Sustained Simulation Performance 2012

Michael M. Resch • Xin Wang • Wolfgang Bez
Erich Focht • Hiroaki Kobayashi
Editors

Sustained Simulation Performance 2012

Proceedings of the joint Workshop on High Performance Computing on Vector Systems, Stuttgart (HLRS), and Workshop on Sustained Simulation Performance, Tohoku University, 2012

Springer

Editors
Michael Resch
Xin Wang
High Performance Computing Center
Stuttgart (HLRS)
University of Stuttgart
Stuttgart
Germany

Wolfgang Bez
NEC High Performance Computing
Europe GmbH
Düsseldorf
Germany

Erich Focht
NEC High Performance Computing
Europe GmbH
Stuttgart
Germany

Hiroaki Kobayashi
Cyberscience Center
Tohoku University
Sendai
Japan

Front cover figure: Flow simulation of a F1 model by Building Cube Method. Illustration by Cyberscience Center, Tohoku University, 6-3 Aramaki-aza-aoba, Aoba, Sendai 980-8578, Japan.

ISBN 978-3-642-32453-6 ISBN 978-3-642-32454-3 (eBook)
DOI 10.1007/978-3-642-32454-3
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012948969

Mathematics Subject Classification (2010): 68Wxx, 68W10, 68Mxx, 68U20, 76-XX, 86A10, 70FXX, 92Cxx

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Sustained simulation performance is a widely ignored issue in high-performance computing. Typically the focus is on peak performance and on Linpack results. Rarely we do see a discussion about real-world applications and their performance on large-scale systems.

This book presents the results of the 14th Teraflop Workshop which was hosted by the High Performance Computing Center Stuttgart/Höchstleistungsrechenzentrum Stuttgart (HLRS) in December 2011 as well as the contributions of the 15th workshop in this series. In order to adapt the title of the workshop series to the changing technology in high-performance computing the workshop was renamed to "Workshop on Sustained Simulation Performance." It was held in March 2012 at the Tohoku University in Sendai, Japan.

This book contains contributions focused on the issue of sustainable performance on large-scale systems. This includes issues of exaflop computing as one important part. The three other parts are focusing on real-world applications and their performance on state-of-the-art HPC architectures. These contributions together give an overview of the level of performance that is available for the scientific community today.

The workshop series is based on a project that was initiated in 2004. The Teraflop Workbench Project was founded initially as a collaboration between the High Performance Computing Center Stuttgart (HLRS) and NEC Deutschland GmbH (NEC HPCE) to support users to achieve their research goals using high performance computing.

Since then a series of workshops have put their focus on sustainable performance. These workshops have become a meeting platform for scientists, application developers, international experts, and hardware designers to discuss the current state and future directions of supercomputing with the aim of achieving the highest sustained application performance.

Work in the Teraflop Workbench project gives us insight into the applications and requirements for current and future HPC systems. We observe the emergence of multi-scale and multi-physics applications, the increase in interdisciplinary tasks, and the growing tendency to use today's stand-alone application codes as modules

in prospective, more complex coupled simulations. At the same time, we notice the current lack of support for those applications. Our goal is to offer an environment that allows users to concentrate on their area of expertise without spending too much time on computer science itself.

The first stage of the Teraflop Workbench project (2004–2008) concentrated on user's applications and their optimization for the 72-node NEC SX-8 installation at HLRS. During this stage, numerous individual codes, developed and maintained by researchers or commercial organizations, have been analyzed and optimized. Several of the codes have shown the ability to outreach the TFlop/s threshold of sustained performance. This created the possibility for new science and a deeper understanding of the underlying physics.

The second stage of the Teraflop Workbench project (2008–2012) focuses on current and future trends of hardware and software developments. We observe a strong tendency towards heterogeneous environments at the hardware level. At the same time, applications become increasingly heterogeneous by including multi-physics or multi-scale effects. The goal of the current studies of the Teraflop Workbench is to gain insight into the developments of both components. The overall target is to help scientists to run their applications in the most efficient and most convenient way on the hardware best suited for their purposes.

We would like to thank all the contributors of this book and the Teraflop Workbench project. We thank especially Prof. Hiroaki Kobayashi for the close collaboration over the past years and are looking forward to intensifying our cooperation in the future.

Stuttgart, June 2012

Michael Resch
Xin Wang
Uwe Küster

Contents

Part I
Exascale Computing: New Challenges in Software and Hardware

Beyond Exaflop Computing: Reaching the Frontiers of High Performance Computing

Michael M. Resch

Abstract High Performance Computing (HPC) has over the last years benefitted from a continuous increase in speed of processors and systems. Over time we have reached Megaflops, Gigaflops, Teraflops, and finally in 2010 Petaflops. The next step in the ongoing race for speed is the Exaflop. In the US and in Japan plans are made for systems that are supposed to reach one Exaflop. The timing is not yet clear but estimates are that sometime between 2018 and 2020 such a system might be available. While we debate how and when to install an Exaflop system discussions have started about what we have to expect beyond Exaflops. There is a growing group of people who have a pessimistic view on High Performance Computing assuming that the continuous development might come to an end. However, we should have a more pragmatic view. Facing a change in hardware development should not be seen as an excuse to ignore the potential for improvement in software.

1 Introduction

In 1893 Frederick Jackson Turner wrote an essay on the significance of the frontier in American history [1]. Referring to a bulletin of the Superintendent of the Census for 1890 he found that the impressive move westwards of the US-American settlers had come to an end. In his description of the advance of the frontier Turner identifies five barriers that were reached over time: the Alleghenies, the Mississippi, the Missouri, the Rocky Mountains, and finally the Pacific Ocean. With the advent of the settlers at the Pacific Ocean, Turner argues, the development of the USA turned mostly inwards and focused on the development of the settled country.

M.M. Resch (✉)
University of Stuttgart, Höchstleistungsrechenzentrum Stuttgart (HLRS), Nobelstrasse 19, 70569 Stuttgart, Germany
e-mail: resch@hlrs.de

M. Resch et al. (eds.), *Sustained Simulation Performance 2012*,
DOI 10.1007/978-3-642-32454-3_1,

In Supercomputing we have seen a similar breath taking advance. Over only a few decades we have reached Megaflops, Gigaflops, Teraflops, and Petaflops and are approaching Exaflops. While many prepare for the usage of such Exascale systems, others start to doubt whether we will be able to reach Exaflops or go beyond that barrier. From a technical point of view there is no doubt that Exaflops are possible. The driving factor, however, is no longer innovation but rather a massive usage of standard parts and components. Quantitative growth has replaced qualitative improvement.

In this paper we have a look at how we got to the situation in which we are in High Performance Computing today. We will then investigate the ongoing trends and extrapolate future developments. Instead of arguing for new efforts to build faster systems we will emphasize that software is the key to solutions in simulation. Hence, we will argue that the improvement of software is much more important than further heroic efforts in designing ever faster hardware.

2 History

High Performance Computing has seen a long history of progress over the last six decades. For a long time Moore's assumption about the doubling of transistors on a die [2] and the corollary of the doubling of speed every 18 months turned out to be right. When the increase in clock frequency started to level out, parallelism became the driving factor. Parallelism came so far in two waves. The first one started in the late 1980s and by the early 1990s created a number of interesting concepts. However, most of the advanced concepts—with sometimes very large numbers of processors—failed, and a number of companies developing such systems disappeared from the High Performance Computing arena pretty fast.

2.1 Architecture in Review

High Performance Computing grew out of the primeval soup of computing. Historical accounts [3] claim that Seymour Cray at a certain point in time decided he wanted to build the fastest systems in the world, rather than economically interesting ones. Technically Seymour Cray was following some basic principles which we still have to consider today. For example, the round shape of his first systems was owed to an attempt to reduce distances. The further success of his first company proofed that speed and economic success were possible at the same time. His failure with follow-on projects showed that things did change after a while. As long as the computer was a special purpose instrument for a limited number of users High Performance Computing was based on special purpose systems with special prices.

The advent of the personal computer started to change things. The computer became a ubiquitous instrument. Budgets for computer hardware were gradually

moving from special purpose systems to general purpose hardware. Already in 1990 Eugene Brooks coined the term "Killermicros" [4], describing the end of what was considered to be "dinosaur" systems, and their replacement by microprocessor based systems. It did not take too long until Brooks was proven right. The number of systems using specialized processor technology started to dwindle away and by the year 2000 only small "game reserves" were left. By the year 2009 such specialized processors were virtually extinct when NEC announced its withdrawal from the Japanese Next Generation Supercomputing Project. However, recent announcements of NEC suggest that we may see another round of specialized processors in the future. The driving factor behind such specialized processors is the need of applications.

It is interesting to see that the "killermicros" did not only make an end to specialized processors in High Performance Computing. They also started to cannibalize each other. Over a period of about 10 years a number of processor architectures disappeared from the TOP 500 list [5]. While in the year 2000 the TOP 500 presented five different processor architectures with a substantial share of systems, that same list of the year 2012 shows that about 90 % of the systems are based on the x86-architecture. The main non-x86 architecture is the IBM Power architecture which is used in various IBM BlueGene [6] systems.

The trend in microprocessor architectures was accompanied by a trend in system architectures. In 1994 Donald Becker and Thomas Sterling started what they called the "Beowulf Project" [7]. As a result of this project—that was building a High Performance Computing cluster from standard components—clusters became extremely popular in High Performance Computing. In 2000 expectations were high that future High Performance Computing systems would all be clusters based on "Components Off The Shelf" (COTS). To some extent this has become true. Over 80 % of the systems listed in the TOP 500 list in June 2012 actually are clusters.

The situation as described already shows that a small number of trends shape the landscape of High Performance Computing. After this short historical review we now have a look at the current situation and try to estimate the future developments in hardware.

3 Situation

The result of the first wave of parallelism was a number of systems that typically provided a moderate number of processors in a single system. For a while the largest systems were hovering around 1,000 and up to 10,000 processors or cores. The fastest system in the world in 1993—as presented by the TOP 500 list [5]—was based on 1024 processors. The fastest system in 2003 was based on 5120 processors. The increase was a factor of about five. We have to consider though that in 2003 the number one system—the Japanese Earth Simulator from NEC—was using special fast vector processors, which allowed it to provide a relatively high peak performance with a relatively low number of processors. But even if we look at

the top ten systems of the list, we only find an increase in the level of parallelism of about ten—from 340 processors per system in 1993 to 4180 processors per system in 2003. The parallelism of this kind was not easy to master but message passing was a good approach, and software programmers could easily keep track of their thousands of processes.

Since about 2003 we have seen a second phase of parallelism. This was partially driven by the IBM BlueGene project [6] which was using a larger number of slower processors. Over the last 2–3 years graphics processing units (GPUs)—with their hundreds of cores on one card—have further enhanced the trend. The currently valid TOP 500 list of June 2012 shows a system with more than 1.500.000 cores at the top. This is a factor of 300 over the last 9 years. Looking at the top ten systems we see a factor of 100 over the last 9 years—from 4180 cores per system in 2003 to 418.947 cores per system in 2012. So, while we had 10 years to adapt to an increase in number of processors of ten from 1993 to 2003 we now had 9 years to adapt to a factor of 100 from 2003 till 2012. From a programmer's point of view things are getting out of control.

3.1 Processors

The basis for the top systems in High Performance Computing are currently many-core processors. The number one system in 2011—the Japanese K-Computer—relied on the Fujitsu SPARC64 VIIIfx, a many-core processor with 8 cores [8]. More than 88.000 of these processors are bundled. Other large scale systems are based on the AMD Opteron 6200 processor with 16 cores. In both cases the clock frequency is comparably low. It is 2 GHz for the SPARC processor and 2.3 GHz for the AMD processor.

Another standard building block used in very large scale systems is the so called general purpose graphics processing unit (GPGPU). Based on e.g. NVIDIA 2050 cards [9] a high level of peak performance as well as of Linpack performance is made possible. The NVIDIA 2050 comes with 448 cores which again increases the number of cores for the user.

The background of this increase in number of cores is clear. The International Technology Roadmap for Semiconductors [10] indicates that what was suggested by G.E. Moore more than 50 years ago is still valid. The feature size is shrinking and it will keep doing so for a number of years to come. While we cannot increase the clock frequency anymore—basically because of the high leakage that comes with high clock frequencies—we still can substantially increase the number of transistors on a single chip. As a consequence, we are increasing the number of cores on a chip instead of shrinking the chip and increasing the frequency.

The SPARC VIIIfx and the NVIDIA 2050 find themselves on two ends of a spectrum defined in terms of complexity of cores and number of cores. The SPARC VIIIfx processor comes with a rather complex core design. Each of the cores could

be described as "fat and fast". The typical graphic cards like the NVIDIA 2050 come with a very large number of cores. These cores can be described as "slim and slow".

Finally, we should mention some details about a special purpose system from IBM. The IBM BlueGene [6] is an architecture that is currently based on the IBM Power PC A2 processor. It comes with a relatively low clock frequency of 1.6 GHz and has 18 cores of which 16 are used for computing. The processor's architecture is interesting in that it provides one extra core for running an operating system and one extra core as a spare core in case one of the 16 computing cores fails. These two strategies—operating system offloading and hardware support for fault tolerance—are increasingly becoming important. Unfortunately the Power PC A2 is only available in the BlueGene system for High Performance Computing. Its market share will hence be relatively small. It remains to be seen how this architecture will further evolve. Technically the processor can be considered to be closer to the "fat and fast" solution than to the "slim and slow" solution.

3.2 Networks

In the field of internal communication networks we have seen a variety of solutions in the past [11]. For a while several solutions were competing in the field of cluster computing. With the advent of Infiniband [12] the situation has changed. The new technology has practically replaced all other special solutions in the cluster market. Of the 50 fastest systems in the TOP 500 23 are clusters based on Infiniband. The interesting finding is that 26 of the TOP 50 system are using some kind of proprietary network. Only one system is still based on Gigabit Ethernet (ranked number 42 in November 2011).

When we turn the pages of the TOP 500 list we find that starting around a ranking of 150 the number of Gigabit Ethernet installations substantially increases. This indicates that for high-end systems Infiniband is a must. This is also supported when looking at the level of sustained Linpack performance in the list. The typical Gigabit Ethernet system achieves about 50–60 % of peak performance for the Linpack benchmark. For an Infiniband system this ratio is typically in the range between 45 % and 85 %. The big variation indicates that Infiniband is used to build a large variety of different network topologies.

3.3 Architectures

Looking at architectures we find that clusters dominate the TOP 500 list. About 80 % of the fastest systems in the world are in that group. The rest of about 20 % is based on an MPP architecture approach. Even though clusters are the biggest group we look at the MPP architectures. They seem to be outdated but keep a constant share of about 20 % over the last 9 years, while other types of architectures have

disappeared. What is more interesting: in terms of performance MPPs have a much larger share of the TOP 500 list—in the range of 40 %. This is because MPPs can typically be found in the upper part of the TOP 500. So, when talking about real High Performance Computers we find that MPP and clusters are two competing technologies at equal footing.

One of the reasons for a renaissance of the MPPs is the IBM BlueGene architecture. Originally the concept was based on a relatively light-weight processor. The new Blue Gene/Q has a relatively strong processor but comes with a lower clock frequency than comparable systems. The network is proprietary and provides a 5D-Torus.

In general, one of the main features of MPPs systems seems to be the better network connectivity. The basic performance numbers (latency and bandwidth) for MPI are typically comparable to what Infiniband can offer. However, the better network connectivity should increase the level of sustained performance. Analysing the fastest 50 systems in the world in November 2011, we see that proprietary interconnects on average achieve 78 % of sustained performance for the Linpack benchmark. Infiniband based systems achieve about 74 %. This is not a big difference. One may wonder whether this is the reason why Cray decided to give up its proprietary network development in 2012 [13].

A further investigation of the list shows that low sustained performance is caused by the usage of graphics cards. Such cards provide a high level of peak performance but typically do not work well in terms of sustained Linpack performance. The average sustained Linpack performance of the top 50 systems is 76 %. The average of the six systems that make use of NVIDIA cards is 51 % only. This is a clear indication that such systems do not show satisfactory sustained performance for classical High Performance Computing applications.

What is further interesting is the evaluation of network architectures when we eliminate the NVIDIA results. Without these systems both Infiniband based systems and proprietary systems show an average of 80 % of Linpack performance. The maximum for proprietary networks is 93 %, for Infiniband it is 89 %. The minimum for proprietary networks is 72 %, and for Infiniband it is 59 %. However, the relatively low minimum for Infiniband is the exception to the rule.

4 Potential Paths Forward

The last 20 years have shown that changes in technology in High Performance Computing happen all the time and that predictions are difficult to make. However, there are a number of findings.

The number of cores in a High Performance Computer will further increase. There is currently no way of avoiding a situation in which millions of cores form the compute backbone of a High Performance Computer system. We may see solutions where the large number of cores is hidden from the user. However, this is certainly going to happen based on some kind of software solution. Most likely we are going

to see compilers that support a high level of parallelism in a single node—whatever the term "node" is going to mean in the coming years.

Given the actual lack of advantage for proprietary networks one has to expect that Infiniband—or a follow-on technology—is going to gain more ground also in the TOP 500 list. Economic considerations might lead to an end of proprietary network development in much the same way that they have caused a substantial reduction in processor architectures available for High Performance Computing.

4.1 Exaflops

The analysis of technologies for High Performance Computing shows that an Exaflop system can be built but will be extremely difficult to handle. Furthermore the costs for operating such a system will be relatively high. Recent estimates expect a power consumption in the range of 20 up to 75 MW. Even though this may technically be possible it may not make sense financially. A discussion has hence started about the feasibility of Exaflop computing. While some argue for a change of our programming style [14] for Exaflop computing others discuss whether we will ever see such systems [15].

5 Discussion

An investigation of the current technology available for High Performance Computing systems reveals a number of findings that may be helpful for a future strategy in HPC. First and foremost we can safely assume that we will see a further shrinking of feature-size for semiconductors. As a consequence there is still room for improvements of processors. The same cannot be said for clock frequencies. We will have to live with clock rates of a low single digit number of GHz for the coming years. As a result massive parallelism is the only option that we have to increase speed. Assuming that the trend of the last 10 years will continue we may expect to see systems with a billion cores in 8–10 years from now.

Such a number may be prohibitively large since it may lead to power consumptions beyond the financial reach of typical HPC centers. On the other hand our existing programming models will be challenged by such architectures to an extent that may lead to severe problems for programmers in HPC. MPI was not designed for such large numbers of processes. Whether the concept can be adapted has to be seen. OpenMP has proven to work well for a small number of shared memory processes—in the order of 8–16. However, it cannot be considered to be the method of choice for shared memory systems with thousands of cores. So, although we may get a growth in speed of systems, we have to accept that the time is over when supercomputers regularly provided such an increase in speed at a relatively low cost.