Parallel Hierarchical Architectures

Martin Schmollinger

January 21, 2002

Overview

1. **Architectural Overview and Motivation**
   Architecture Promoted by the US-ASCI-project
   PC-based Clusters (dual PCs), NUMA-Clusters, SCOMA and ccNUMA
   Top500: Percentage of Constellations and SMP-Cluster over the last years
   Trends in the Future
2. **Models and Methodologies for Clusters of Multiprocessors**
   SIMPLE
   KeLP2
   (Hierachical) heterogeneous models
3. **Programming of Clusters of Multiprocessors**
   Message-Passing Programming (MPI)
   Distributed-Shared-Memory Programming (OpenMP)
   Mixed-Mode Programming (MPI+OpenMP)
4. **Algorithms, Libraries and Applications**

Abstract

In recent years, clusters of multiprocessor machines became one of the most important parallel architectures. The architecture was promoted by the ASCI (Accelerated Strategic Computing Initiative) project [1]. There are economic, as well as performance reasons for this promotion. The building blocks of such supercomputers are multiprocessor machines (nodes). The nodes consist of a number of processors (2-64) which have a shared memory. The nodes are standard machines or consist of standard components and therefore they can be produced cheaply. They are connected by a fast interconnection network. Depending on the network topology, there are at least two levels of parallel hierarchy. Processors in the same node can communicate very quickly using the shared memory, while the communication between processors of different nodes has to be done over the slower network. The architecture is very scalable. It is possible to add more nodes or even different types of nodes step by

step. The architecture permits to build scalable systems with a good price performance ratio. And they also can compete with other systems concerning peak performance. Looking at the actual Top500 list [35], more than 30% of the fastest supercomputers in the world belong to this kind of architecture. The following is a survey on endeavours to exploit the hybrid architecture in order to receive the peak performance of the hardware. The survey is structured as follows. The first section gives a more detailed overview of the considered architectures and their position in the world of supercomputing. In the second section models and methodologies to develop efficient algorithms on the target platform are presented. Section three shows which languages and libraries are used in practice to programm the systems. The last section deals with algorithms, libraries and applications which are especially build and optimized for clusters of multiprocessors.

1 Architectural Overview and Motivation

In recent decades, the development of parallel architectures was dominated by two types. The first type was the distributed-memory (dm) machine. Each processor has ist own memory and is connected to the other processors by some kind of network. The second type is called shared memory (sm) machine. The processors are connected by their common memory. The processors can communicate using the shared memory. In the first type of architecture, there are two possiblities for interprocessor communication. First, the processors can communicate by sending messages over the network to each other (message passing, mp). Second, there is a software layer which simulates a common address space. This case is called a distributes shared memory (dsm). In the end of the 1990s a new parallel architectures was promoted by the US-ASCI (United States Accelerated Strategic Computing Initiative) project [1]. The aim was to combine the two mentioned types above, in order to create new, more efficient, less expensive, more scalable parallel computers. The idea was to take standard sm-machines as building blocks (nodes) and to connect them with a fast network. Hence, the resulting architecture has at least two levels of hierarchy. Processors within one node can communicate very fast over their shared memory while processors of different nodes have to communicate through the slower network. The situation may even get more complex, if the network does not guarantee that the access times between all nodes is equal. The number of systems of that type increased enourmously in the last 5 years. Looking at the list of the fastest fivehundred supercomputers in the world (Top500 [35]), we can see that 37,6% (Constellation and Clusters) belong to that kind of architecture. And what is also important, there are 3 systems under the top ten. That shows that these systems can also compete concerning peak performance. At the moment, the worlds biggest project of building a supercomputer, the *Earthsimulator* [34], belongs to that kind of architecture. It will consist of 640 nodes where each has 8 processors. The total amout of memory will be 10 Tera bytes and the total peek performance will be about 40 Tera FLOPS.

to be continued by:

- Further description of sub-archtitectures: SCOMA, ccNUMA, PC-Based-Clusters, NUMA-Clusters
- Perhaps more detailed data out of the top500 statistics
- Explicit Listing of all aims and advantages of the architecture

Literature:
[21], [35], [7], [6], [14], [31], [1]

2 Models and Methodologies for Clusters of Multiprocessors

In sequential computing all algorithms are formulated using the von-Neumann model or the RAM-model. Also the programming models are based on them. In parallel computing, there is no such general purpose computational model, which can be used as a bridge between theory an practice. A lot of models have been proposed to close the gap, but no one has really reached this aim. Roughly speaking, in the history of modelling parallel computers there are two main streams which are still alive. The first stream of models can be called fine-grained models and the second one course-grained. The difference between the two approaches is mainly

- in the amount of data per processor.
- in the communication and its costs.

(Here i have to be more precise!)

While the fine-grained models correspond to sm-machines, the course-grained models belong to the distributed memory machines. Well-known representatives of the corse-grained models are BSP [30] and LogP [13]. A fine-grained model is for example the QSM model [16] which is, in the end, derived from the PRAM model [15]. As mentioned above, the new architecture is different from the sm- and dm-machines. Hence, it seems to be obvious, that there has to be some work in order to receive a model or a methododlogy with which it is possible to develop efficient algorithms for the new architecture.

At this point there will be the decription of the following models and methodologies:

- SIMPLE [4],
- KeLP2 [2]

As mentioned in the abstract, the nodes of the cluster can be of different types. In recent years, some models for heterogenous clusters have been developed. Here are some related papers. [33], [32], HiHCoHP [8]. But at the moment i am not sure, if it is really necessary to consider the heterogenous models. Perhaps this would enlarge the work too much.

3 Programming of Clusters of Multiprocessors

Looking at parallel programming paradigms, we can expose two extrems. First, there is the message passing paradigm. Each processor has its local memory and communicates with other processors by exchanging messages. This paradigm represents a "shared nothing" approach. Second, there is the shared-memory programming paradigm. Here all the data can be accessed by all processors. The programmer has to formulate race-conditions to avoid deadlocks or inconsistencies. This is the "shared-everything" approach. Looking at clusters of multiprocessors, it is possible to use both paradigms. But it is obvious that without optimization it is not possible to receive peak performance, because both do not consider the existing parallel hierarchy. Despite of that, there are developments which try to adapt standard message passing libraries and distributed shared-memory systems to clusters of multiprocessors. Recent development in this area was made in order to receive a mixed-mode programming model. The idea behind that, is to use shared-memory programming inside the nodes and to use message passing between the nodes. This is a very promising approach, because it fits to the architecture. In the following there are three subsections about programming of clusters of multiprocessors. The first subsection is about using message-passing as programming paradigm. It will report which optimizations were made to the standard message-passing library MPI [24][25], in order to improve performance on clusters of multiprocessors. Subsection two is about shared-memory programming. The standard for programming shared-memory systems is openMP [26]. In the subsection there is a summary on how openMP is adapted to clusters of multiprocessors. The third subsection is about mixed-mode programming. Most work describes the usage of MPI and openMP and compares the resulting performance with the performance ordinary MPI-programs.

3.1 Message-Passing Programming (Focus: MPI)

Literature:
[29]

3.2 Distributed Shared-Memory Programming (Focus: OpenMP)

Literature:
[28], [20], [23], Distributed Cilk [22] KeLP2 [2]

3.3 Mixed-Mode Programming (Focus: OpenMP-MPI)

Literature:
[19], [9], [11], [10], [17], [12], SIMPLE [4]

- Some slides to introduce mixed-mode programming:
  http://www.research.ibm.com/actc/Talks/DavidKlepacki/MixedMode/index.htm

4 Algorithm, Libraries and Applications

What we learned up to now is how clusters of multiprocessors can be best exploited. But now we what to have a look at the output of this efforts. This section is a summary of algorithms designed for, libraries build for and applications developed on top of clusters of multiprocessors.
Literature:
[3], [5], [27], [18]

5 Conclusions

The following questions should be answered:

1. Which computer projects are in progress?
2. Are the proposed models and methodologies sufficient to develop efficient algorithms?
3. Is there a potential in developing completely new algorithms or is it sufficient to provide methodologies to migrate standard algorithms (bsp-algorithms) to clusters of multiprocessors?
4. Is it necessary to design a new programming paradigm or are the common paradigms sufficient (perhaps with certain optimizations)?
5. Where are the problems of mixed-mode programming?
6. Will there be enough algorithms and libraries to use the clusters efficiently in the future? Which projects are in progress.
7. Where are lacks of suitable solutions?

References

[1] Accelerated strategic computing initiative (ASCI). Technical Report UCRL-MI-125923, Lawrence Livermore National Laboratory, 1998.

[2] S. B. Baden and S. J. Fink. A programming methodology for dual-tier multicomputers. *IEEE Transactions on Software Engineering*, 1998.

[3] D. Bader. High-performance algorithms and applications for SMP clusters. In *NASA High Performance Computing and Communications Computational Aerosciences Workshop (CAS 2000)*, NASA Ames Research Center, Moffett Field, CA, February 2000. http://hpc.eece.unm.edu/papers/NASA_CAS2000.ps.

[4] D. Bader and J. JáJá. SIMPLE: A methodology for programming high performance algorithms on clusters of symmetric multiprocessors (SMPs). *Journal of Parallel and Distributed Computing*, 58(1):92–108, 1999. http://hpc.eece.unm.edu/papers/3798.ps.

[5] D. Bader and R. Pennington. Cluster computing: Applications. *The International Journal of High Performance Computing*, 15(2):181–185, May 2001. http://hpc.eece.unm.edu/papers/ijhpca.ps.

[6] M. Baker. Cluster computing white paper, December 2000. http://www.dcs.port.ac.uk/ mab/tfcc/WhitePaper.

[7] G. Bell and J. Gray. High performance computing: Crays, clusters, and centers. what next? Technical Report MSR-TR-2001-76, Microsoft Research, Microsoft Corporation, August 2001.

[8] F. Capello, P. Fraigniaud, B. Mans, and A. L. Rosenberg. HiHCoHP – towards a realistic communication model for hierarchical hyperclusters of heterogeneous processors. In *International Parallel and Distributed Processing Symposium*, 2001.

[9] F. Cappello and D. Etiemble. MPI versus MPI+openmp on IBM SP for the NAS benchmarks. In *Supercomputing*, 2000.

[10] F. Cappello and O. Richard. Performance characteristics of a network of commodity multiprocessors for the NAS benchmarks using a hybrid memory model. In *IFIP PACT 99*, October 1999. http://www.lri.fr/ gk/QUID.

[11] F. Cappello, O. Richard, and D. Etiemble. Investigating the performance of two programming models for clusters of SMP PCs. In *IEEE HPCA6*, January 2000. http://www.lri.fr/ gk/QUID.

[12] E. Chow and D. Hysom. Assessing performance of hybrid MPI/openMP programs on SMP clusters. Technical Report UCRL-JC-143957, Lawrence Livermore National Laboratory, May 2001. http://www.llnl.gov/CASC/people/chow/chow_pubs.html.

[13] D. Culler, R. Karp, D. Patterson, A. Sahay, K. Schauser, E. Santos, R. Subramonian, and T. von Eicken. LogP: Towards a realistic model of parallel computation. In *Proceedings of the 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 1–12, Mai 1993.

[14] J. Dongarra, H. Meuer, H. Simon, and E. Strohmaier. Biannual top-500 computer lists track changing environments for scientific computing. *SIAM News*, 34(9), November 2001.

[15] S. Fortune and J. Wyllie. Parallelism in random access machines. In *Proceedings of the 10th ACM Symposium on Theorie of Computing*, pages 114–118, 1978.

[16] P. B. Gibbons, Y. Matias, and V. Ramachandran. Can shared-memory model serve as a bridging model for parallel computation? In *ACM Symposium on Parallel Algorithms and Architectures*, pages 72–83, 1997.

[17] L. Giraud. Combining shared and distributed memory programming models on clusters of symmetric multiprocessors: Some basic promising experiments. http://www.cerfacs.fr/algor/reports/2001/WN_PA_01_19.ps.gz, 2001.

[18] D. R. Helman and J. JáJá. Sorting on clusters of SMPs. *Informatica: An International Journal of Computing and Informatics*, 23, 1999.

[19] D. S. Henty. Performance of hybrid message-passing and shared-memory parallelism for discrete element modeling. In *Supercomputing 2000*, Edinburgh Parallel Computing Centre, 2000. http://www.hpc.gatech.edu/SC2000/techpapers/papers/PAP154.PDF.

[20] Y. C. Hu, H. Lu, A. Cox, and W. Zwaenepoel. OpenMP for networks of SMPs. In *Proc. of the Second Merged Symp. IPPS/SPDP 1999)*, 1999.

[21] W. E. Johnston. Rationale and strategy for a 21st century scientific computing architecture: The case for using commercial symmetric multiprocessors as supercomputers. *International Journal of High Speed Computing*, June 1998. http://www-itg.lbl.gov/ johnston/papers.html.

[22] L.Peng, W. Wong, M. Feng, and C. Yuen. Silkroad: A multithreaded runtime system with software distributed shared memory for SMP cluster. In *IEEE International Conference on Cluster Computing (CLUSTER 2000)*, pages 243–249, December 2000. http://www.comp.nus.edu.sg/ wongwf/publications.html.

[23] J. Merlin, D. Miles, and V. Schuster. The portland group distributed OMP: Extensions to openMP for SMP-cluster. EWOMP 2000, September 2000. http://www.epcc.ed.ac.uk/ewomp2000/Presentations/MerlinSlides.pdf.

[24] Message Passing Interface Forum. MPI: A message-passing interface standard. Technical Report CS-94-230, Computer Science Department, University of Tennessee, Knoxville, TN, May 1994.

[25] Message Passing Interface Forum. MPI-2: Extensions to the message-passing interface, July 1997.

[26] OpenMP Forum. OpenMP C and C++ application program interface, version 1.0. http://www.openmp.org, October 1998.

[27] A. Petitet, H. Casanova, R. Whaley, J. Dongarra, and Y. Robert. A numerical linear algebra problem solving environment designer's perspective. In *SIAM Annual Meeting*, Atlanta, GA, May 1999. http://www.netlib.org/utk/people/JackDongarra/PAPERS/libraries-tools1.pdf.

[28] M. Sato, S. Satoh, K. Kusano, and Y. Tanaka. Design of openmp compiler for an smp cluster, 1999.

[29] T. Takahashi, F. O'Carroll, H. Tezuka, A. Hori, S. Sumimoto, H. Harada, Y. Ishikawa, and P. H. Beckman. Implementation and evaluation of MPI on an SMP cluster. In *IPPS/SPDP Workshops*, pages 1178–1192, 1999.

[30] L. G. Valiant. A bridging model for parallel computation. *Communications of the ACM*, 33(8):103–111, 1990.

[31] A. J. van der Steen and J. Dongarra. Overview of recent supercomputers, November 2001.

[32] T. Williams and R. Parsons. The heterogeneous bulk synchronous parallel model. In *Workshop on Advances in Parallel and Distributed Computational Models, in conjunction with the 14th International Parallel and Distributed Processing Symposium*, 2000. http://ipdps.eece.unm.edu/2000/advances/18000102.pdf.

[33] T. Williams and R. Parsons. Exploiting hierarchy in heterogeneous environments. In *Workshop on Advances in Parallel and Distributed Computational Models, in conjunction with the 15th International Parallel and Distributed Processing Symposium*, May 2001.

[34] WWW. http://www.es.jamstec.go.jp/eng/menu.html, 2001.

[35] WWW. http://www.top500.org, 2001.