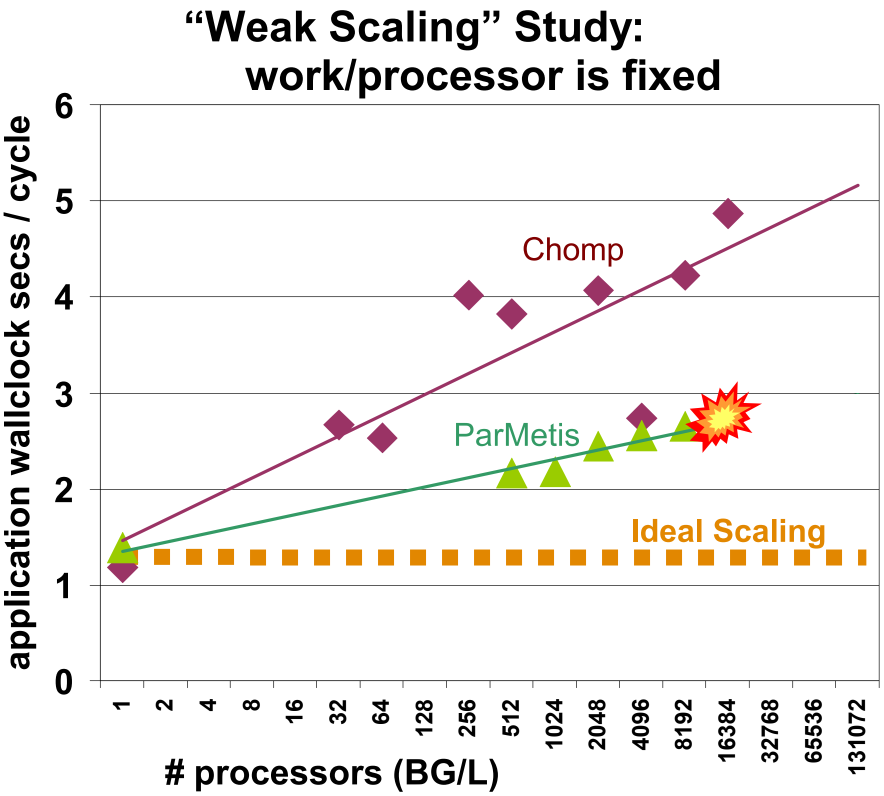
Abstract

The purpose of this TechBase was to evaluate the Zoltan load-balancing library from Sandia National Laboratories as a possible replacement for ParMetis, which had been the load balancer of choice for nearly a decade but does not scale to the full 64,000 processors of BlueGene/L. This evaluation was successful in producing a clear result, but the result was unfortunately negative. Although Zoltan presents a collection load-balancing algorithms, none were able to meet or exceed the combined scalability and quality of ParMetis on representative datasets.

1. BACKGROUND AND MOTIVATION

Assigning work to processors is a fundamental operation to maximize parallelism in scientific simulations. For the broad category of partial differential equation (PDE)-based simulations on complex geometries, heuristic algorithms are used to partition the descretized geometry, or mesh, into processor sized chunks called "domains." This partitioning can occur "statically" at the start of a computation, or "dynamically" where the work assignments are adjusted periodically as the computation progresses. There is a sizeable body of research in partitioning for mesh-based PDE codes and sparse linear solvers. We recommend [DBK06] as a comprehensive survey article of the field.

The motivation for our work is that at BG/L scale, the scaling and quality characteristics of current partitioners are no longer sufficient. The quality of a partition directly affects the amount of communication overhead and ultimately the overall performance of a mesh-based PDE simulation. Scaling has recently become an issue because high quality partitioners, such as ParMetis [KK99],



do not run on more than 16K processors of BlueGene/L (see Fig. 1).

Zoltan [Z07] is a collection of parallel load balancing algorithms supported by ASC at Sandia National Laboratories and includes ongoing algorithmic research [C07] by the discrete math group at Sandia as well their academic collaborators. Zoltan had already been ported to a wide variety of computing architectures, but not BlueGene.

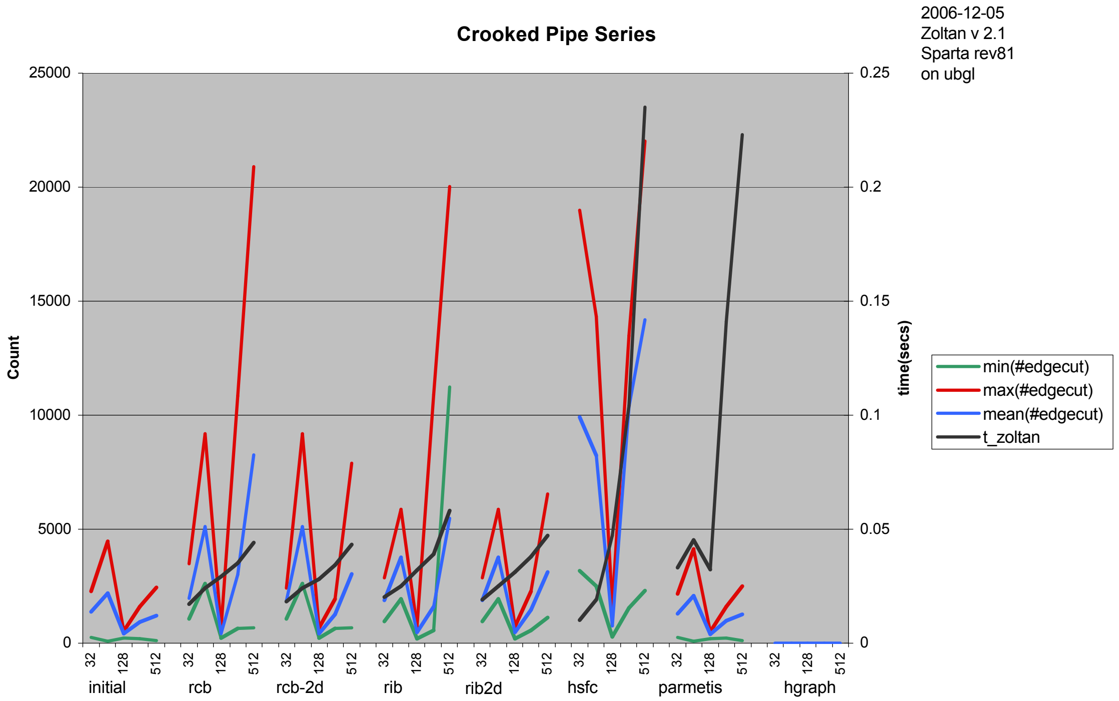
2. Technical Approach

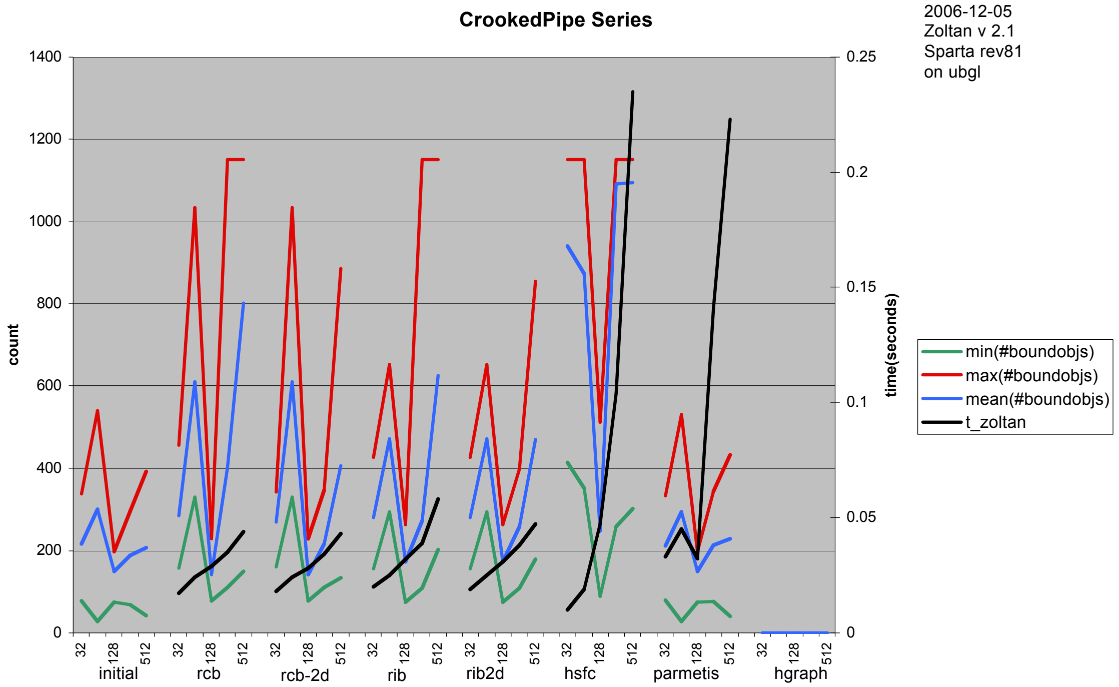
We ported Zoltan to BlueGene and created enough software scaffolding to load up representative problems given to us from programmatic partners. We generally relied on Zoltan's internal capabilities to report generally accepted quality metrics such as balance, number of cut edges, and number of boundary zones, but also produced the new domain assignments (a.k.a. "colormaps") for programmatic partners to independently.

This TechBase also benefited from the enthusiastic participation of Karen Devine from Sandia, Albuquerque who is a researcher and developer of Zoltan. During the course of this TechBase she also visited LLNL, met with programmatic partiners, and gave a talk in CASC.

3. Results

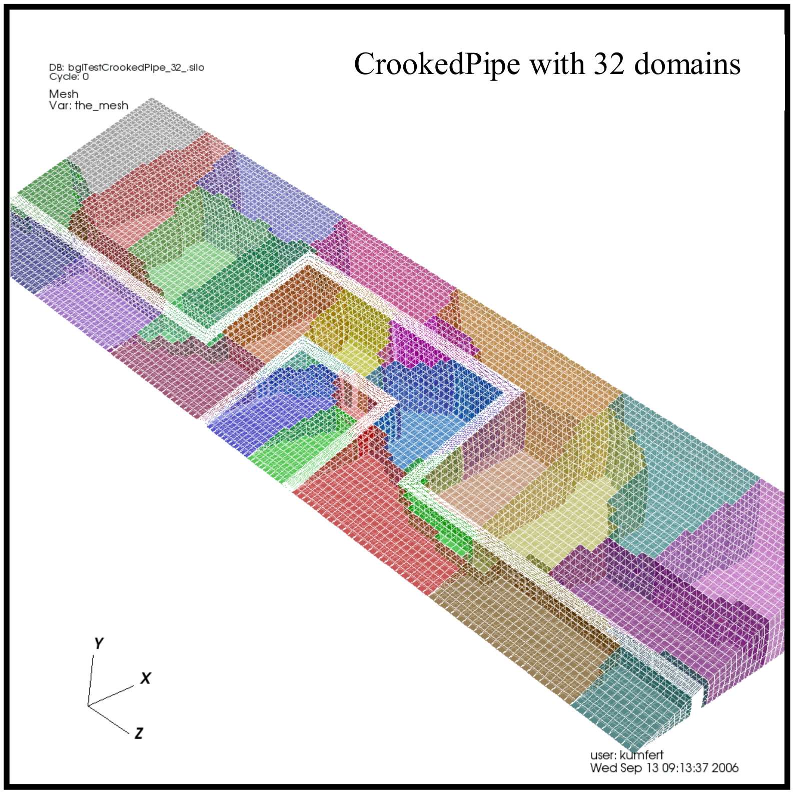
The following results were using the CrookedPipe sample mesh series. This series of meshes were generated for 32, 64, 128, 256, and 512 subdomains at a fixed ratio of 1150 zones per processor.





These tests were run on uBGL using Zoltan version 2.1.

The algorithms we employed in this study were Recursive Coordinate Bisection (RCB), RCB constrained to cuts in 2 of the 3 dimensions (mainly because the sample mesh is generated by extending a 2-D mesh around an axis of symmetry.), Recursive Inertial Bisection (RIB), RIB with the 2-D constraint, Hilbert Space Filling Curve (HSFC), Multilevel Graph Partitioning, and Hypergraph Partitioning. The two



CrookedPipe with 32 domains

quality metrics we consider was number of edges cut and number of boundary objects. Each of these metrics appear in their own graph with the min, max, and mean. The cost in wallclock time is superimposed on both charts and uses the right vertical axis. The quality of the partitioning has a disproportionate effect on the total time to solution on a full calculation, so the times in any of these load balancing steps is largely negligible.

The initial partitioning of the sample tests is very good, and very similar to the multilevel graph partitioning algorithm in Zoltan. Upon further inquiry, it was revealed that the sample meshes were load-balanced by ParMetis before being written to disk and given to us. Furthermore, Zoltan does not implement its own multilevel graph algorithm, but uses ParMetis for that case. So the corrolation turned out to be another validation that we are reading the input files correctly and presenting the appropriate callbacks for Zoltan.

There is no data for the BlueGene because that algorithm did not run to completion successfully on that architecture. This algorithm is based on a newer model than standard graph partitioning and promises to generate high quality load balancing. However, it is also known by its implementers to be even more memory intensive than ParMetis, so we did not pursue this algorithm further.

Another surprise was with the maximum number of boundary zones in a subdomain generated by some of these algorithms. Remembering that there is only 1150 zones in a subdomain, RCB, RIB, and HSFC algorithms all generated at least one domain with all surface zones, no purely internal zones that could be computed independently. This is an extremely undesirable characteristic.

One open question is why the 2-D RIB algorithm is better than the 3-D version. Whereas RCB is locked into cutting along the Cartesian planes, RIB is supposed to be able to cut at whatever angle it chooses. Certainly the input mesh is “almost” 2-D. The RIB algorithm could reasonably be expected to only make cuts perpendicular to that plane on its own accord without extra guidance from the operator.

4. CONCLUSION

Zoltan appears to be a capable implementation and front-end to several load-balancing algorithms. Unfortunately the algorithms simply aren't sufficient for the extreme scale and memory constraints presented by BlueGene/L. Investing in algorithmic research appears to be necessary to effectively balance the computation and communication on these machines at full scale.

REFERENCES

[C07] U. Catalyurek, E. Boman, K. Devine, D. Bozdag, R. Heaphy, L.A. Riesen. Hypergraph-based Dynamic Load Balancing for Adaptive Scientific Computations. Proceedings of IPDPS'07, Best Algorithms Paper Award, March 2007.

[DBK06] Karen Devine, Eric Boman, and George Karypis. Partitioning and Load Balancing for Emerging Parallel Applications and Architectures. Chapter in *Frontiers of Scientific Computing*. Heroux, Raghavan, and Simon, eds. SIAM Press. 2006.

[KK99] Gorge Karypis and Vipin Kumar. Parallel Multilevel k-way Partitioning Scheme for Irregular Graphs. SIAM Review 41:2 (1999) pp 278-300