UNIVERSITY OF SÃO PAULO
Institute of Mathematics and Statistics
Parallel and Distributed Programming

Monograph

MAC5742 - Introduction to Parallel and Distributed Programming

1st semester of 2017

The Gordon Bell Prize

Candy Tenorio Gonzales - USP 10427641
cvtenoriog@gmail.com
Décio Lauro Soares - USP 8790723
deciolauro@gmail.com
Fatemeh Mosaiyebzadeh - USP 10120239
mahshid.msy179@gmail.com

Advisors:
Prof. Alfredo Goldman, PhD.
gold@ime.usp.br
T.A. Pedro Bruel
pedro.bruel@gmail.com

Departament of Computer Science
IME - USP
July 3, 2017

Abstract

Development of parallel software has traditionally been thought of as time and effort intensive. This can be largely attributed to the inherent complexity of specifying and coordinating concurrent tasks, a lack of portable algorithms, standardized environments, and software development toolkits.

Meanwhile, one may argue that in the natural world, many complex, interrelated events are happening at the same time, yet within a temporal sequence.

Also, compared to serial computing, parallel computing is much better suited for modeling, simulating and understanding complex, real world phenomena.

During the past decades, the trends indicated by ever faster networks, distributed systems, and multi-processor computer architectures (even at the desktop level) points to show that parallelism is the future of computing.

Nowadays, we are racing to achieve Exascale Computing, but in the late 80's, the early days of the "parallel and distributed" paradigm, the Gordon Bell Prize was created with the to promote the development of parallel processing.

The main goal of this work is to present a broad introduction of this award as also try to analyse its impact on the development of the field from its origins (in 1987) up to today, including a brief overview of selected award winners along the years.

Introduction

This work is part of the evaluation process for the disciple MAC5742 (Introduction to Parallel and Distributed Programming) given by Prof. Alfredo Goldman, PhD. associate professor of the Institute of Mathematics and Statistics at the University of São Paulo during the first semester of 2017.

It consists of the analysis and presentation of a chosen topic related with the contents of the discipline (in our case the **Gordon Bell Prize**) and will be presented in 6 chapters.

The first chapter (**Motivation** 1) will bring a general overview for our motivation in choosing this topic.

After that, the chapter (**Gordon Bell** 2) will describe a brief biography for the prize creator, including his accomplishments, prizes and outstanding works.

In the chapter **ACM Gordon Bell Prize** 3 we will discuss in general lines the ideals behind this award, including its conception and prize criteria.

Following, the chapter **Prize Details** 4 will dive into details about the awards given so far, including the categories, a list of winners, the winners distribution around the globe and so on.

Complementing the details presented in the previous chapters, the chapter **Field Development** 5 will present an analysis for the development of the High Performance Computing field (HPC) in correlation with the beginning of the award.

The above chapters are the main core of our work.

After that, we will present a brief overview for three article award winners in the last years.

This overview will be presented in the chapter **Articles award analysis** 6 that will include a explanation for the criteria used in the article selection and will be divided in three sections.

In the first section (**2016 Article Award** 6.1), we will briefly discuss the most recent award winner. Section **2009 Article Award** 6.2 will also present a short analysis for the 2009 winner.

Finally, section **2010 Article Award** 6.3 will bring a more detailed analysis for the 2010 award winner.

To conclude, we also present a bibliography with a list of all the cited (or relevantly consulted) material used during the confection of this work.

Contents

List of Figures

List of Tables

Motivation

As we previously mentioned, this work is part of the evaluation process for the graduate program in the University of São Paulo and we had to choose among a list of topics that were related with the contents of the disciple to present our work.

In our initial discussions, we decided that we would like to deepen our knowledge in the High Performance Computing (HPC) subarea and, among the topics, one that caught our attention was the ACM Gordon Bell Prize.

Firstly because it was an award conceived with the intention of promoting the development of parallel processing and HPC areas, as also it was hosted by a two renowned publications, the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE).

Secondly, by choosing this topic, we could not only get a better understanding about the evolution of HPC along the years, as also try to infer about the correlation between the award creation and the development in this field.

Finally, by choosing the ACM Gordon Bell Prize topic, we could also get a better understanding about the state-of-the-art research in the High Performance Computing subarea.

Gordon Bell

Chester Gordon Bell 2.1 (born August 19, 1934) is an American electrical engineer and manager that received his B.S. (1956) and M.S. (1957) in electrical engineering from MIT.

He is considered by some as the "father of the minicomputer" and have accomplished and received many honors along his career, like:

- Fellow of the American Academy of Arts and Sciences (1994)
- American Association for the Advancement of Science (1983)
- Association for Computing Machinery (1994)
- IEEE (1974)
- Member of the National Academy of Engineering (1977)
- National Academy of Science (2007)
- Fellow of the Australian Academy of Technological Sciences and Engineering (2009)
- Member of the advisory board of TTI/Vanguard

Also, among the awards he received are some outstanding achievements like:

- Was the first recipient of the IEEE John von Neumann Medal (1992),
- AeA Inventor Award
- Vladimir Karapetoff Outstanding Technical Achievement Award of Eta Kappa Nu
- National Medal of Technology (1991)

Figure 2.1: Gordon Bell (Image provided by the Queensland University of Technology)

2.1 Bell's law of computer classes

In 1972 it proposed what was called as *Bell's law of computer classes.*
This law states that:

Definition 2.1.1 (Bell's law of computer classes)**.** Technology advances in semiconductors, storage, interfaces and networks enable a new computer class (platform) to form about every decade to serve a new need.

Like the *Moore's law*, the *Bell's law* was very accurate in the first decades of its proposal, but started to fail in represent the fast changes that happened in the last decades. Table 2.1 shows how the law developed in the last decades.

Table 2.1: Bell's law of computer classes

Year	Class
1960	Mainframes
1970	Minicomputers
1980	Networked workstations and personal computers
1990	Browser-web-server structure
1995	Palm computing
2000	Web services
2003	Convergence of cell phones and computers
2004	Wireless Sensor Networks

2.2 Publications

Despite the article where the Bell's law was proposed and few other, Gordon Bell also wrote two technical books:

- Computer Structures: Readings and Examples (1971)
- Computer Engineering (1978)

However, even with the articles and works mentioned before, Gordon Bell is really know for the creation of the *ACM Gordon Bell Prize.*

ACM Gordon Bell Prize

The *ACM Gordon Bell Prize* is an award presented by the Association for Computing Machinery (ACM) each year during the Supercomputing Conference (SC).

It was conceived in 1987 by Chester Gordon Bell which main purpose was to track the progress over time of parallel computing, by acknowledging and rewarding innovation in applying high-performance computing to applications in science, engineering, and large-scale data analytics.

From its creation (in 1987) up to the last award given (by the time of this writing the 2016 award) the prize rewarded more than 13 article categories (that will be fully explained in the following chapter 4) and gave a total of 59 awards,with 15 prizes in the last decade.

Despite recognition, the winners also take a \$ 10.000 prize that supposedly is given by a fund managed by Gordon Bell itself.

3.1 Prize Criteria and Current Committee Members

In order to apply for the award, the authors need to specifically submit the article for the prize committee. Table 3.1 show the current committee along with the title of each member.

Table 3.1: Current Committee Members for the ACM Gordon Bell Award

Title	Name
Chair	Subhash Saini
Member	Arndt Bode
Member	Bronis De Supinski
Member	Michael Heroux
Member	Satoshi Matsuoka
Member	Keshav Pingali

The articles are evaluated by the committee which primarily goal is to recognize performance achievements that demonstrate:

- Evidence of important algorithmic and/or implementation innovations
- Clear improvement over the previous state-of-the-art
- Solutions that don't depend on one-of-a-kind architectures (systems that can only be used to address a narrow range of problems, or that can't be replicated by others)
- Performance measurements that have been characterized in terms of scalability (strong as well as weak scaling), time to solution, efficiency (in using bottleneck resources, such as memory size or bandwidth, communications bandwidth, I/O), and/or peak performance
- Achievements that are generalizable, in the sense that other people can learn and benefit from the innovations

In earlier years, multiple prizes were sometimes awarded to reflect different types of achievements. According to current policies, the Prize can be awarded in one or more categories, depending on the entries received in a given year.

The following chapters will present a detailed analysis of the winners (chapter 4) and the development of the field (chapter 5) in the last years.

Prize details

This chapter will discuss in better details about all the awards presented so far.

In order to make this discussion, we will begin presenting in section 4.1 the list of the categories awarded since the prize started in 1987.

Following that, we will list (in section 4.2) a relation of the award winners in the last decade.

Finally we will present a brief analysis for those awards in section 4.3

4.1 Prize categories

After performing an analysis on the award winners, we can say that the ACM Gordon Bell Prizes are centralized in few categories.

Those categories are defined by the award committee and table 4.1 shows a list of the categories awarded so far, with the inclusion of our abbreviation for each categories.

Table 4.1: List of abbreviations for the categories names

Abbreviation	Full category name
AI	Algorithm Innovation
BP	Best Performance of a High Performance Application
OA	Outstanding Achievement in High-performance Computing
P	Performance
PP	Peak Performance
P/P	Price / Performance
SA	Special Achievement
SC	Special Category
SP	Sustained Performance
STS	Scalability and Time to Solution

4.2 Award winners in the last decade

As it was mentioned before, each year a committee from the Award Board 3.1 selects several articles that attend to the criteria mentioned in chapter 3.

After this initial selection, the committee choose one or more articles to receive the prize. Table 4.2 shows a list of winners in the last decade.

As we can see, there are years, like 2011, in which the committee granted two prizes at the same year and even three in 2009.

Each article that pass the initial selection receives a prize of $ 100, and each winner receives a prize of $ 10.000.

4.3 Award Analysis

Based on the data presented in table 4.2 and the data for the remaining winners, we plotted the distribution of the winners according with the category of acceptance.

We can see that during 1987–2016 most of the article winners was in category called *Peak Performance* with 20 award winners 4.1.

Following that we have:

Table 4.2: List of Categories that were winner each year

Year	Cat	Title
SC16	OA	10M-core scalable fully-implicit solver for nonhydrostatic atmospheric dynamics.
SC15	OA	An extreme-scale implicit solver for complex PDEs: highly heterogeneous flow in earth's mantle.
SC14	BP	Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer
SC13	BP	11 PFLOP/s simulations of cloud cavitation collapse.
SC12	STS	4.45 Pflops astrophysical N-body simulation on K computer: the gravitational trillion-body problem
SC11	SP	First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer
	STS	Peta-scale phase-field simulation for dendritic solidification on the TSUBAME 2.0 supercomputer.
SC10	P	Petascale Direct Numerical Simulation of Blood Flow on 200K Cores and Heterogeneous Architectures.
SC09	P/P	42 TFlops hierarchical N-body simulations on GPUs with applications in both astrophysics and turbulence
	PP	A scalable method for ab initio computation of free energies in nanoscale systems.
	SC	Millisecond-scale molecular dynamics simulations on Anton.
SC08	PP	New algorithm to enable 400+ TFlop/s sustained performance in simulations of disorder effects in high-Tc superconductors.
	AI	Linearly scaling 3D fragment method for large-scale electronic structure calculations.
SC07	P	Extending stability beyond CPU millennium: a micron-scale atomistic simulation of Kelvin-Helmholtz instability
SC06	PP	Large-scale electronic structure calculations of high-Z metals on the BlueGene/L platform.
	SA	The BlueGene/L supercomputer and quantum ChromoDynamics.

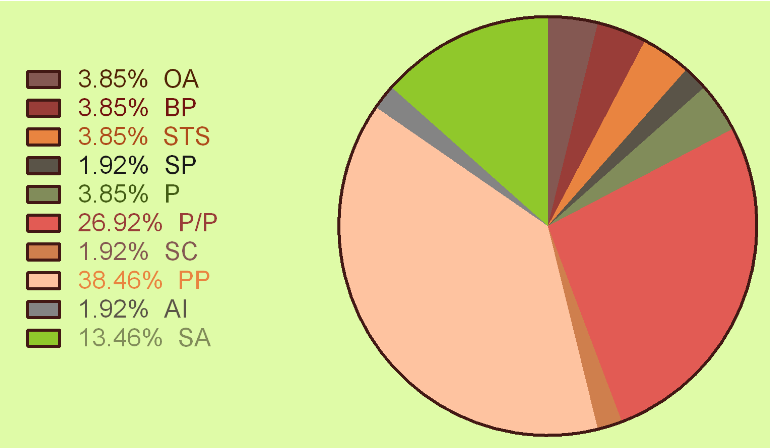


Figure 4.1: The distribution of Categories Winners 1987 – 2016

- Price/Perfomance with 14 paper winners
- Special Achievement with 7 paper winners
- Outstanding Achievement in High-performance Computing with 2 paper winners
- Best performance with 2 paper winners
- Scalability and Time to Solution with 2 paper winners
- Performance with 2 paper winners
- Sustained performance with 1 paper winner
- Special Category with 1 paper winner
- Algorithm Innovation with 1 paper winner

Also, based on the University of the authors, we plotted the award distribution around the globe.

Figure 4.2 shows this distribution. It is important to mention that this plot is a little misleading, since there are many winners with a great number of authors.

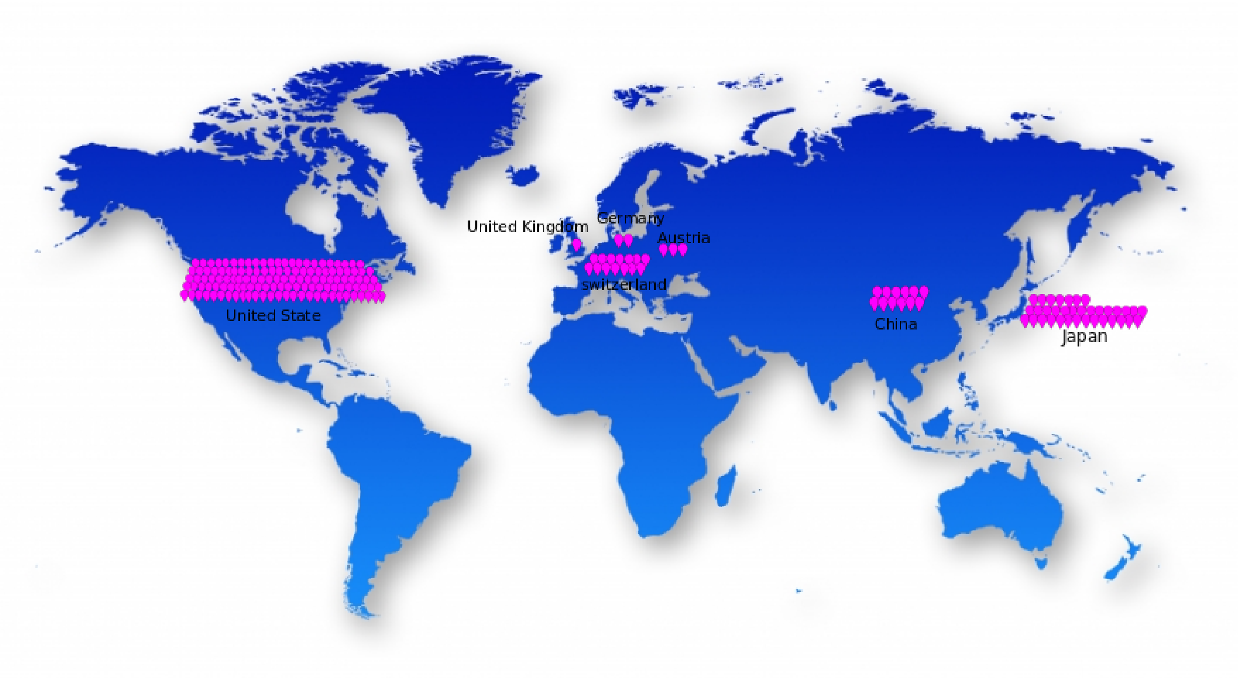


Figure 4.2: The distribution of Countries winner 2006 – 2016

As an example, China has 12 winners, but all of those come from just one award in the last year.

Table 4.3 briefly resumes the data presented in figure 4.2

Table 4.3: Top 7 country award winners

Position	Country	Number of awards	Number of researchers
1st	USA	10	129
2nd	Japan	4	35
3rd	Switzerland	3	14
4th	China	1	12
5th	Austria	1	3
6th	Germany	1	2
7th	United Kingdom	1	1

Field Development

Established in 1988, the annual SC conference has built a diverse community of participants including researchers, scientists, application developers, computing center staff and management, computing industry staff, agency program managers, journalists, and congressional staffers.

This diversity is one of the conference's main strengths, making it a yearly "must attend" forum for stakeholders throughout the technical computing community.

The technical program is the heart of SC. It has addressed virtually every area of scientific and engineering research, as well as technological development, innovation, and education.

Its presentations, tutorials, panels and discussion forums have included breakthroughs in many areas and inspired new and innovative areas of computing.

This chapter will make an analysis of the development of the High Performance Computing field since the start of the Gordon Bell Award.

In section 5.1 we present the current acceptance rate for the articles submitted to the SC conference.

Following that, section 5.2 will present a analysis of the number of articles published per category since the creation of the prize.

5.1 Super Computing conference acceptance rate

Table 5.1 presents the number of articles published in each annual conference from its creation until today, according with the data gathered from ACM.

We can see that the overall acceptance rate is **24%** ($1,455$ of $6,046$ submissions).

Figure 5.1 shows a graphical representation of this overall acceptance divided by year.

5.2 Development along the years

As we have said in the previous chapter, the prize is divided in few categories 4.1 and we selected and analyzed 8 of those categories.

From those categories, we were able to obtain an approximate number of articles from the creation of the conference until today.

We did this by applying a series of filters on Google Scholar with the purpose of having a research interest approach according to categories.

The data gathered is presented in table 5.2 and from that we could make few affirmations:

- Performance, was the category with the greatest interest in publications;
- Scalability and Time Solution, was the category with the highest growth, reaching to exceed in the last period a sustained performance and Price/Performance;
- Algorithm Innovation did not achieve significant growth in the last two periods;
- Best Performance, was the category that kept the research interest during the 3 periods.

Table 5.1: Paper Acceptance Rate

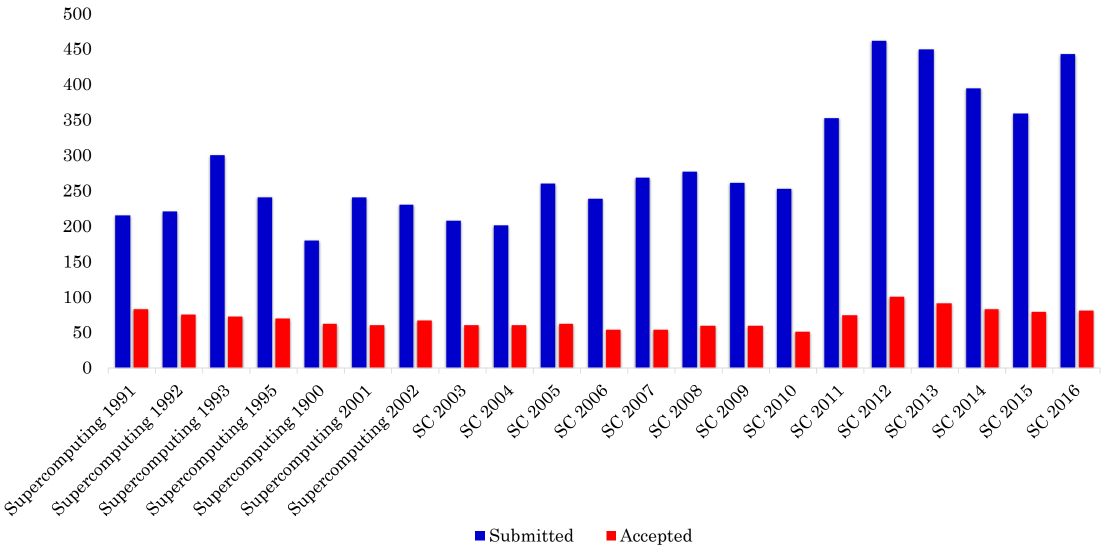
Year	**Submitted**	**Accepted**	**Rate**
Supercomputing 1991	215	83	39%
Supercomputing 1992	220	75	34%
Supercomputing 1993	300	72	24%
Supercomputing 1995	241	69	29%
Supercomputing 2000	179	62	35%
Supercomputing 2001	240	60	25%
Supercomputing 2002	230	67	29%
SC 2003	207	60	29%
SC 2004	200	60	30%
SC 2005	260	62	24%
SC 2006	239	54	23%
SC 2007	268	54	20%
SC 2008	277	59	21%
SC 2009	261	59	23%
SC 2010	253	51	20%
SC 2011	352	74	21%
SC 2012	461	100	22%
SC 2013	449	91	20%
SC 2014	394	83	21%
SC 2015	358	79	22%
SC 2016	442	81	18%
Overall	**6,046**	**1,455**	**24%**

Source: ACM (2017)

Table 5.2: Publications per category between the years 1987 - 2016

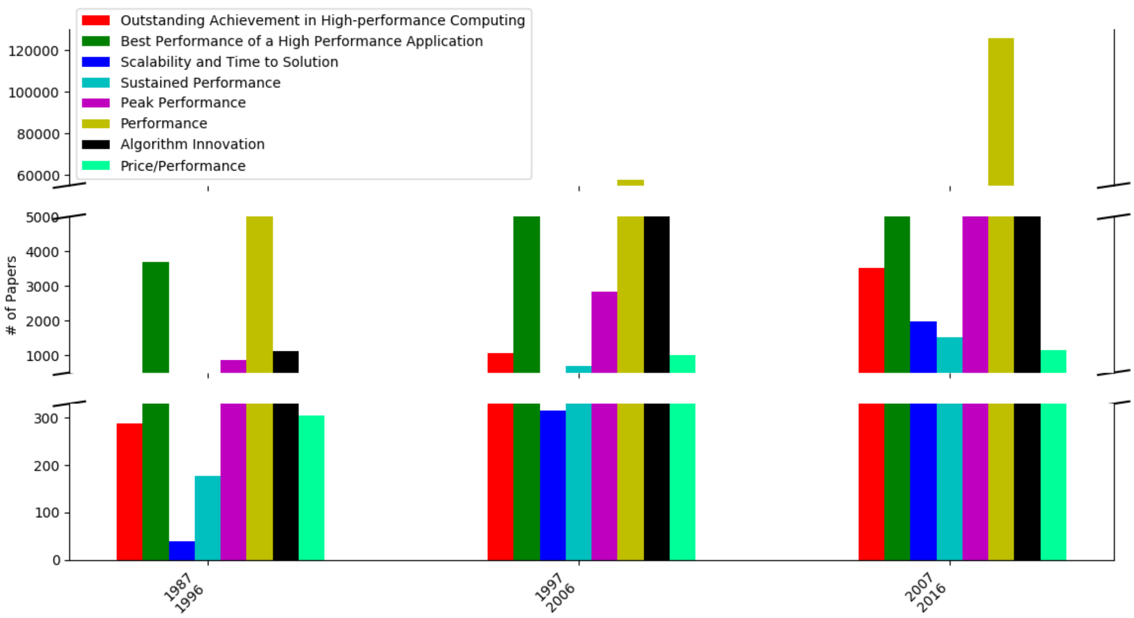
Cat	**1987 1996**	%	**1997 2006**	%	**2007 2016**	%
OA	288	1,37	1.060	1.35	3.520	1.97
BP	3.680	17.55	9.560	12.13	17.200	9.63
STS	38	0.18	316	0.40	1.970	1.10
SP	177	0.84	683	0.87	1.520	0.85
PP	869	4.14	2.830	3.59	10.200	5.71
P	14.500	69.15	57.600	73.10	126.000	70.53
AI	1.110	5.29	5.740	7.28	17.100	9.57
P/P	306	1.46	1.010	1.28	1.150	0.64
Total	20968	100	178.660	100	278.427	100

Figure 5.1: Paper Acceptance Rate



Source: ACM (2017)

Figure 5.2: Publications per category between the years 1987 - 2016



Article award analysis

Now we will discuss the state-of-the-art research in High Performance Computing, by presenting a brief analysis of three Gordon Bell Award winners.

Initially our intention was to present a quick overview for all the winners, or at least, for all the most recent ones.

However, that would not only be a very demanding task, as also the objective for this chapter is to just give a general idea of the type of the work that are awarded by this prize.

Because of that, we decided that instead of presenting the short overview for all articles, we could select few and present a combination of short and detailed discussion for this selection.

Based on that, the following sections will present two short discussions (2016 and 2009) that brings a brief justification for the article selection as also a quick overview and conclusion.

It is in section 6.3 that we present a more detailed discussion, with the intention of utilize the opportunity to look deeper into one of the state-of-the-art in the HPC field.

Table 6.1 summarizes the list of articles that will be discussed next in which the column type resumes the level of details that the section will be presented for a given article. Also, a list of the complete names for each categorie can be found at 4.1.

Table 6.1: List of the articles that will be addressed in the following sections

Year	Type	Cat	Title
2016	Briefly	OA	10M-core scalable fully-implicit solver for nonhydrostatic atmospheric dynamics
2010	With Details	P	Petascale Direct Numerical Simulation of Blood Flow on 200K Cores and Heterogeneous Architectures
2009	Briefly	SC	Millisecond-scale molecular dynamics simulations on Anton

6.1 2016 Article Award

The motivation for this article choice was to present the most recent winner (by the time of this writing).

The 2016 award (Yang et al. (2016)) was given in the category of Outstanding Achievement in High-performance Computing (OA).

6.1.1 Core Ideas

The core ideas behind this article was:

- Shows a benchmark for the TOP 1 Super Computer (Sunway TaihuLight)
- Present its performance when computing in a fully-implicit solver for hyperbolic conservation laws in nonhydrostatic atmospheric dynamics
- Describe the algorithm for this solver (including scalability, topology, ...)

6.1.2 Article conclusions

Since the core ideas behind the article was to present a benchmark for the algorithm described along the text, their main conclusions orbit around this. One can say that their main conclusions can be summarized in:

- Provide the benchmark to show why Sunway TaihuLight SC (STL-SC) is the top 1
- Show that it was able to perform 7.95 PFLOPS in double-precision with over than 10.5M heterogeneous cores
- Computed over than 770 billion unknowns with 0.07 simulated-years-per-day

To finish, it is also interesting to notice how the difficult of the task impact in the number of PFLOPS. We can see from the conclusions above that during the execution it was able to perform 7.95 PFLOPS against the 93 PFLOPS it was able to perform by testing over Linpack.

6.2 2009 Paper Award

Millisecond-scale Molecular Dynamics Simulations on Anton (Shaw et al. (2009)) was selected as a winner from Special Category (SP) in 2009.

The motivation for choosing this article was to state the importance of the High Performance Computing in biomolecular systems.

Also, we had special interest in Anton, that is a special-purpose supercomputer designed for molecular dynamics (MD) simulations of biomolecular systems.

In 2009, the machine's specialized hardware increased the speed of Molecular Dynamics calculations and, for the first time, biological molecules could be simulated at an atomic level of detail for periods on the order of a millisecond.

That result was impressive since it was about two orders of magnitude beyond the previous state-of-the-art supercomputers at the time.

6.2.1 Core Ideas

The core ideas behind this article was:

- To give ability to scientists to trace atomic motions and help them to produce deep insights into molecular mechanisms that experimental approaches could not have achieved alone.
- To give power to scientists to perform simulations at rates above 15 microseconds.
- To maximize the use of specialized hardware.

6.2.2 Overview of the Anton Architecture

As we can see from figure 6.1, the architecture of Anton comprises a set of nodes connected in a toroidal topology and the 512-node machines was the topic discussed in the paper.

Also, we can see that each node includes an ASIC with two major computational subsystems 6.1.

The first is the high-throughput interaction subsystem (HTIS) designed for computing massive numbers of range-limited pairwise interactions of various forms using an array of 32 hardwired pairwise point interaction pipelines (PPIPs).

This PPIPs can compute interactions with a number of different functional forms.

The second is the flexible subsystem, which is composed of programmable cores used for the remaining, less structured part of the MD calculation.

This flexible subsystem contains eight geometry cores (GCs) that were designed in-house to perform fast numerical computations, four Tensilica LX processors that control the overall data flow in the Anton system, and four data transfer engines that allow communication to be hidden behind computation.

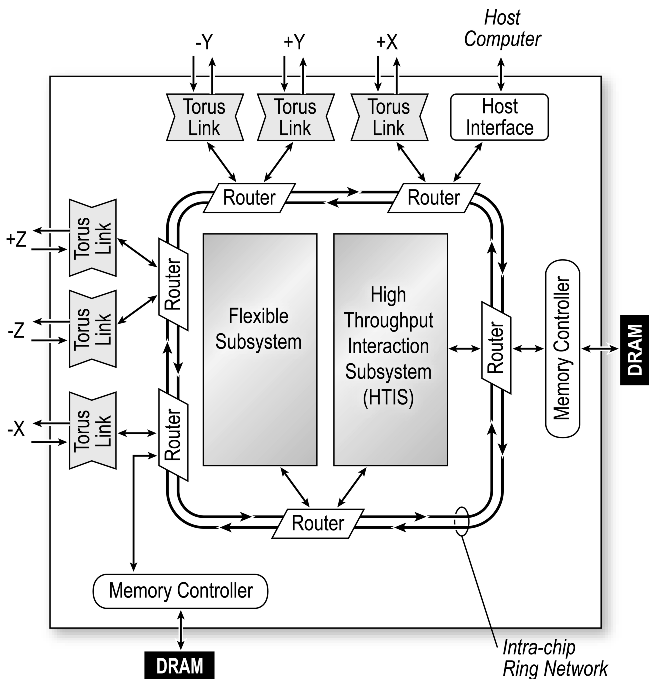


Figure 6.1: Anton ASIC block diagram

6.2.3 Maximizing the use of specialized hardware

To accelerate a computation with specialized hardware, one typically begins by looking for an "inner loop" that accounts for the majority of the computational time. Figure 6.2Figure 6.2 shows an execution time profile of the widely used GROMACS MD package on an x86 core.

	x86 core		Anton	
	small cutoff (9 Å) fine mesh (64^3)	**large cutoff (13 Å) coarse mesh (32^3)**	**small cutoff (9 Å) fine mesh (64^3)**	**large cutoff (13 Å) coarse mesh (32^3)**
Nonbonded forces				
Range-limited forces	56.6 ms (64%)	164.4 ms (89%)	1.4 μs (4%)	1.9 μs (12%)
FFT & inverse FFT	12.3 ms (14%)	1.4 ms (1%)	24.7 μs (63%)	8.9 μs (58%)
Mesh interpolation	9.6 ms (11%)	8.8 ms (5%)	9.5 μs (24%)	2.0 μs (13%)
Correction forces	4.0 ms (5%)	3.8 ms (2%)	2.5 μs (6%)	2.5 μs (16%)
Bonded forces	2.7 ms (3%)	2.7 ms (1%)	3.5 μs (9%)	4.1 μs (27%)
Integration	3.4 ms (4%)	3.4 ms (2%)	1.6 μs (4%)	1.6 μs (10%)
Total	88.5 ms (100%)	184.5 ms (100%)	39.2 μs (100%)	15.4 μs (100%)

Figure 6.2: Effect of electrostatics parameters on performance, illustrated by execution time profiles for GROMACS on a single x86 core and for Anton

Figure 6.2 shows a typical simulation where the x86 core spends 64% of its time computing range-limited forces electrostatic and van der Waals interactions between pairs of atoms separated by less than 9 Å.

These computations are mapped to Anton's PPIPs, which accelerate the computations by over two orders of magnitude relative to a conventional processor core.

Of the tasks listed in 6.2, only the Fourier computation, bonded force evaluation and integration are mapped to Anton's flexible cores. These tasks together constitute only 4% of the x86's execution profile with typical Anton parameters.

The remaining workload is mapped to Anton's fast, specialized PPIPs.

Through a combination of novel algorithms, parameter optimization, and hardware optimization, they significantly reduced the computational workload mapped to Anton's programmable cores.

6.2.4 Article conclusions

The main conclusions of this article are:

- Commodity computing benefits from economies of scale but imposes limitations on the extent to which an MD simulation can be accelerated through parallelization.
- To present a combination of a unique hardware architecture and carefully CO-DESIGN algorithms that gave power to scientist to perform MD way faster than before.

At the time of the publication, Anton gave the ability to the scientists, for the first time, to perform MD simulations on the order of a millisecond.

That was two orders of magnitude longer than any atomically detailed simulation reported on general-purpose hardware and three orders of magnitude longer than any simulation reported previously on special-purpose hardware.

6.3 2010 Paper Award

The motivation for this article choice was to present the category with the highest number of publications.

The 2010 award (Rahimian et al. (2010)) was given in the category of Performance (P).

6.3.1 Core Ideas

The general ideas behind the paper was:

- Present a new computational infrastructure, MoBo[1], for Stokesian[2] particulate flows.
- Accurately resolve hydrodynamic interactions between RBC's[3];
- Allow for the highly non-uniform distribution of RBCs in space.

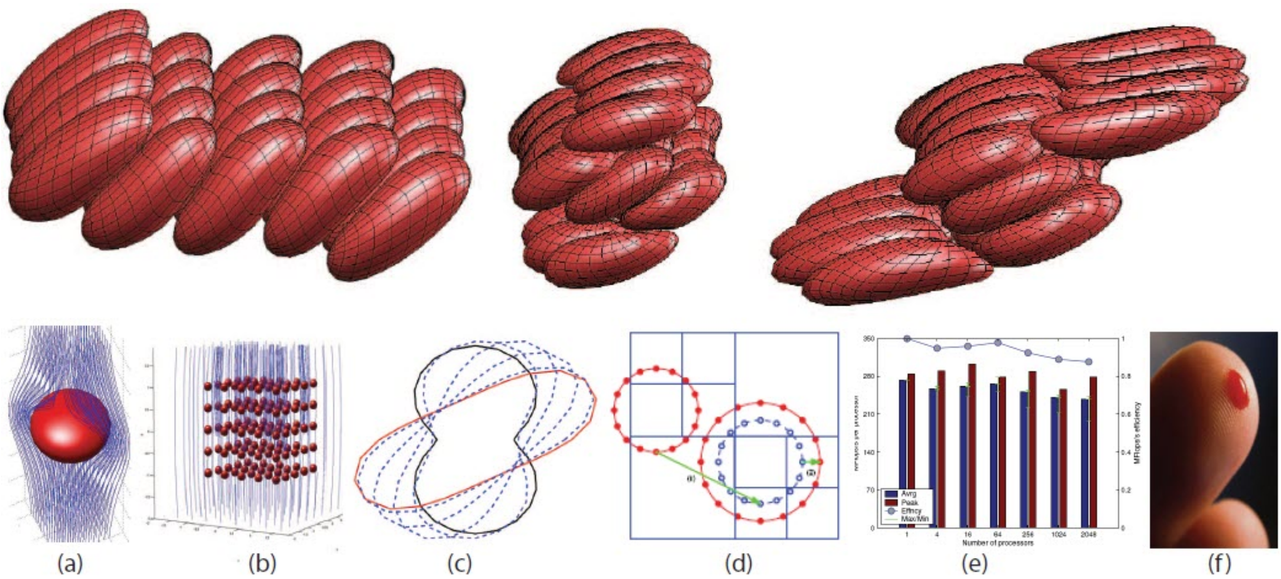
This new method has been implemented in the software library MoBo, designed to support parallelism at all levels, including inter-node distributed memory parallelism, intra-node shared memory parallelism, data parallelism (vectorization), and fine-grained multithreading for GPUs (Intel/AMD x86 and NVidia Tesla/Fermi platforms) for single and double floating point precision. MoBo is a new computational infrastructure that enables the direct numerical simulation of several microliters of blood (Just one microliter of blood of a healthy individual contains approximately four million RBCs).

[1]Moving Boundaries.
[2]Stokesian dynamics is a solution technique for the Langevin equation.
[3]Red Blood Cells.

6.3.2 Summary of the Computational Infrastructure for Direct Numerical Simulation of Blood Flow

Figure 6.3 explain the technique overview of this paper. In the top row, it's depict a few snapshots from the flow of twenty RBCs that are immersed in plasma (which is not visualized). At every time step, a Stokes problem must be solved in the exterior and interior of the RBCs. This is quite challenging, first, because of the complex geometries and second because the Stokes equations require implicit solvers. The authors developed computational tools for the efficient direct numerical simulation of blood using a boundary integral formulation that addresses some of the numerical approximation issues. The main algorithmic components include: 6.3-(a) spectral RBC shape representations and quadratures for singular integrals on these shapes; 6.3-(b) accurate modeling of the hydrodynamic interactions between many-RBCs; 6.3-(c) nonlinear solvers for the mechanics of RBC deformations; 6.3-(d,e) parallel, kernel-independent, treebased, fast summation methods. The advantage of boundary integral methods is that only the RBC boundary is discretized and no discretization of the space between RBCs is necessary. This is crucial for reducing the number of degrees of freedom and eliminates the need for difficultto-parallelize 3D unstructured mesh generation. These tools enable parallel and highly accurate simulations of microcirculation phenomena of blood flow. They have achieved the direct numerical simulation of $O(50)$ microliters of blood flow; and 6.3-(f) One can think of the volume of a single blood drop as being roughly equivalent to one microliter.

Figure 6.3: Technique Overview



Source: Rahimian et al. (2010)

6.3.3 Compare to Results

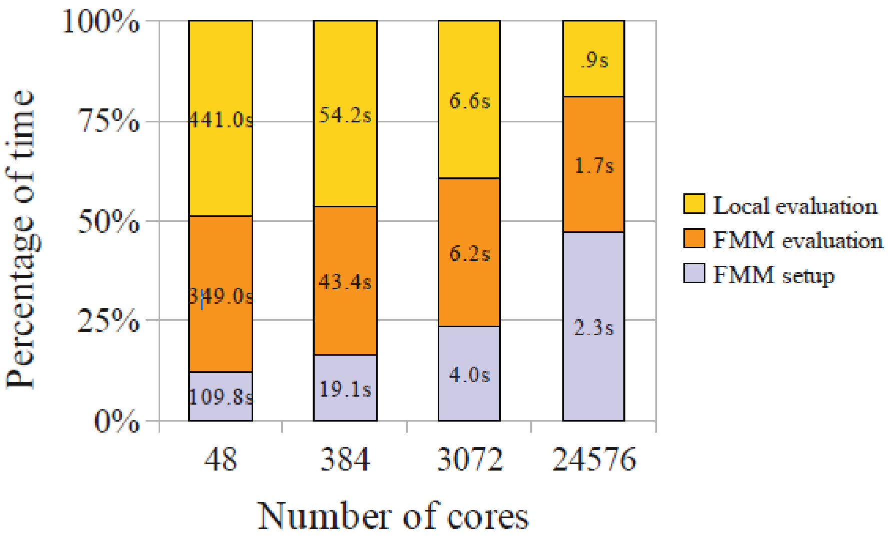
Platforms and architectures: The large scale weak and strong scalability results have been obtained on the Jaguar PF platform at the National Center for Computational Sciences (UT/ORNL), a Cray XT5 system with 224,256 cores (2.6 GHz hex-core AMD Opteron, 2GB/core) and a 3D-torus topology. Jaguar is ranked first in the top-500 list of supercomputers (`www.top500.org`) as of April of 2010. The GPU scalability results have been obtained on TeraGrid's

Lincoln at the National Center for Supercomputing Applications (UIUC/NSF), a Dell cluster with NVIDIA Tesla S1070 accelerators, 1536 cores(Intel Harpertown/2.33 Ghz dual-socket quad-core 2GB/core),384 GPUs (4GB/GPU), and InfiniBand (SDR) interconnect. The results on Fermi were obtained on a single node AMD machine at ORNL. The Nehalem tests where performed in an in-house 8-node cluster, with 16 sockets and one NVIDIA T10P-based GPU per socket. In all of the experiments on Jaguar, they used one MPI process per socket and six threads per socket. Both local and global interactions calculations have been multithreaded using OpenMP. Also, inn all of our GPU experiments, they used one MPI process per socket.

MPI, strong scalability tests on Jaguar: The results are reported in Figure 6.4. The problem size is 300,000 RBCs with 84 points per RBC, which corresponds to 100,000,000 unknowns. The strong scalability results demonstrate excellent speed up resulting in an unprecedented five seconds per time-step on 24,576 cores. The strong scalability result for 262,144 vesicles, and total number of 22M grid points. There are 6 cores (and 6 OpenMP threads) per MPI process. The finest level of the octree is nine and the coarsest is three.

Figure 6.4: Strong Scalings on Jaguar PF

p	48	384	3072	24576
Time (sec)	899.8	116.7	16.7	4.9
Efficiency	1.00	0.96	0.84	0.35



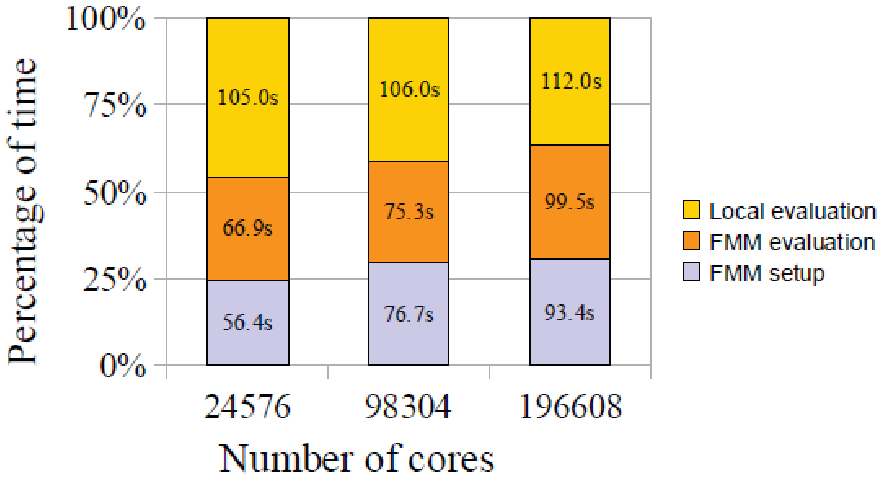
Source: Rahimian et al. (2010)

They got excellent speed up and we require less than 10 seconds per time step for 300,000 RBCs. The efficiency, of course, is reduced for the largest processor count as the memory traffic dominates the computations.

MPI, weak scalability tests on Jaguar: The results are reported in Figure 6.5. The problem size (number of RBCs) per core is kept fixed to 8000 RBCs, again with 84 points per RBC for the line-distribution on the Poiseuille flow. We can observe the the calculation of local interactions remains almost constant, whereas the cost of the tree-setup and global interactions increase. The setup is not multithreaded. Despite, the need for further optimizations, they achieved good utilization of the resources: the local interactions phase sustains over 18 GFlops/s per core (single precision) and the global interactions phase sustains over1.2GFlops/per core (double precision). Overall, the code exceeds 0.7 Petaflops of sustained performance.

Figure 6.5: Weak Scalings on Jaguar PF

p	24576	98304	196608
Time (sec)	228.3	258	304.9
Efficiency	1.00	0.88	0.75



Source: Rahimian et al. (2010)

6.3.4 Article conclusions

One can say that their main conclusions can be summarized in:

- It was presented MoBo, a software library that enables simulations of blood;
- It was showed that is possible scale the different parts of the method, observing good scalability across different architectures;
- This study showed that MoBo opens the way for blood flow simulations of hight fidelity.

To finish, MoBo in 2010 opened the way for blood flow simulations of unprecedented fidelity. It enabled the simulation of microfluidic and nanofluidic devices. For example, the 2D version of MoBo, had already resulted in significant scientific discoveries Kaoui et al. (2009).

Bibliography

ACM, I. (2017). ACM Proceedings. http://dl.acm.org/proceedings.cfm. [Online; accessed 2017-06-22].

Kaoui, B., Biros, G., and Misbah, C. (2009). Why do red blood cells have asymmetric shapes even in a symmetric flow? *Phys. Rev. Lett.*, 103:188101.

Rahimian, A., Lashuk, I., Veerapaneni, S., Chandramowlishwaran, A., Malhotra, D., Moon, L., Sampath, R., Shringarpure, A., Vetter, J., Vuduc, R., Zorin, D., and Biros, G. (2010). Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '10, pages 1–11, Washington, DC, USA. IEEE Computer Society.

Shaw, D. E., Dror, R. O., Salmon, J. K., Grossman, J., Mackenzie, K. M., Bank, J. A., Young, C., Deneroff, M. M., Batson, B., Bowers, K. J., Chow, E., Eastwood, M. P., Ierardi, D. J., Klepeis, J. L., Kuskin, J. S., Larson, R. H., Lindorff-Larsen, K., Maragakis, P., Moraes, M. A., Piana, S., Shan, Y., and Towles, B. (2009). Millisecond-scale molecular dynamics simulations on anton. *SC09*.

Yang, C., Xue, W., Fu, H., You, H., Wang, X., Ao, Y., Liu, F., Gan, L., Xu, P., Wangk, L., Yang, G., and Zheng, W. (2016). 10m-core scalable fully-implicit solver for nonhydrostatic atmospheric dynamics. *SC16*.