Evaluation of HPC architectures for BRAMS numerical weather model

Eugenio Sper de Almeida[1,2]**, Michael Bauer**[2]**, and Alvaro Luiz Fazenda**[3]
[1]Center for Weather Forecast and Climate Studies, National Institute for Space Research, Cachoeira Paulista, SP, Brazil
[2]Department of Computer Science, The University of Western Ontario. London, ON, Canada
[3]Institute of Science and Technology, Federal University of São Paulo, São José dos Campos, SP, Brazil

Abstract - *This paper investigates the performance of a weather forecasting application (Brazilian Regional Atmospheric Modeling System - BRAMS) on a number of selected HPC clusters in order to understand the impact of different architectural configurations on its performance and scalability. We simulated atmosphere conditions over South America for 24 hours ahead with BRAMS, using 100 cores as a starting point (100 cores step). An extra set of executions took place from 10 to 100 cores (10 cores step) to identify more details about BRAMS performance. Results reveal differences in BRAMS performance and its relationship with interconnection (technology and topology). In conclusion, interconnection can limit application performance even with code improvement.*

Keywords: Performance, BRAMS, Numerical Weather Prediction (NWP) model, High Performance Computing (HPC), parallel processing, multi-core architecture.

1 Introduction

Increasing resolution has resulted in improved model simulations and predictions of key atmospheric phenomena [1]. As a result, the execution time of Numerical Weather Prediction (NWP) models increase exponentially as the number of grid points increase in the x, y and z directions [2]. This can lead to delays in the timely delivery of meteorological information, resulting in the actual occurrence of the atmospheric phenomena before it can be predicted.

The configuration of HPC resources are critical in ensuring that sufficient computing and communication resources are available to deliver enough performance for the timely use of NWP models. The exponential improvement in the accuracy of these computational models, however, represents a challenge for many meteorological centers. Consequently, NWP models must be tailored to get the best performance provided by an HPC system.

Recently, Fazenda et. al. [3] identified limitations in BRAMS (Brazilian Regional Atmospheric Modeling System) scalability due to algorithm implementation. They identified bottlenecks in BRAMS code and developed new solutions, leading to a decrease of BRAMS execution time and a gain of scalability on HPC clusters. In addition, they developed an efficient solution for parallelism scalability, showing performance gains up to 700 cores.

In analyzing the Weather Research and Forecasting (WRF) Model performance, a NWP similar to BRAMS, [4] stated that choosing the right interconnect technology was essential for maximizing HPC system efficiency. Slow interconnects delay data transfers between servers slowing execution of simulations and causing inefficient utilization of computational resources. Their results, using 24 servers each with two AMD Quad-Core processors, identified WRF's communication-sensitive points and demonstrated its dependency on high-speed networks and fast CPU to CPU communication.

According to [5], a communication bottleneck in an HPC cluster may lead to a significant loss of overall performance and so network communication is another key factor that affects application performance on HPC clusters.

Rodrigues et al [6] show the impact of applying a process mapping approach in the BRAMS model, since the communication link speeds on a specific cluster vary with process selection. They developed a method to obtain close to optimal application process placement on cluster cores.

Clusters with Intel EM64T (78.4%) and AMD x86_64 (11.4%) processors dominate the TOP500 list [7], a ranked list of general purpose systems of common use for high end applications. These systems use a number of different interconnection technologies: Gigabit Ethernet (45,6%), Infiniband (42,6%), Myrinet (0.8%) or Quadrics (0.2%). Even though only 0.20% of the HPC systems on the TOP500 list report that their interconnection topology is a fat tree, it is likely that many of them build their systems with this topology using Gigabit Ethernet, Infiniband, Myrinet or Quadrics interconnection technology.

In a fat tree network, processors may be interconnected by a tree structure, in which the processors are at the leaves of the

tree, and the interior nodes are switches. When one moves up the tree from leaves to the root, the links become "fatter" [8]. An advantage of a tree structure is that communication distances are short for local communication patterns. A drawback, however, is that the root and higher-level nodes become bottlenecks for more global communication.

This paper investigates BRAMS performance and scalability on a number of different clusters available within SHARCNET (Shared Hierarchical Academic Research Computing NETwork) [9]. BRAMS is a limited area forecast model that runs on a broad range of computational systems: from mono-processor desktops to clusters with many processors. We evaluate the BRAMS performance on GigaBit Ethernet, Infiniband, Quadrics, and Myrinet networks, as well as in different AMD and Intel dual-core and quad-core architectures. In Section 2 we describe BRAMS and SHARCNET. We describe the experiments in Section 3. Performance results from BRAMS execution on different HPC clusters are presented in Section 4 and conclusions are provided in Section 5.

2 BRAMS and SHARCNET overview

The SHARCNET is a consortium of 17 academic and research organizations in Ontario whose primary mandate is to provide shared high performance computing facilities and associated services to enable forefront computational research.

Clusters are the main SHARCNET resources and basically serve for two categories of computing programming models: those allowing serial (non-parallel) application to take advantage of a clusters parallelism and those with explicit parallelization of a program [10]. SHARCNET clusters have different interconnection networks and types of AMD and Intel architectures, based on dual-core and quad-core processor chips.

The Lightweight Directory Access Protocol (LDAP) controls account management, enabling a researcher to access to any of the systems through a single account. On each cluster, the Load Sharing Facility (LSF) performs job scheduling [11]. As a user account belongs to a global storage system, codes compiled on a user account can be executed on any appropriate SHARCNET cluster.

BRAMS, a version of the RAMS [12][13] tailored to the tropics, has explicit parallelization. The BRAMS/RAMS model is a multipurpose numerical weather model designed to simulate atmospheric circulations, well suitable for HPC clusters. Analysis and boundary conditions from an atmospheric global circulation model are the data input for BRAMS simulation, which is governed by a RAMSIN parameter definition file. It contains all parameterization related to a specific simulation [14].

3 Experiments

"Downscaling" refers to a technique used to achieve detailed regional and local atmospheric data by using either fine spatial-scale numerical atmospheric models (dynamical downscaling), or statistical relationships (statistical downscaling). An Atmospheric Global Circulation Model (AGCM) run is typically the starting point for downscaling. The downscaled high resolution data can also then be inserted into other types of numerical simulation tools such as hydrological, agricultural, and ecological models [15].

Many meteorological centers in Brazil use INPE/CPTEC AGCM outputs as input for their regional area models, consequently providing a more accurate forecast at regional and local scale. This AGCM runs four times a day (00, 06, 12 and 18 UTC) providing numerical weather forecast outputs for 15 days ahead with resolution T162L28 mode; T refers to spectral truncation type (triangular) in zonal wave 62 (resolution of 100x100 km) and L refers to the number of vertical levels (28 levels) [16].

We simulated this downscaling approach (Figure 1), with the BRAMS model, to forecast weather 24 hours ahead in a spatial resolution of 20x20 km over South-America (grid size of 340 by 370 horizontal points). The analysis and boundary conditions for the BRAMS model came from INPE/CPTEC AGCM model outputs from October 23, 2010.

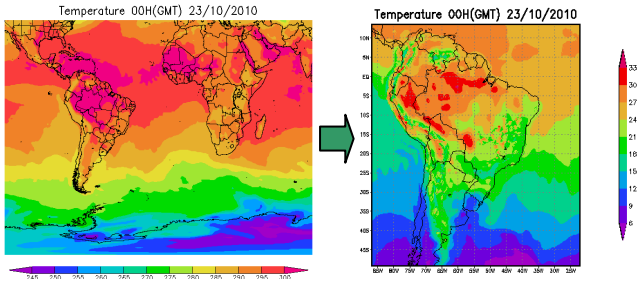

Figure 1. BRAMS downscaling.

NWP models run daily on meteorological centers on HPC resources, at a predetermined window time as part of their operational suite. Scientific visualization tools convert NWP outputs to meteorological maps, meteorologists analyze those maps to produce meteorological forecast and finally publish on meteorological center website for the society.

In order to understand the performance of the BRAMS model, we benchmarked BRAMS execution time starting with 100 cores and incrementing the number of cores by 100. In order to have a closer look at the network influence over the BRAMS performance, we perform additional executions up to 100 processors (incrementing the number of cores by 10). This experiment took place on selected SHARCNET HPC clusters:

- **Bull (384 cores):** HP Linux cluster running XC 3.1 with 96 nodes, four Opteron Mono-Core processor @ 2.4 GHz (QsNet-2/Elan4), and 32 GB of memory;
- **Saw (2688 cores):** HP Linux cluster running XC 4.0 (RHEL 5.1) with 336 nodes, two Xeon Quad-Core processors @ 2.83 GHz (Infiniband), and 16 GB of memory;
- **Requin (1536 cores):** HP Linux cluster running XC 3.1 with 768 nodes, one Opteron Dual-Core processor @ 2.6 GHz (QsNet-2/Elan4), and 8 GB of memory;
- **Narwhal (1068 cores):** HP Linux cluster running XC 3.1 with 267 nodes, two Opteron Dual-Core processor @ 2.2 GHz (Myrinet 2g-gm), and 8 GB of memory;
- **Whale (3072 cores):** HP Linux cluster running XC 3.2.1 with 768 nodes, two Opteron Dual-Core processor @ 2.2 GHz (GigabitEthernet), and 4 GB of memory.

"Bull" and "Narwhal" have direct connected topology interconnects. Fat tree topology interconnects exist on "Saw" (three layers with 2:1 oversubscription), "Requin" (two layers) and "Whale" (three layers) nodes. Table I presents information about switch type, and nominal latency/bandwidth of the selected SHARCNET HPC clusters.

Table I. Latency and bandwidth of SHARCNET clusters interconnection.

Cluster	**Interconnection features**		
	Switch type	***latency (μs)***	***bandwidth (MB/s)***
Saw	InfiniBand/DDR	1.3	1600
Requin	QsNet2/Elan4	1.4	900
Bull	QsNet2/Elan4	1.4	900
Narwhal	Myrinet 2g (GM)	3.8	250
Whale	GigabitEthernet	50	120

By measuring and comparing BRAMS performance, we extend previous performance analysis from [3]. We compiled BRAMS code using Fortran90/C compilers from Intel and HPMPI libraries.

4 Performance results and discussions

Message sizes, exchanged by the computing nodes of a HPC cluster, decrease when increasing the number of cores for BRAMS execution.

We identify differences in BRAMS execution time (Figure 2) when increasing the number of cores and we order the HPC clusters according to the best execution time of BRAMS:

- "Requin", "Bull", "Narwhal", and "Saw" up to 60 cores;
- "Requin", "Saw", and "Bull" from 70 cores to 80 cores;
- "Saw", "Requin", "Bull", and "Whale" from 90 cores to 200 cores;
- "Saw", "Bull" "Requin", and "Whale" for more than 300 cores.

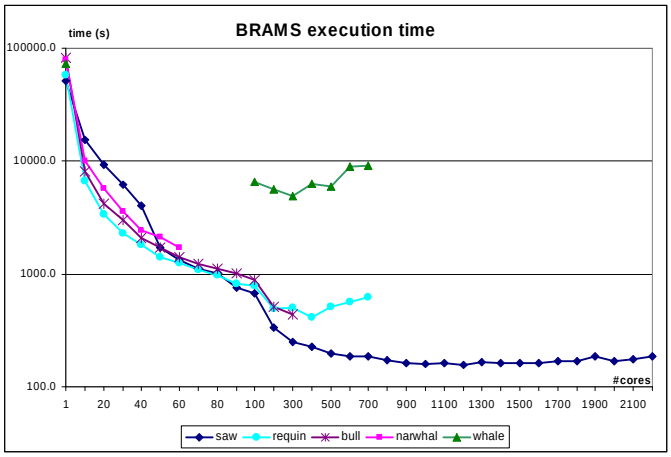

Figure 2. BRAMS model execution time 24h forecast.

BRAMS best execution time was 155.8s on "Saw" with 1200 cores, 410.0s on "Bull" with 376 cores, 416.0s on "Requin" with 400 cores and 1713.4s on "Narwhal" with 60 cores. "Bull" achieved the best BRAMS execution time using its total number of cores (376).

BRAMS execution was limited to 60 cores on "Narwhal" due to unknown problems. "Whale" was decommissioned during our experiment, so we only have results for the 100-700 cores range.

According to Eager [17], speedup and efficiency are the two performance metrics of particular interest when evaluating a parallel system. Speedup (1) is defined as the ratio of the elapsed time when executing a program on a single processor (Ts) to the execution time for n processors (Tp(n)):

$$\text{Speedup} = Ts/Tp(n) \qquad (1)$$

Efficiency (2) is a metric for the utilization of the n allocated processors. It provides information about how well the processors are utilized in executing a parallel application:

$$\text{Efficiency} = (Ts/(n*Tp(n)))*100\% \qquad (2)$$

Figure 3 shows BRAMS speedup on the systems. The order of the HPC clusters based on the best speedup and efficiency of BRAMS are as follows:

- "Bull", "Requin", "Narwhal" and "Saw" up to 60 cores;
- "Bull", "Requin" and "Saw" from 70 to 90 cores;

- "Bull", "Saw", "Requin", and "Whale" from 100 to 200 cores;
- "Saw", "Bull", "Requin", and "Whale" for more than 300 cores

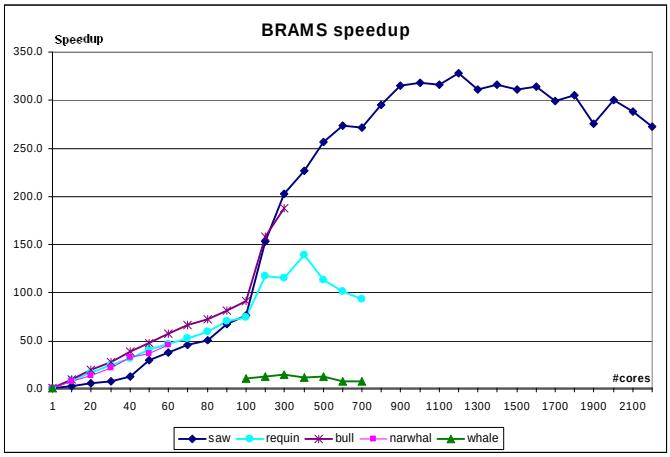

Figure 3. BRAMS model speedup for 24h forecast.

Some of the non-linearity in the execution times and speed-up of BRAMS observed in Figures 2 and 3 arise from multiple latency and bandwidth effects due to system interconnections. Some of these effects can be seen more clearly in the BRAMS efficiency graph (Figure 4).

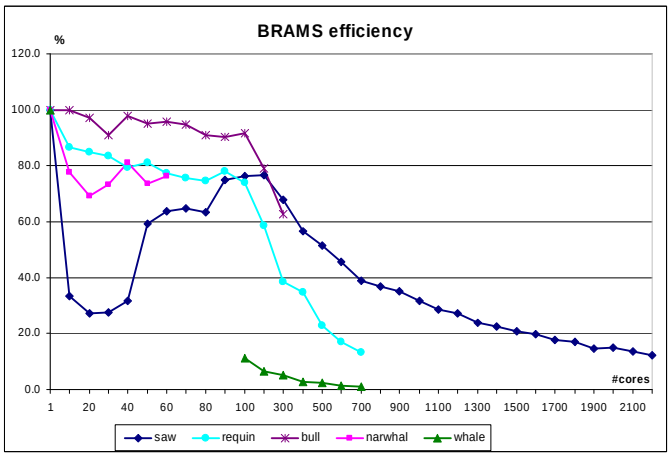

Figure 4. BRAMS model efficiency for 24h forecast.

Our results show that BRAMS performance is not only related to processor performance but that, as demonstrated by [5], switches plays an important role in HPC computing since their latency and throughput increase as packet size grows. Switches with low latencies tend to be more adequate for small message sizes and switches with high bandwidth are more adequate to big message size applications. In other words, applications that exchange small messages take advantage of low latency switches while applications that exchange big messages perform better on high bandwidth switches.

We have identified that the communication processes can become a bottleneck for the scalability of the BRAMS model. Network contention, specifically, is becoming an increasingly important factor affecting overall performance.

In order to gain a deeper understanding of cluster interconnection, we ran Single Transfer Benchmarks (STB) using Intel MPI Benchmarks (IMB) [18] to evaluate cluster MPI latency and bandwidth. It focuses on measuring startup and throughput of a single message transferred between two processes. We used Ping–Pong, where a single message is sent between two processes. Process 1 sends a message of size "x" to process 2 and process 2 sends "x" back to process 1.

Carrying this benchmark between the nearest nodes and farthest nodes of each HPC cluster helped us understand how interconnection affects BRAMS performance. The results from Ping-Pong benchmark revealed that communication with the furthest nodes had a higher latency and lower bandwidth than nodes that were closer. In the worst case, the latency in far nodes increased up to 92.5%, 22.4% and 22.4% in the furthest nodes, respectively for "Saw", "Requin" and "Narwhal". In addition, we observed more bandwidth and latency variation between the furthest nodes than in the closest nodes (Figure 5).

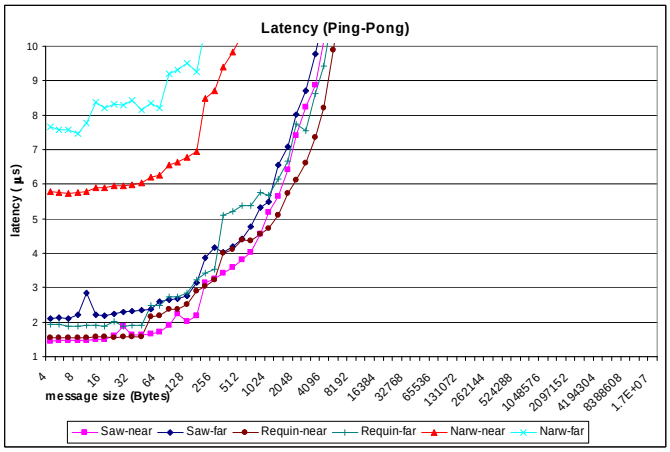

Figure 5. MPI Latency and bandwidth for cluster near and far nodes

We also observed a decrease in the effective bandwidth (Figure 6) to the furthest nodes by 92.6% on "Saw", 21.4% on "Requin" and 16.8% on "Narwhal". This was surprising, considering that "Saw" has the interconnection highest bandwidth of them, "Bull" and "Requin" presented a larger effective bandwidth than "Saw" for messages size up to 21 kB.

MPI implementations usually use an eager protocol for small messages and a rendezvous protocol for large messages. The rendezvous protocol needs a handshake between the sender and the receiver, thus requiring host intervention for MPI over InfiniBand and Myrinet. In other words, the rendezvous protocol limits their abilities for overlapping computation and

communication, so messages near critical size do not receive optimal performance. MPI over Quadrics is able to make communication progress asynchronously by taking advantages of the programmable network interface card. Thus it shows much better overlapping potential for large messages [19].

This effect is seen to a greater extent with "Saw" and to a lesser extent with "Narwhal". Figure 6 shows bandwidth decreases by 33% (794-533 MB/s) on "Saw" and by 17% (178-148 MB/s) on "Narwhal". This happens for message sizes between 13-21 kB for "Saw" and 16-43 kB for "Narwhal".

"Bull" presents the same latency and bandwidth values as "Requin" for the closest nodes, since it has the same interconnection technology and all nodes are directly connected to a single switch.

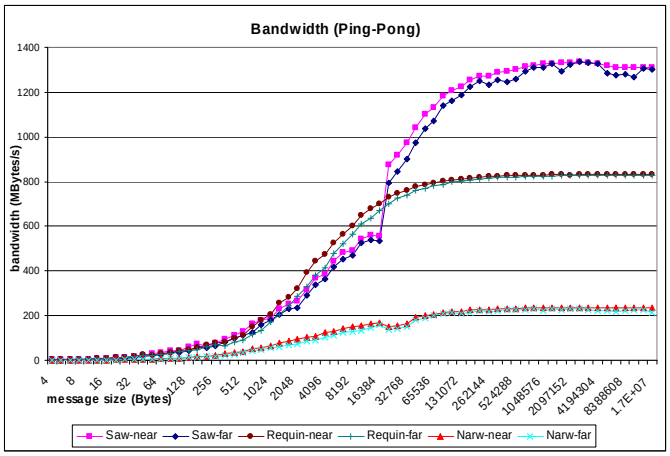

Figure 6. MPI bandwith for cluster near and far nodes

Despite the nominal switch latency and bandwidth presented in Table I, these values are message size dependent (Figure 5 and 6). So, the higher nominal bandwidth of "Saw" is not reflected in its performance, since this nominal bandwidth is related to a range of message sizes message sizes. In reality, latency and bandwidth vary with the size of message exchanged.

The bandwidth decreases by 30% for message sizes smaller than 28K on "Saw", 7kB on "Bull/Requin", and 1.7kB on "Narwhal". The latency increases by 30% for message sizes bigger than 84Bytes on "Saw", 52Bytes on "Bull/Requin", and 212 Bytes on "Narwhal".

We observe that the effective bandwidth for "Saw" is better than "Bull" for message sizes larger than 16KB and for "Requin" for message sizes larger than 21kB. In particular, the effective bandwidth for "Saw" takes a substantial increase for message sizes greater than these. For smaller message sizes, "Bull" and "Requin" have better effective bandwidth and this leads to better execution time for BRAMS on "Bull" and "Requin" with 70 and 90 cores, respectively (Figure 2).

As can be seen in Figure 6, as the message size increases, so does the effective bandwidth for the systems, though it does so in a non-linear manner. This partially explains the non-linearity of BRAMS performance as the number of cores used grows (Figure 2).

We infer that BRAMS execution time on "Bull", "Requin" and "Narwhal" is better than on "Saw", for a small number of cores, mainly because:

- "Saw" has a three layer topology for interconnection, with 2:1 oversubscription, which limits message exchange between nodes;
- The sharp decrease in "Saw" bandwidth for message sizes smaller than 21 kB.

The "Whale" cluster presented the worst performance for BRAMS, mainly because of high latency and low latency of GigaBit Ethernet interconnection.

"Bull" (direct connected topology) has better speedup and efficiency than "Requin" (fat tree topology), though both have the same interconnection (QsNet2/Elan4), because of the interconnection topology. Sometimes on "Requin", jobs are submitted to nodes connected to the same switch, providing similar performance to that of "Bull", but at other times jobs are submitted to nodes connected to different switches, increasing the execution time and decreasing BRAMS performance. This happen due to increased latency and lower bandwidth on nodes not connected to the same switch.

BRAMS performance is better in a direct connect topology than in fat tree topology. When using a small number of cores we observe a variation in BRAMS execution due to the effects of the latency introduced by the fat tree connectivity. Despite being the less expensive way to interconnect clusters, it can be difficult to get application performance when compared with direct connect topology [19].

The job submission system allocates cluster nodes according to its scheduling policy and does not consider the interconnection topology. As a result, a job that requires a number of switch ports that match a single switch may require more than one switch in a fat tree topology. For example, we noticed that even with the same interconnection technology, the performance variation is greater in "Requin" than in "Bull". The fat tree topology of "Requin" requires that a message pass in a certain number of hops for a communication between two cores.

Following the suggestion of Rodrigues et al [6], in this case a process mapping approach could be utilized in order to optimize the overheads between process communications, especially if a fat-tree topology is used. The algorithm used in that paper could be easily adapted to consider a tree structure

representing the different connections linking cores, processor, nodes and switches.

According to [20], choosing a network topology really depends upon the performance you desire, the price you are willing to pay, and perhaps secondarily, the simplicity of the topology and the ability to upgrade the system. In addition, he states that to save costs, typically links are oversubscribed and hence, in practice, we do not see “true" fat tree networks

5 Conclusions

In this paper, we have presented an analysis of BRAMS model performance and scalability over different HPC clusters architectures and configurations of SHARCNET.

As seen from the results obtained from this experiment, even with application code improvement, performance and scalability depends on cluster interconnection technology and topology.

HPC clusters using Infiniband presented the best performance and scalability results for BRAMS, followed by clusters with QsNet, Myrinet and Gigabit Ethernet. However, this order changes with the number of cores involved in BRAMS computation. Clusters with QsNet/Elan4 and Myrinet/2G present better performance for a small number of cores.

We identified how the eager and rendezvous protocols in MPI implementations interfere with interconnection bandwidth performance, especially on HPC clusters with Infiniband, but also with Myrinet, thus affecting application performance.

The results also present the benefits of a direct connect topology over fat tree topology. Even though being a popular topology, a fat interconnection tree represents a challenge in achieving application performance and scalability. Since switch port selection is not part of ordinary job submission system policies, the best performance of an application may not be achieved in a fat tree topology.

6 Acknowledgment

This work was made possible by FAPESP (process: 2010/05823-7) financial support and the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca) and Compute/Calcul Canada. We would like to thank Baolai Ge, from SHARCNET, and Luiz Flávio Rodrigues, from INPE/CPTEC, for their support. The author Alvaro L. Fazenda was partially supported by FAPESP (São Paulo Research Foundation) and CNPq-Brazil (National Council for Scientific and Technological Development).

7 References

[1] P. J. Roebber, D. M. Schultz, B. A. Colle, and D. J. Stensrud, “Toward improved prediction: High-resolution and ensemble modeling systems in operations”, Weather and Forecasting, 19, 936–949, 2004. doi: 10.1175/1520-0434(2004)019<0936:TIPHAE>2.0.CO;2

[2] G. Cats, “24 More Years of Numerical Weather Prediction: A Model Performance model”, KNMI Scientific report WR 2008-1, 2008.

[3] A. L. Fazenda, J. Panetta, P. Navaux, L. F. Rodrigues, D. M. Katsurayama, and L. F. Motta, “Challenges and solutions to improve the scalability of an operational regional meteorological forecasting model”, Paper accepted to apper in Int. J. High Performance Systems Architecture, 2011.

[4] G. Shainer, T. Liu, J. Michalakes and J. Liberman, “Weather Research and Forecast (WRF) Model Performance and Profiling Analysis on Advanced Multi-core HPC Clusters”, The 10th LCI InternationalConference on High-Performance Clustered Computing. Boulder, CO, 2009.

[5] B. Huang, M. Bauer and M. Katchabaw, “Hpcbench – a Linux-based network benchmark for high performance networks”, 19th International Symposium on High Performance Computing Systems ans Applications (HPCS’05). 2005.

[6] E. R. Rodrigues, F. L. Madruga, P.O.A. Navaux, J. Panetta, “Multi-core Aware Process Mapping and its Impact on Communication Overhead of Parallel Applications”, Proceedings of the 14th IEEE Symposium on Computers and Communications (ISCC 2009), July 2009, doi:10.1109/ISCC.2009.5202271.

[7] “Top 500 Supercomputer Sites”, March 2011. [Online]. Available: http://www.top500.org/.

[8] C. E. Leiserson, “Fat-trees: Universal Networks for Hardware-Efficient Supercomputing, IEEE Transactions on Computers, 34(10), October 1985.

[9] “SHARCNET homepage”, March 2011. [Online]. Available: http://www.sharcnet.ca.

[10] C. S. Yeo, R. Buyya, H. Pourreza, R. Eskicioglu, P. Graham and F. Sommers, “Cluster Computing: High-Performance, High-Availability, and High-Throughput Processing on a Network of Computers”, Handbook of Innovative Computing, Albert Zomaya (editor), Springer Verlag, 2005.

[11] M. A. Bauer, “High performance computing: the software challenges”, PASCO ’07: Proceedings of the 2007

international workshop on Parallel symbolic computation. New York, NY, USA: ACM, 2007, pp. 11–12.

[12] “Brazilian Regional Atmospheric Modeling System (BRAMS)”, March 2011. [Online]. Available in http://www.cptec.inpe.br/brams.

[13] R. Pielke, W. Cotton, R. Walko, C. Tremback, W. Lyons, L. Grasso, M. Nicholls, M. Moran, D. Wesley, T. Lee, and J. Copeland, “A comprehensive meteorological modeling system – RAMS”, Meteorology and Atmospheric Physics, 49(1-4):69–91, 1992.

[14] A. L. Fazenda, D. S. Moreira, E. H. Enari, J. Panetta and L. F. Rodrigues, “First time user's guide (BRAMS version 3.2)”, Cachoeira Paulista. 24p, 2006.

[15] C. L. Castro, R. A. Pielke and G. Leoncini, “Dynamical downscaling: Assessment of value retained and added suing the regional atmospheric modeling system (RAMS)”, J. Geophys. Res., 110 , D05108, doi:10.1029/2004JD004721.

[16] J. P. Bonatti, “Modelo de circulação geral atmosférico do CPTEC”, Climanálise Especial (10 anos Edição Especial), 5 pp., 1996, Centro de Previsão de Tempo e Estudos Climáticos (CPTEC), Cachoeira Paulista, Brazil. March 2011. [Online]. Available: www6.cptec.inpe.br/products/climanalise/cliesp10a/bonatti.html.

[17] D. L. Eager, J. Zahorjan, E.D. Lazowska, "Speedup versus efficiency in parallel systems", Computers, IEEE Transactions on , vol.38, no.3, pp.408-423, Mar 1989 doi: 10.1109/12.21127

[18] “Intel MPI Benchmarks 3.2.2”, March 2011 [Online]. Available: http://software.intel.com/en-us/articles/intel-mpi-benchmarks/

[19] J. Liu, B. Chandrasekaran, J. Wu, W. Jiang, S. Kini, W. Yu, D. Buntinas, P. Wyckoff, D.K. Panda, "Performance Comparison of MPI Implementations over InfiniBand, Myrinet and Quadrics", Supercomputing, 2003 ACM/IEEE Conference , p. 58, Nov. 2003, doi: 10.1109/SC.2003.10007.

[20] A. Bhatele, “Automating Topology Aware Mapping for Supercomputers”, PhD Thesis, Department of Computer Science, University of Illinois, 2010, http://hdl.handle.net/2142/16578.