Extreme Scalability Challenges in Micro-Finite Element Simulations of Human Bone

C. Bekas[*], A. Curioni
IBM Research, Zurich Research Laboratory, Switzerland

P. Arbenz, C. Flaig,
Computer Science Dept., ETH Zurich, Switzerland

G. H. van Lenthe, R. Müller, A. J. Wirth
Institute for Biomedical Engineering, ETH Zurich, Switzerland

February 19, 2008

Abstract

Coupling recent imaging capabilities with microstructural finite element (μFE) analysis offers a powerful tool to determine bone stiffness and strength. It shows high potential to improve individual fracture risk prediction, a tool much needed in the diagnosis and treatment of osteoporosis that is, according to the WHO[1], second only to cardiovascular disease as a leading health care problem. We adapted a multilevel preconditioned conjugate gradient method to solve the very large voxel models that arise in μFE bone structure analysis. The intricate microstructure properties of bone lead to sparse matrices with billions of rows, thus rendering this application to be an ideal candidate for massively parallel architectures such as the BG/L Supercomputer. In this work we present our progress as well as the challenges we were able to identify in our quest to achieve scalability to thousands of BG/L cores.

1 Introduction

High resolution in vivo peripheral quantitative computed tomography (pQCT) provides detailed information on bone structure (see Fig. 1). The underlying voxel model admits to estimate the local bone density. The analysis of bone density (using other, more commonly available technology) is today's

[*] Corresponding author: `bek@zurich.ibm.com`

[1] World Health Organization

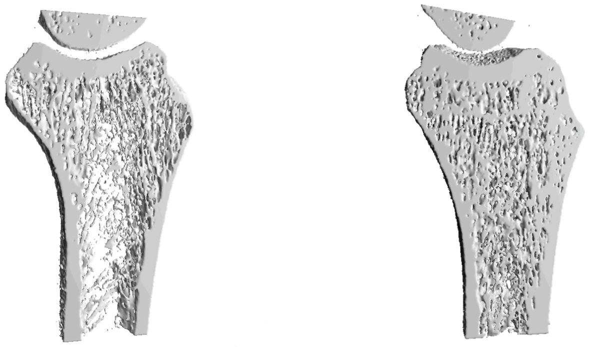

Figure 1: Bone specimens from human subjects. Left: Low density (osteoporotic) specimen from a 72 year old male subject. Right: High density (normal) specimen from a 78 year old male subject.

approach of predicting bone strength and fracture risk in diseases like osteoporosis that is, according to the WHO, second only to cardiovascular disease as a leading health care problem.

Such a quantitative analysis of bone density does not take into account the microarchitectural structure of the bone. Coupling recent imaging capabilities with microstructural finite element (μFE) analysis offers a powerful means to determine bone stiffness and strength. It shows high potential to improve individual fracture risk prediction, a tool much needed in the diagnosis and treatment of osteoporosis. μFE models are created from CT scans by a direct voxel-to-element conversion (see Fig. 2). The intricate microarchitectural structure of bone entails that these μFE models possess a very large number of elements and, by consequence, degrees of freedom.

2 Computational model

Most μFE analyses of bone usually rely on linear elasticity in displacement form. The intricate geometric structure of the bone is approximated by voxels. Each displacement component is a continuous piecewise trilinear function that is determined by its values in the vertices of the voxels. The large number of voxels entails large linear symmetric positive-definite systems of equations that have to be solved. The method of choice is the preconditioned conjugate gradient (PCG) algorithm (see for example [7]).

Recently, we have devised a matrix-free variant of PCG that does not require building the system matrix but still is able to construct an aggregation-based AMG preconditioner [1]. The method is implemented in the software

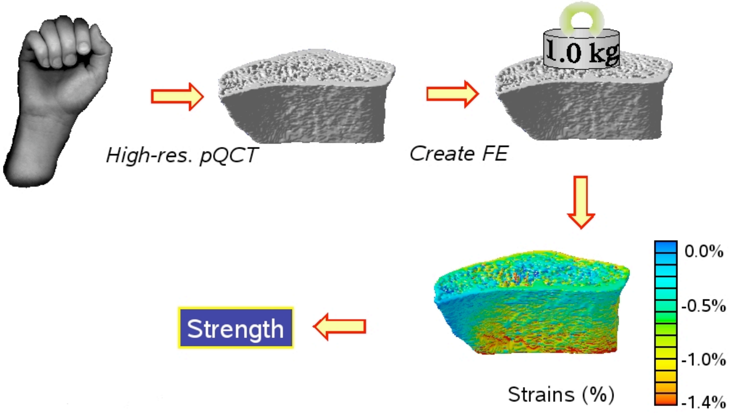

Figure 2: Simulation model.

package ParFE[2] which is parallelized using MPI[3] and is based on the public-domain object-oriented software framework Trilinos [5]. Data is distributed by means of ParMETIS [6] and stored in the HDF5 data format [4]. In [1] we presented results obtained on the Cray XT/3 at the Swiss National Supercomputing Centre, (CSCS) in Manno, Switzerland[4], that validaded the correctness and usefulness of our approach.

In the present work we curry this important research to the next level. In particular, the accurate study of realistic simulations of large human bone specimens requires extreme scale-out that stretches the limits of both algorithms and computational platforms. Here, we report our progress as well as the challenges that we face towards this goal. We focused on the IBM BG/L Supercomputer with the goal of exploiting it's excellent scale-out potential that can allow us to attempt realistic simulations that involve one order of magnitude larger computational load. Indeed, the BG/L Supercomputer, coupled with algorithmic advances we introduced into our methods, led to the simulation of a human vertebra bone specimen that is the largest of it's kind.

3 The study

We conducted a study of artificial as well as human bone specimens resulting in very large sparse systems of up to about 1.5 billions of unknowns. These runs always required less than half an hour, using up to 8 racks (8192 nodes) of the BG/L system at the T.J. Watson Research Center. Pre- and

[2] `http://parfe.sourceforge.net/`
[3] `http://www.mpi-forum.org/`
[4] `http://www.cscs.ch`

cores	repart	precond	solution	total	iters
1	2.50	27.5	113	149	94
8	6.60	45.2	116	179	86
27	7.10	51.5	113	185	80
64	7.10	53.6	124	199	86
125	7.60	55.7	122	202	81
216	8.00	65.6	119	207	79
343	8.60	55.0	119	211	77
512	9.10	67.5	118	214	75
729	10.4	70.5	118	216	74
1000	12.0	87.0	126	248	77
1728	18.5	185	145	376	81

Table 1: Run times in seconds for the weak scalability test. The last column indicates the number of PCG iterations.

postprocessing of data took place at the CSCS.

We have conducted two series of experiments. In the first, we measured the so called weak scalability of the code on a sequence of carefully constructed artificial bone samples that increase in size proportional to the increase of the BG/L cores employed. The second series involves strong scalability (the number of BG/L cores increases while the problem size stays fixed) on a real human bone specimen which (to the best of our knowledge) is the largest simulation of its kind so far, the results of which provide valuable information about the strength of osteoporotic bone. A second important target of this experiment was to identify the parts of the code that exhibit the best as well as the worst scalability. In particular, we focused on measuring the performance of the graph repartitioning, the construction of the preconditioner and the solution phase (application of the preconditioned CG algorithm).

3.1 Weak scalability test

The problems were obtained by mirroring a small cube of human trabecular bone[5], scanned with a high-resolution microCT system (see Fig. 3). We have tested weak scalability using a series of 12 cubes (see Table 1), starting from 1 to 1728 BG/L nodes, where the k-th cube required k^3 BG/L nodes. The smallest cube ($k=1$) involved about 300.000 degrees of freedom, the largest cube ($k=12$) about 500 million. All runs were done in co-processor mode in which the second PPC440 core of the BG/L nodes handles MPI communication. We observed satisfactory scalability up to 1000 CPUs. However, load

[5] The interior of the bone, see Fig. 1.

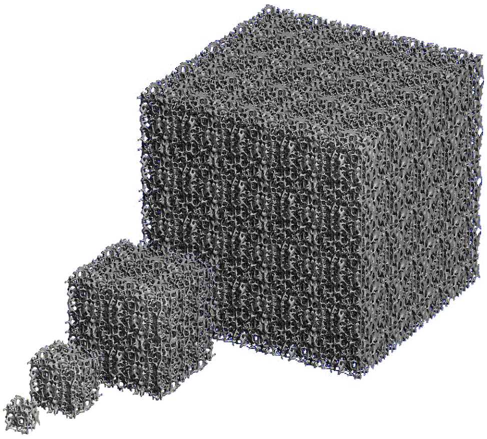

Figure 3: Sequence of artificial bone samples for the weak scalability test.

cores	repart	precond	solution	total	iters
4800	157	614	482	1341	71
5000	187	420	447	1143	68
5400	230	577	431	1323	67
5800	304	362	417	1165	69
6200	397	430	425	1332	69
6800	497	368	416	1359	70
7900	749	427	380	1632	71
8100	775	418	365	1635	70

Table 2: Run times in seconds for the strong scalability test. The last column indicates the number of PCG iterations.

imbalance caused by poor repartitioning started to manifest in the largest cases entailing the increase of the time for constructing the preconditioner.

3.2 Strong scalability test

We conducted the largest simulation of its kind so far (1.5 billions degrees of freedom) and calculated the effective strain of a vertebral bone specimen. This enabled a highly detailed analysis of bone deformation under load, and calculation of bone stiffness and strength. Fig. 4 illustrates the effective strain of the bone specimen. For this specimen we conducted a strong scalability test (see the run times in Table 2). No less than 4800 BG/L nodes were required for the memory requirements of the problem to be satisfied (ca. 2.4 TBytes). We scaled our runs up to eight (8) BG/L racks (8192 BG/L dual core nodes).

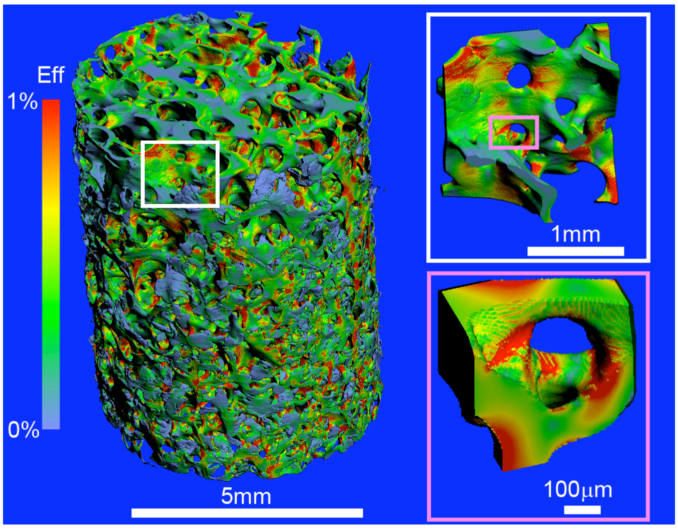

Figure 4: Effective strain on a human bone (vertebra) specimen.

The solution phase scales quite well and uniformly (column 'solution') as the number of employed BG/L cores increases. The construction of the preconditioner exhibits similar overall scalabilty (column 'precond'). However, this is not as uniform as before. The reason is the load imbalance that is caused by the graph repartitioning (ParMETIS). Fig. 5 illustrates the box-plots for the finite element mesh-node distribution to the available processing units (BG/L cores in our case). Box-plots graphically show the lower, median and upper quartile values of the node distribution (contained in the box). More importantly the crosses indicate outlier data, i.e., nodes per core values that are far different from the bulk of the data. It it clear that we are experiencing significant load imbalance! Observe that only a few cores are assigned many fewer mesh nodes than the average, while several cores are assigned a significantly larger node load than the average. This causes cores with a light load to wait at synchronization points and thus to register much larger communication times than the average. The cases for that mesh-node distribution disparity is the smallest are those that achieve the best scalability in the preconditioner construction phase (i.e., the case of 6800 BG/L cores).

Returning to the scalability of the solution phase we note that the preconditioned conjugate gradient method is dominated by the sparse matrix-vector products (including the application of the preconditioner) and the global reductions (`MPI_ALLREDUCE`) for the calculation of vector inner products. For this purpose we utilize the global tree network available on the BG/L platform that achieves excellent latency as well as bandwidth which is important (remember that global reduction involves more that 10GBytes

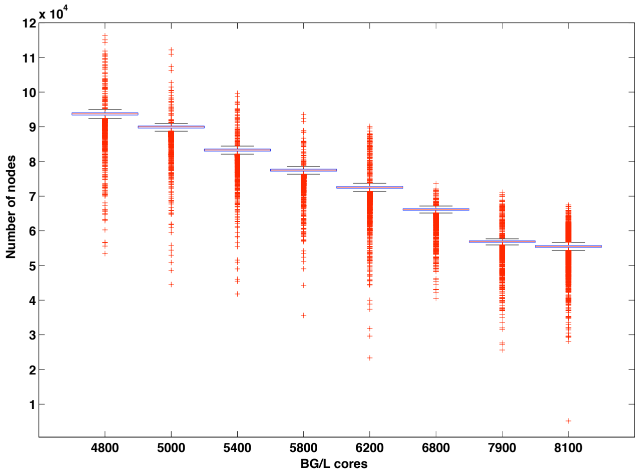

Figure 5: Box-plots of the mesh-node distribution achieved by graph repartitioning (ParMETIS) for the strong scalability test. According the MATLAB help: "The boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the boxes to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers".

of data since we have more than 1.5 billion degrees of freedom).

3.3 Repartitioning scalability

We observed that the major bottleneck in extreme scale-out of ParFE is in graph repartitioning. The top viewgraph in Fig. 6 illustrates a comparison of the run times for the construction of the preconditioner, the solution phase, and the time required for graph repartitioning. It is clear that the latter dominates the overall runtimes.

In order for ParMETIS to be suitable to run on 8 racks of a BG/L system (with 512 MBytes of memory per node) we introduced a number of algorithmic modifications. The main change involved the geometric partitioning algorithm which relies on a serial version of the quicksort algorithm. Geometric partitioning is used to achieve an initial partitioning of CT scan data to the available cores on which k-way multilevel repartitioning is applied at a second step. We implemented a fully parallel mergesort algorithm to replace the serial quicksort, since using the latter on 8192 BG/L nodes (8 racks) requires 512 MBbytes of memory per node leaving no memory for any other calculations (or even for the light weight Linux OS kernel on the compute cores), thus causing the program to halt. A second important modification was to replace asynchronous all-to-all communication with MPI collective

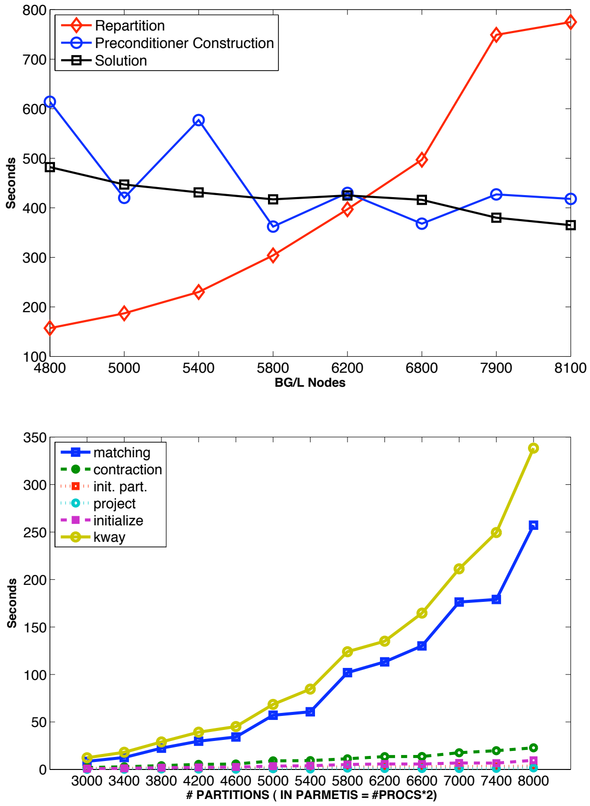

Figure 6: Top: Strong scalability test on the large bone specimen. Timings for the repartition (ParMETIS), construction of preconditioner, and solution phase. Bottom: Timing analysis for repartitioning on a smaller bone sample.

communication. The former floods the network with messages, quickly overflowing the MPI buffers, while the latter is very efficiently implemented on the BG/L communication layer.

In achieving scale out on thousands of compute cores the challenges we face are aligned towards two main directions.

Load imbalance. Complicated geometry of the application domain, such as the intricate structure of trabecular bone (see Figs. 1 and 4) can entail significantly imbalanced partitions that have a strong negative impact when thousands of processors are used (see Fig. 5). It is well known that graph partitioning is an extremely challenging combinatorial problem for which the best algorithms utilize complicated heuristics. Without doubt during the last years multilevel coarsening-decoarsening algorithms (such as the ones implemented in popular software such as ParMETIS, Zoltan[6], or SCOTCH[7] [3]) have dominated the scene. However, our experiments indicate that in the era of massively parallel machines with tens (or even hundreds) of thousands of computing cores this model of graph partitioning has reached its limits.

Scalability. The scalability of parallel graph partitioning tools on tens of thousands of processors appears to be a formidable task. The bottom viewgraph of Fig. 6 illustrates a run time breakdown for ParMETIS on a smaller real human bone sample. There are two main observations.

- There are two phases that clearly do not scale at all. These are the coarsening and decoarsening (partition refinement) phases. They involve the solution of a series of graph maximal-matching and matching refinement problems for which many point to point communications are needed. In ParMETIS, these are implemented in an asynchronous manner (in order to avoid dead-locks and race conditions among other targets) that pose great difficulties when tens of thousands of computing cores are involved. Finding the optimal scheduling for these communications, so that they can be safely performed concurrently, is a formidable task on massively parallel platforms.
- Observe that the run times of the phases that appear to cause much less scalability problems, in comparison with the above, increase with a slope that suggests they will also not scale when several tens of thousands of cores will be used. Thus, we claim that achieving extreme scale-out on real world applications with intricate geometry characteristics, we will have to use either a different algorithmic approach, one that naturally scales in massively parallel machines, or we will need to consider a different mapping of applications to available resources, one that does not rely on graph partitioning.

4 Conclusion

We have presented our progress in very large scale simulations of human bones that can provide crucial information and insight for appreciating the risk of fractures due to osteoporosis. It is well known that osteoporosis is a leading health care problem, thus the development of a software tool that can render this kind of risk analysis to be a routine procedure is very important. In this work we have shown that it is possible to achieve simulations of unprecedented size in a few minutes, thanks to the synergy of powerful algorithms and the excellent scale-out potential of the BG/L Supercomputer.

[6] `http://http://www.cs.sandia.gov/Zoltan/`

[7] `http://www.labri.fr/perso/pelegrin/scotch/`

Our work has led us to understand a number of important challenges, which we believe will become even more demanding in the next few years. In particular, in anticipation of the petaflop machines, efficient mapping of real world applications on millions of processing elements will require next generation algorithms and efficient mapping models. We believe that the models of mapping such as graph partitioning or even the most advanced ones such as hypergraph partitioning (see for example [2]), will need to be extended or modified so that the particular underlying architecture is seriously taken into consideration. Furthermore, these mapping tools will need to be naturally suited for extreme scale-out as well, otherwise they will become the bottleneck of the whole process.

Acknowledgment

We acknowledge the support of the Data Analysis & Visualisation group at the Swiss National Supercomputing Centre (CSCS).

References

[1] P. Arbenz, G. H. van Lenthe, U. Mennel, R. Müller, and M. Sala, A Scalable Multi-level Preconditioner for Matrix-Free μ-Finite Element Analysis of Human Bone Structures. Internat. J. Numer. Methods Eng. 73 (7): 927–947 (2008)

[2] Ü.V. Çatalyurek and C. Aykanat, A Hypergraph-Partitioning Approach for Coarse-Grain Decomposition. Proceedings of the 2001 ACM/IEEE conference on Supercomputing, p. 28. doi:10.1145/582034.582062

[3] C. Chevalier and F. Pellegrini. PT-Scotch: A tool for efficient parallel graph ordering. Parallel Computing. Article in press. doi:10.1016/j.parco.2007.12.001.

[4] HDF5: Hierarchical Data Format. Reference Manual and User's Guide are available from http://hdf.ncsa.uiuc.edu/HDF5/doc/.

[5] M.A. Heroux, et al., An overview of the Trilinos project. ACM Trans. Math. Softw. 31(3), 397–423 (2005).

[6] G. Karypis and V. Kumar, A Parallel Algorithm for Multilevel Graph Partitioning and Sparse Matrix Ordering. J. Parallel Distr. Comput. 48 (1): 71–85 (1998).

[7] Y. Saad, Iterative Methods for Sparse Linear Systems, Second Edition SIAM, 2003