John von Neumann Institute for Computing

Scalable Systems for Computational Biology

Ch. Pospiech

published in

From Computational Biophysics to Systems Biology (CBSB08),
Proceedings of the NIC Workshop 2008,
Ulrich H. E. Hansmann, Jan H. Meinke, Sandipan Mohanty,
Walter Nadler, Olav Zimmermann (Editors),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **40**, ISBN 978-3-9810843-6-8, pp. 31-36, 2008.

http://www.fz-juelich.de/nic-series/volume40

Scalable Systems for Computational Biology

Christoph Pospiech

IBM Deutschland GmbH, Freiberger Straße 35, 01067 Dresden, Germany
E-mail: pospiech@de.ibm.com

Driving up processor clock speed is about to hit several limitations, a soaring energy consumption being among them. The BlueGene system® from IBM is shown to provide performance in a scalable and energy efficient way. Results from NAMD and similar codes show, how the system can be used in the area of Computational Biology.

1 The Promise of Moore's Law

The frequently quoted term "Moore's Law" goes back to a paper published in the Electronics Magazine[7]. In this paper, G. E. Moore states that

> "the complexity for minimum component costs has increased at a rate of roughly a factor of two per year ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer."

The "factor of two per year" was later relaxed into a "doubling every two years". This statement became known as "Moore's Law". It was quoted many times, not always in full length or precise form. So it started shifting its meaning. As a consequence, the wikipedia[11] entry for Moore's Law lists no less than 9 different formulations on exponential growth laws related to integrated circuits.

Hans Werner Meuer in a recent paper[6] uses linear regression on a logarithmic scale to fit data from the Top500 list to an exponential growth law. This is illustrated in Fig. 1. He finds a similar growth rate as "predicted" by Gordon Moore. It is therefore tempting to see this as another variation of Moore's law.

These data might generate the impression that computers grow with exponential rate in about every aspect. But there are already some limitations visible on the horizon. The original quotation of Moore's law was a statement about the number of components on an integrated circuit. Cramming more components to the same space forces the components to get smaller in size. In the current CMOS technology the gates sizes are now approaching molecular levels[9,8], and start leaking like a sieve. To make up for the lost electrons, the energy density has to be increased. In conjunction with an increased clock speed this leads to the heat flux curves as shown in Fig. 2[9]. The heat flux is directly connected to the energy consumption as energy is needed to generate the heat and also to cool it again.

It might be tempting to anticipate a new curve to the right of the existing two curves in Fig. 2, but there is yet no technology that allows jumping again to a cooler regime. One way to evade from the heat and energy trap is to trade low energy for the single core

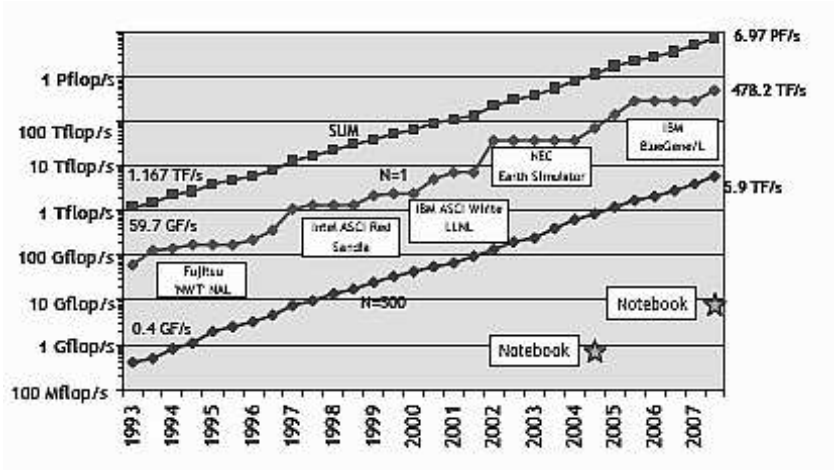

Figure 1. Top500 performance data fitted with linear regression on a logarithmic scale to an exponential growth law.

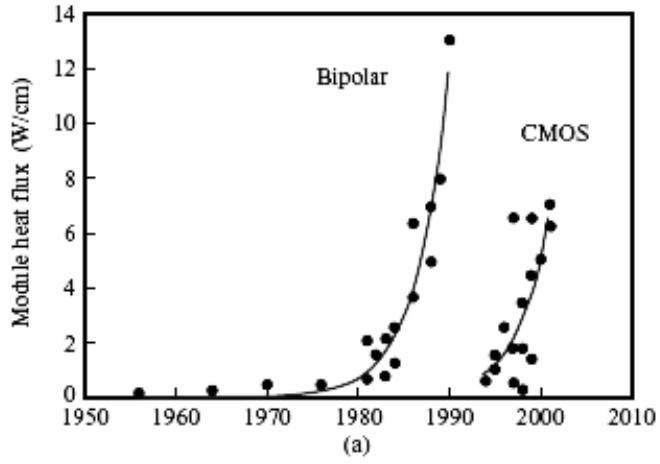

Figure 2. Module heat flux from various systems in bipolar and CMOS technology.

against high clock rate. To obtain high application performance out of low clocked cores, many cores have to be thrown at the problem. The IBM Blue Gene® is one example for a system that follows this concept. The "green500" list[10] from February 2008 shows that this concept may lead to systems with both, high performance and high energy efficiency.

2 Blue Gene

On June 26, 2007, IBM announced the Blue Gene/P™ system as the leading offering in its massively parallel Blue Gene® supercomputer line, succeeding the Blue Gene/L™ system. The following description of the Blue Gene/P™ system is based on a recent paper by the IBM Blue Gene® team[2].

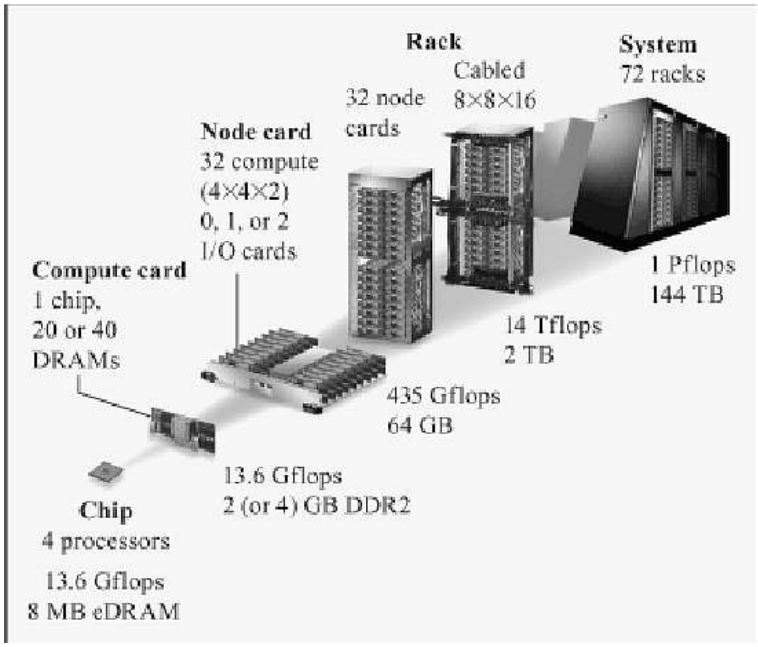

Figure 3. The packaging hierarchy of a Blue Gene/P™ system.

The packaging of the Blue Gene/P™ system is shown in Fig. 3. The main building block is a single ASIC (application-specific integrated circuit), with four IBM PowerPC 450 (PPC450)-embedded 32-bit processor cores, arranged as an SMP. A dual-pipeline floating-point unit (FPU) is attached to each PPC450 core. The design of this dual FPU is logically identical to the one used in the Blue Gene/L™ system. It supports two simultaneous double-precision floating-point calculations in SIMD (single-instruction, multiple-data) fashion, along with instruction set extensions for complex number arithmetic. The dual-pipeline FPUs can simultaneously execute two fused multiplyadd instructions per machine cycle, each of which is counted as 2 FLOPs (floating-point operations). Thus, each processor unit (PPC450 and FPU) has a peak performance of 4 FLOPs per machine cycle.

In addition to the compute nodes, the Blue Gene/P™ system contains a configurable number of I/O nodes. The I/O nodes are physically the same compute cards as described above, but their position in the system differentiates their logical function. I/O nodes have the 10-Gigabit Ethernet (GbE) interface enabled for communication with a file system and host computers.

Thirty-two compute cards and, optionally, up to two I/O cards are packaged onto the next-level board, called the node card. Sixteen node cards are plugged from both sides into a vertical midplane card, completing an assembly of 512 compute nodes in an $8 \times 8 \times 8$ configuration. The inbound and outbound network connections for this 512-way cube are routed to four link cards that carry a total of 24 Blue Gene/P™ link (BPL) chips. The assembly of 16 node cards, 4 link cards, and an additional service card is called a midplane or a 512-way. The BPL chips are relatively simple switches that, depending on the size and configuration of a user partition of the system, route network signals back into the midplane (completing the wraparounds for an $8 \times 8 \times 8$ torus) or route the network signals through cables to another midplane for larger partitions. Two midplanes, one on top of the other, complete a rack. Thus, a rack has 1,024 nodes, or 4,096 cores, giving a peak performance of 13.9 teraflops (Tflops). Scaling upward, a 72-rack system can package 72K nodes (288K cores) (where K stands for 1,024) into a 1-petaflops (Pflops) (peak) system, and larger configurations up to 256 racks (3.56 Pflops peak) are possible.

In the Blue Gene/P™ system, three networks are used for node-to-node communication: a 3D torus network, a collective network, and a global barrier network. On the Blue Gene/L™ system, the processor cores were responsible for injecting (or receiving) packets to (or from) the network. On the Blue Gene/P™ system, a direct memory access (DMA) engine has been added to offload most of this responsibility from the cores, thus enabling better overlap of communication and computation. Specifically, the DMA interfaces with the torus network. The combination of the DMA, torus network, and memory system is capable of supporting high-bandwidth communications.

The Blue Gene® system software approach is to start with a minimal-functionality system stack, but one that is designed to scale. The Blue Gene® strategy to achieve scalability and high performance has been to start simply, for example, with space sharing, one job per partition, no paging, and one thread per core.

3 Application Examples

3.1 CPMD (Car-Parrinello Molecular Dynamics)

The CPMD code is a plane wave/pseudopotential implementation of Density Functional Theory, particularly designed for ab-initio molecular dynamics. Its first version was developed by Jurg Hutter at IBM Zurich Research Laboratory starting from the original Car-Parrinello codes[1]. CPMD runs on many different computer architectures and it is well parallelized. The application is mainly written in Fortran, parallelized for distributed-memory with MPI, with an option to add an additional level of parallelism using OpenMP for multi-processor nodes. CPMD makes extensive use of three-dimensional FFTs, which require efficient all-to-all communication[3]. The scalability was improved using a task-group implementation of the FFT with a special mapping to the Blue Gene® torus network[4]. Moreover, overlap matrices, which were replicated in the standard CPMD code, have been distributed on a subset of the nodes to be able to handle large systems (more than 3000 electronic states). The single processor performance of CPMD was optimized for Blue Gene® using SIMD-enabled routines for the most common calls such as DGEMM, DCOPY, AZZERO, and FFT.

The following comparison was taken from benchmarks with a methanol vapor/liquid

OpenMP speedup	
2 OpenMP threads	4 OpenMP threads
1.91	3.6

Table 1. OpenMP speedup on Blue Gene/P^{TM} for CPMD.

interface ($\sim$700 atoms , 70Ry). Table 1 shows that the Blue Gene/P^{TM} takes significant advantage of an MPI/OpenMP hybrid programming paradigm.

3.2 NAMD (NAnoscale Molecular Dynamics)

NAMD (nanoscale molecular dynamics) is a production molecular dynamics (MD) application for biomolecular simulations that include assemblages of proteins, cell membranes, and water molecules. In a biomolecular simulation, the problem size is fixed and a large number of iterations must be executed in order to understand interesting biological phenomena. Hence, we need MD applications to scale to thousands of processors, even though the individual timestep on one processor is quite small. NAMD has demonstrated its performance on several parallel computer architectures.

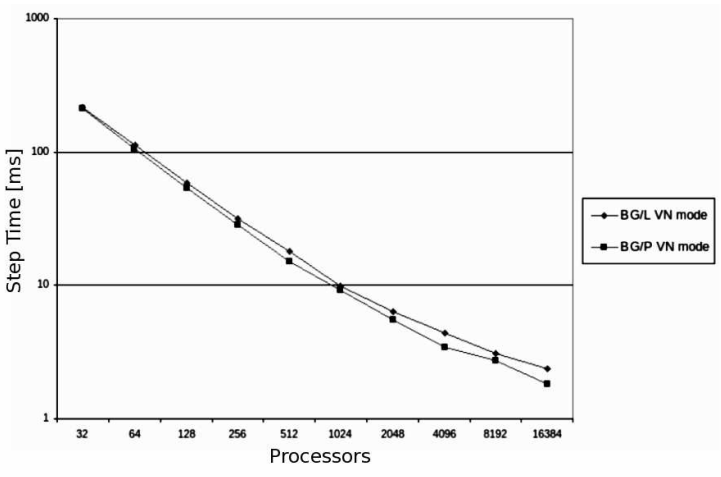

Figure 4. Comparison of NAMD execution times on Blue Gene/L^{TM} and Blue Gene/P^{TM}.

Figure Fig. 4 shows the comparison of times per execution step for Blue Gene/L^{TM} and Blue Gene/P^{TM}. The application was a 92K atom APoA1 benchmark with PME every 4 steps. The data for Blue Gene/L^{TM} were taken from a recent paper by S. Kumar et al.[5]. The Blue Gene/P^{TM} results are very recent and conveyed to the author via private communication.

The comparison shows that Blue Gene/P^{TM} performs better for all compared number of processors. In opinion of S. Kumar, the clock-speed only accounts for a 5% speed up. The DMA accounts for the rest of the difference.

4 Summary

Moore's law might generate the impression that computers grow with exponential rate in about every aspect, but there are already some limitations visible on the horizon. One way to evade from the heat and energy trap is to trade low energy for the single core against high clock rate. To obtain high application performance out of low clocked cores, many cores have to be thrown at the problem. The IBM Blue Gene® is one example for a system that follows this concept. The BlueGene system® from IBM is shown to provide performance in a scalable and energy efficient way. Results from NAMD and similar codes show, how the system can be used in the area of Computational Biology.

References

1. R. Car and M. Parrinello, *Unified Approach For Molecular Dynamics and Density Functional Theory*, Phys. Rev. Lett. **55(22)**, 2471, 1985.
2. IBM Blue Gene team, *Overview of the IBM Blue Gene/P project*, IBM Journal on Research and Development **52 no. 1/2**, 199–220, 2008.
3. J. Hutter and A. Curioni, *Dual-level parallelism for ab initio molecular dynamics: Reaching teraflop performance with the CPMD code*, Parallel Computing **31**, 1–17, 2005.
4. J. Hutter and A. Curioni, *Car-Parrinello Molecular Dynamics on Massively Parallel Computers*, ChemPhysChem **6**, 1788–1793, 2005.
5. S. Kumar et. al., *Scalable molecular dynamics with NAMD on the IBM Blue Gene/L system*, IBM Journal on Research and Development **52 no. 1/2**, 145–158, 2008.
6. H. W. Meuer, *The TOP500 Project: Looking back over 15 years of Supercomputing Experience*, to be published in INFORMATIK SPEKTRUM, Springer Verlag, `http://www.top500.org/files/TOP500_Looking_back_HWM.pdf`.
7. G. E. Moore, *Cramming more components onto integrated circuits*, Electronics Magazine **38**, no. 8, 1965, `ftp://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf`.
8. E. J. Nowak, *Maintaining the benefits of CMOS scaling when scaling bogs down* IBM Journal on Research and Development **46 no. 2/3**, 169–180, 2002.
9. R. R. Schmidt, B. D. Notohardjono, *High-end server low-temperature cooling*, IBM Journal on Research and Development **46 no. 6**, 739–751, 2002.
10. `http://www.green500.org/lists/2008/02/green500.php`.
11. `http://en.wikipedia.org/wiki/Moore's_law`.