GeneCrunch and Europort, Examples of Hierarchical Supercomputing at SGI

Reinhard Schneider
EMBL Heidelberg , Meyerhofstr. 1, 69012 Heidelberg, Germany

Michael Schlenkrich
European Chemistry Technology Center
SGI , Kaegenstrasse 17, 4153 Reinach, Switzerland

Abstract . The SGI POWER CHALLENGEarray ™ represents a hierarchical supercomputer because it combines distributed and shared memory technology. We present two projects, Europort and GeneCrunch, that took advantage of such a configuration. In Europort we performed scalability demonstrations up to 64 processors with applications relevant to the chemical and pharmaceutical industries. GeneCrunch, a project in bioinformatics, performed an analysis of the whole yeast genome using the software system GeneQuiz. This project showcased the future demands of HPC in pharmaceutical industries in tackling analysis of fast growing volumes of sequence information. GeneQuiz, an automated software system for large-scale genome analysis developed at the EMBL /EBI , aims at predicting the function of new genes by using an automated, rigorous, rule-based system to process the results of sequence analysis and database searches to build databases of annotations and predictions. In GeneCrunch more than 6,000 proteins from baker's yeast, for which the complete genomic sequence was completed in 1996, were analyzed on a SGI® POWER CHALLENGEarray with 64 processors (R8000® at 90MHz) in three days rather than the seven months predicted for a normal workstation.

1. Hierarchical Parallel Supercomputing

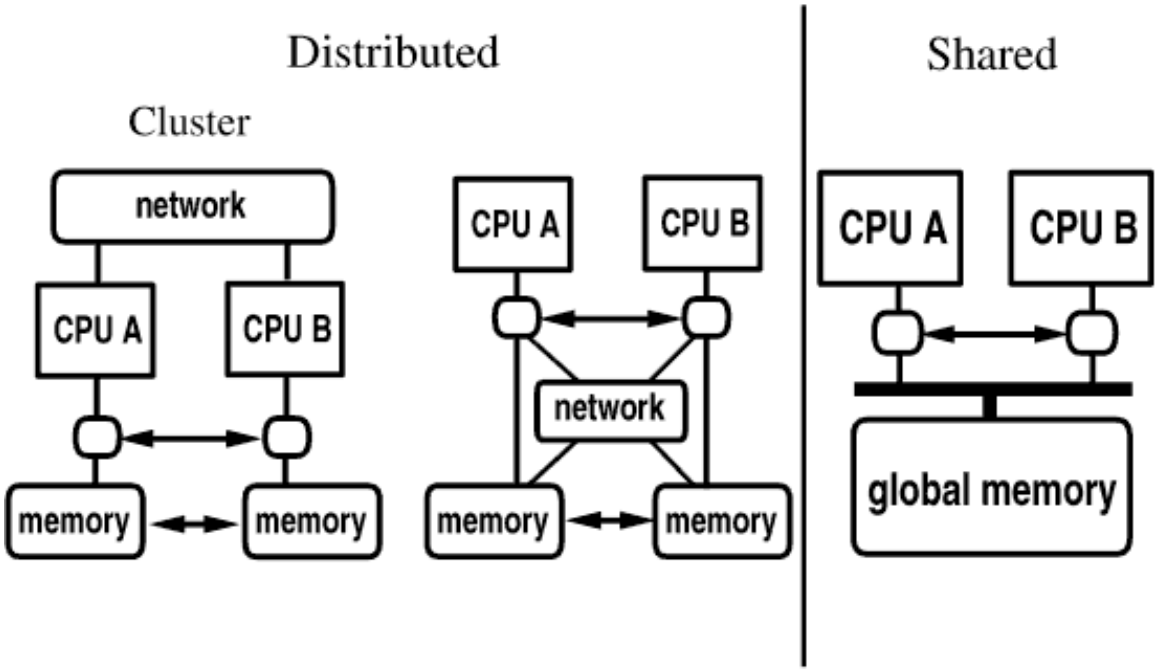


Figure 1: Schematic representation of distributed and shared memory systems. Arrows mark the memory system for which data coherency has to be guaranteed. The small unlabeled boxes represent the caches. On the SGI shared memory system the data coherency is achieved in hardware, while normally cluster solutions rely on a software.

The design of parallel supercomputers can be separated into two classes: the distributed memory and the shared memory system (Figure 1). The essential differentiators are the location and access methods of the memory and the data coherency model. In a distributed memory machine each process element (PE) has its own memory subsystem, coherency of the data is implemented in software, the shared memory concept provides a unified memory for all the processors, and data coherency is maintained in hardware.

The distributed memory systems can be subdivided into two classes in which the first class is the cluster of individual workstations. In this design, access to the memory of a neighboring PE has to involve the CPU of the PE, which results in the push-and-pull mechanism. This concept is very latency sensitive, since two processors have to be synchronized. Allowing high bandwidth requires a huge investment in the network capabilities of each individual PE, which normally prohibits a high bandwidth between the PE. Data coherency among the PEs has to be in software, which requires an explicit message passing within the parallel application. However, the great advantage of this design is the expandability, since there is no real

limit to the number of PEs.

The second class of distributed memory systems has a more sophisticated memory access network that bypasses the processor of the PE on which the memory access occurs. This greatly reduces the latency using a more complicated and therefore more expensive coupling of PE. Normally such designs do not include any cache-coherency protocols among the processors. Therefore, applications have to explicitly ensure data coherency.

The shared memory concept provides a global memory to all processors, and the coherency of the data in the caches is maintained in hardware. This enables the used of shared memory parallel programming, which does not require the explicit programming of the data flow among the processors. This concept does not rule out parallel programming using message-passing libraries, since these libraries can be optimized to make use of the shared memory concept to allow very low latency and high bandwidth among the individual threads. The downside of a shared memory architecture is the expandability, which is currently limited to 36 processors on the POWER CHALLENGE ™ product line.

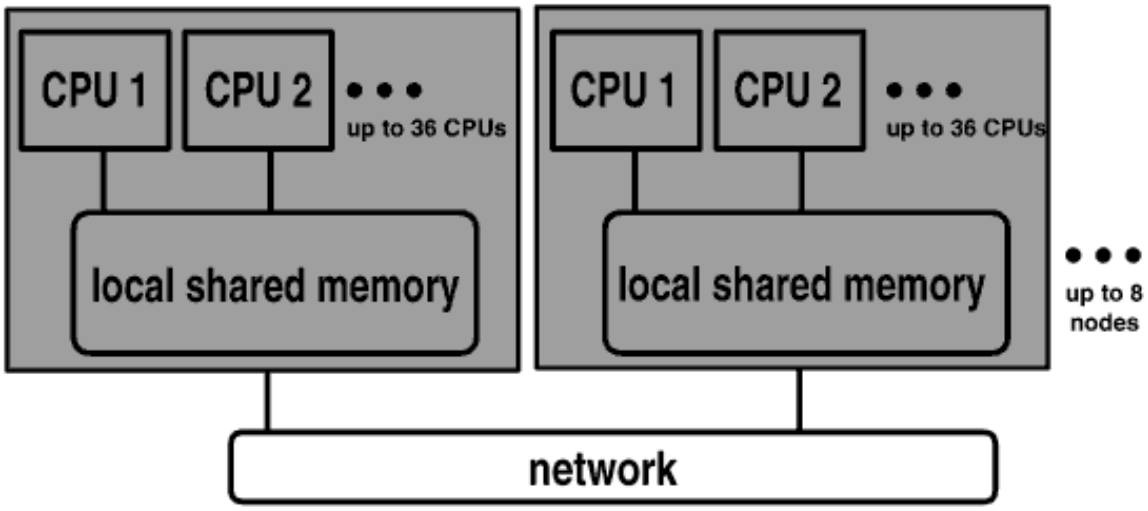


Figure 2: Schematic representation of the POWER CHALLENGEarray.

The POWER CHALLENGEarray combines both concepts to exploit the best of both worlds. Figure 2 shows a schematic representation of the POWER CHALLENGEarray with the individual shared memory nodes coupled with a network. Since the number of individual nodes is low, the relative cost of the network is small. This allows the usage of a high-performance network such as HIPPI. While in principle there is no limit to the number of shared memory nodes, a feasible configuration can have up to eight nodes, resulting in a maximum of 288 processors. The data

coherency is maintained in hardware within a node, while a distributed memory model holds among the nodes and software (message passing) is required to maintain data coherency.

There are two programming models that can be used on the POWER CHALLENGEarray,shared memory and message passing. Shared memory parallelization can use up to 36 processors within one shared memory node. Message-passing libraries have to be used if more than 36 processors are used. Currently two message-passing protocols, MPI and PVM, are supported on the POWER CHALLENGEarray. These libraries utilize the shared memory hardware if messages are exchanged among threads residing in one shared memory node. If messages are exchanged among threads residing on different nodes, PVM and MPI switch to a socket mechanism that is optimized for the HIPPI hardware. An interesting approach is to use both programming paradigms: having shared memory parallel threads running on the shared memory nodes talking to other threads on different nodes via message passing. Hereby we could exploit both the coarse and the fine grained parallelism in applications.

2. EUROPORT

Europort was a European initiative, the primary objective of which is to increase the awareness and confidence in parallel high-performance computing (HPC) platforms for commercial and industrial applications. Over 20 different consortia were formed to approach the individual industry sectors. Three of these consortia target applications in the computational chemistry and bioinformatics area: PACC, IMMP and MAXHOM. Daron Green's article in this proceeding focuses on the results of these activities.

Here we will describe the highlights of using the POWER CHALLENGEarray configuration at the SGI Supercomputing Center in Cortaillod, Switzerland. Figure 3 shows the setup of the system having 64 processors and a total of 8GB of main memory.

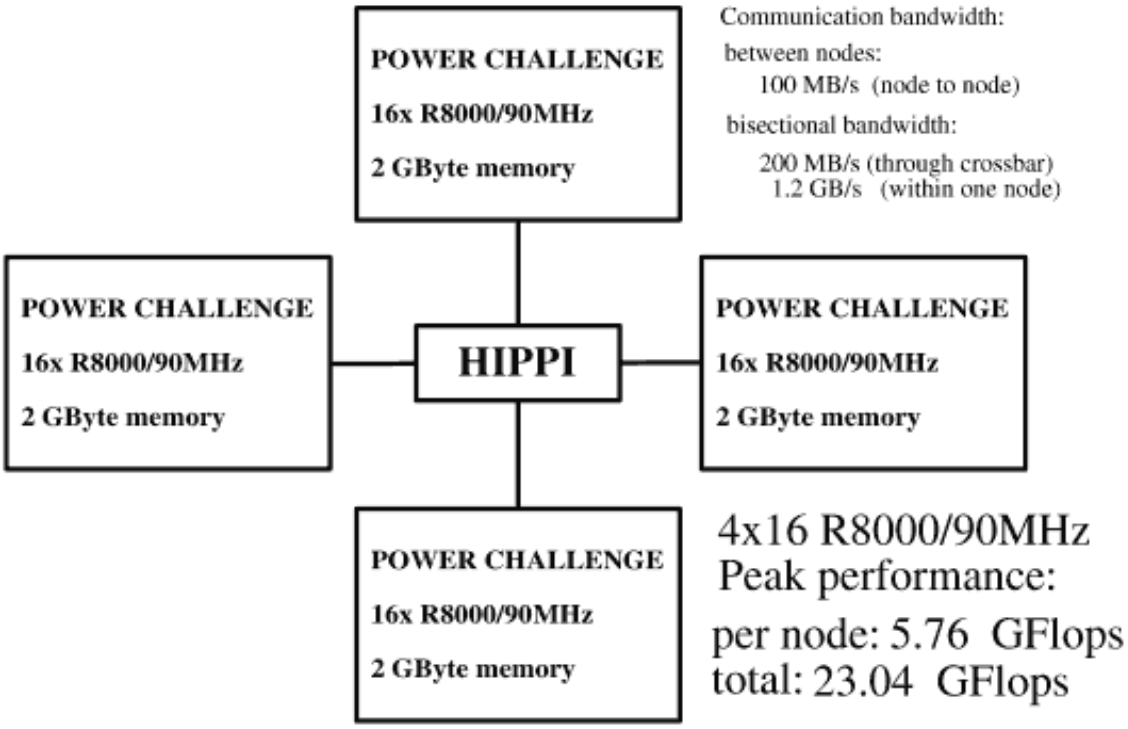


Figure 3: The POWER CHALLENGEarray setup of SGI Supercomputing Technology Center in Cortaillod, Switzerland.

Turbomole, an ab initio application, was parallelized in the PACC consortia [1]. The speed-up curve, up to 64 processors of a direct SCF calculation with more than 1,000 basis function, is shown in Figure 4. Runs with 16 or fewer processors were performed within one shared memory node, utilizing the fast implementation of the PVM message passing library. Runs with more than 16 processors were performed on multiple nodes. The scalability of the code does not change going from a single to a multiple node configuration. This shows that the HIPPI connection between the nodes can supply the required data latency and bandwidth.

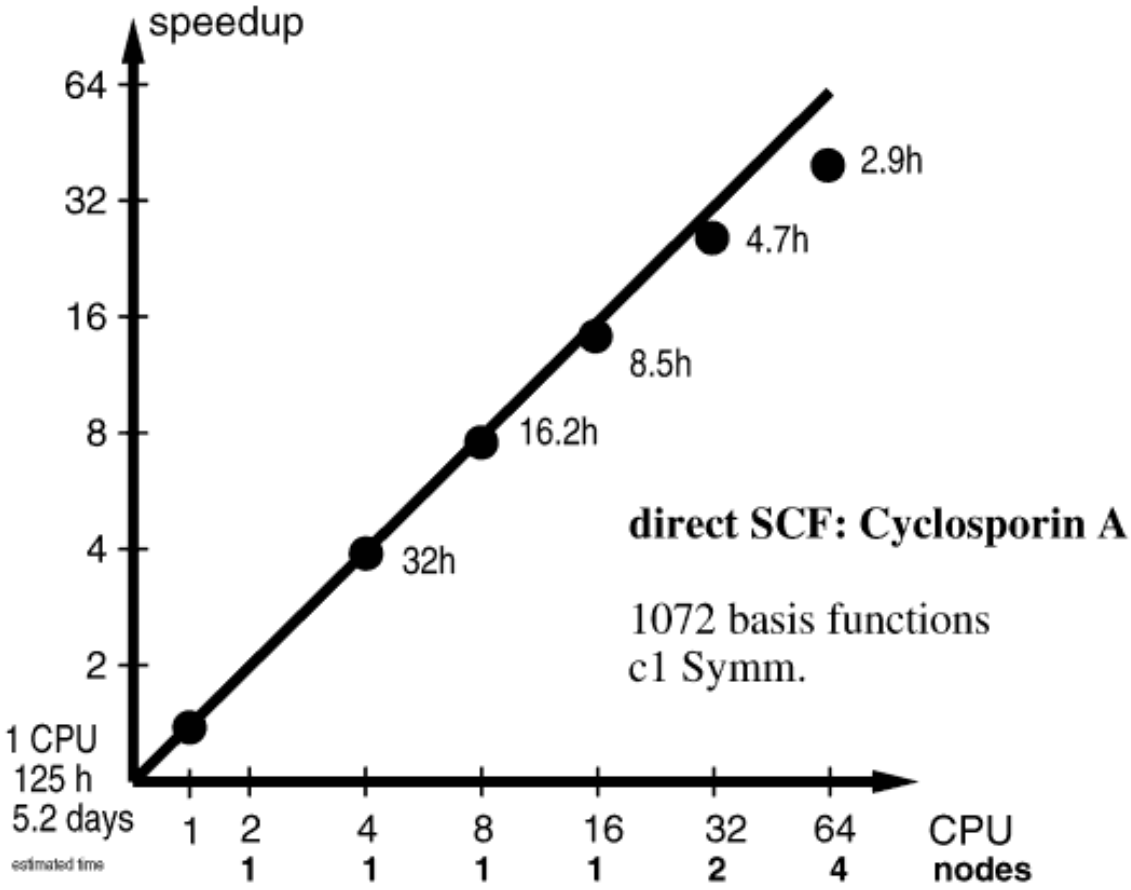
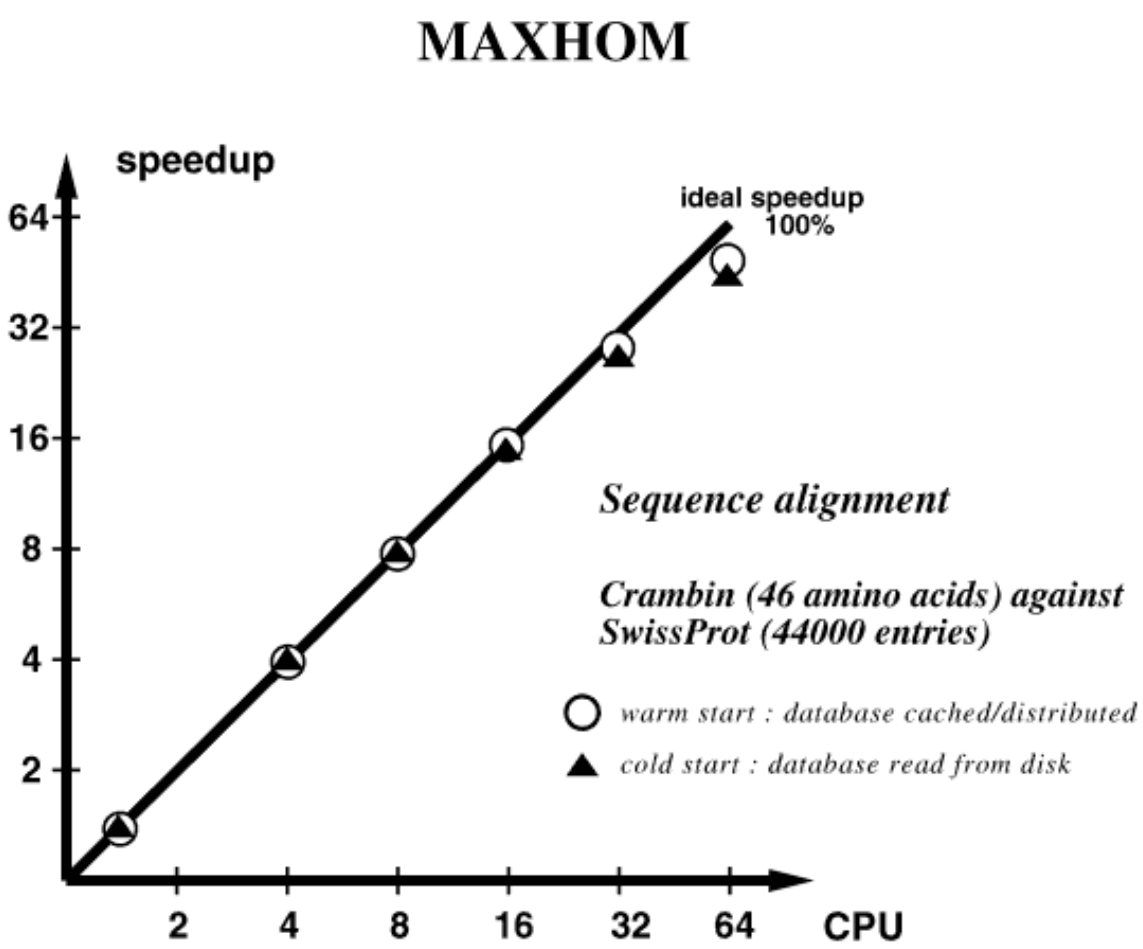


Figure 4

The primary computational task in ab initio computations is the generation of the Fock Matrix. The individual contributions to the Fock Matrix are purely additive; therefore, it is relatively easy to parallelize over these contributions and merge the results together. Since this part contributes to approximately 99 percent of the computation, the scalability is good. The second part is the diagonalization of the dense Fock matrix. Due to the size of the matrix and its dense form, the scalability is limited and requires a low-latency and high-bandwidth communication among the threads [2]. Here the benefits of the POWER CHALLENGEarray come into the game, since the processors within one node are tightly coupled, allowing a parallel implementation of the diagonalization on 8 CPUs.

Figure 5



MAXHOM [3], a second example of an ideally performing application on the POWER CHALLENGEarray, is shown in Figure 5. MAXHOM performs multiple sequence alignment of a target sequence against the SWISSPROT database. Crucial in this benchmark is that the database was located on a filesystem mounted to one of the shared memory nodes. Threads running on a different node accessed the database via NFS ™. A second experiment was to load the database in memory in an initialization phase and then perform the benchmark. As one can see from Figure 5, both curves are almost identical, demonstrating the speed of file access through the HIPPI network.

3. GeneCrunch

3.1. Motivation: Large-Scale Sequence Analysis and the Need for Automatic Tools

Since genome sequence data is being produced at an accelerating pace, there is a need for faster and more reliable methods of large-scale sequence analysis. There exists a multitude of algorithms, a large number

of sequence and bibliographic databases, and various single methods that can be useful in the prediction of protein function. From this large collection of tools, an optimal constellation that satisfies the requirements for accurate and sensitive function prediction by homology must be chosen. Speed is also an important factor for the analysis, but accuracy should not be sacrificed.

The technical challenges are twofold:
How to quickly identify sequence similarities in molecular databases efficiently without losing sensitivity
How to integrate existing software and databases and annotate, evaluate, and document the findings of experts in a multiuser interactive environment

Large-scale sequence analysis differs from traditional practices in two basic respects:
With the current and foreseen growth of the biological databases, computational efficiency using fast algorithms, certain heuristics, and supercomputers are essential
Information support for expert users is becoming crucial as the emerging gene and protein families from genome projects extend beyond the areas of expertise of a single individual

Therefore, the development of a system that performs the necessary analytical steps for a large number of sequences as well as provides access to molecular and bibliographic databases is required.

The most compelling question in computational genome analysis is the identification of homologies in search of a function. However, the issue of function prediction for proteins is partly a problem of definition. We can define function prediction as any evidence of the identification of various protein sequence characteristics indicative of substrate recognition and catalysis, interactions, localization, and evolutionary relationships. Therefore, the characterization of a protein sequence (or an ORF) usually takes place at various levels of accuracy, for example, from prediction of cell membrane spanning regions to the derivation of a three-dimensional model, on the basis of homology to a well-characterized protein.

3.2. The GeneQuiz system

GeneQuiz [4, 5] is an integrated system for large-scale biological sequence analysis that goes from a protein sequence to a biochemical function using a variety of search and analysis methods and up-to-date

protein and DNA databases. Applying an "expert system" module to the results of the different methods, GeneQuiz creates a compact summary of findings. It focuses on deriving a predicted protein function, based on the available evidence, including the evaluation of the similarity to the closest homologue sequences in the database. The analysis yields a great portion of the information that can possibly be extracted from the current databases, including three-dimensional models by homology, when the structure can be reliably calculated.

The principal design requirement is the complete automation of all repetitive actions: database updates, efficient sequence similarity searches, sampling of results in a uniform fashion, and evaluation and interpretation of the results using expert knowledge coded in rules (Figure 6). For handling such a heterogeneous set of tools and tasks in the GeneQuiz system we chose the *perl* script language [6].

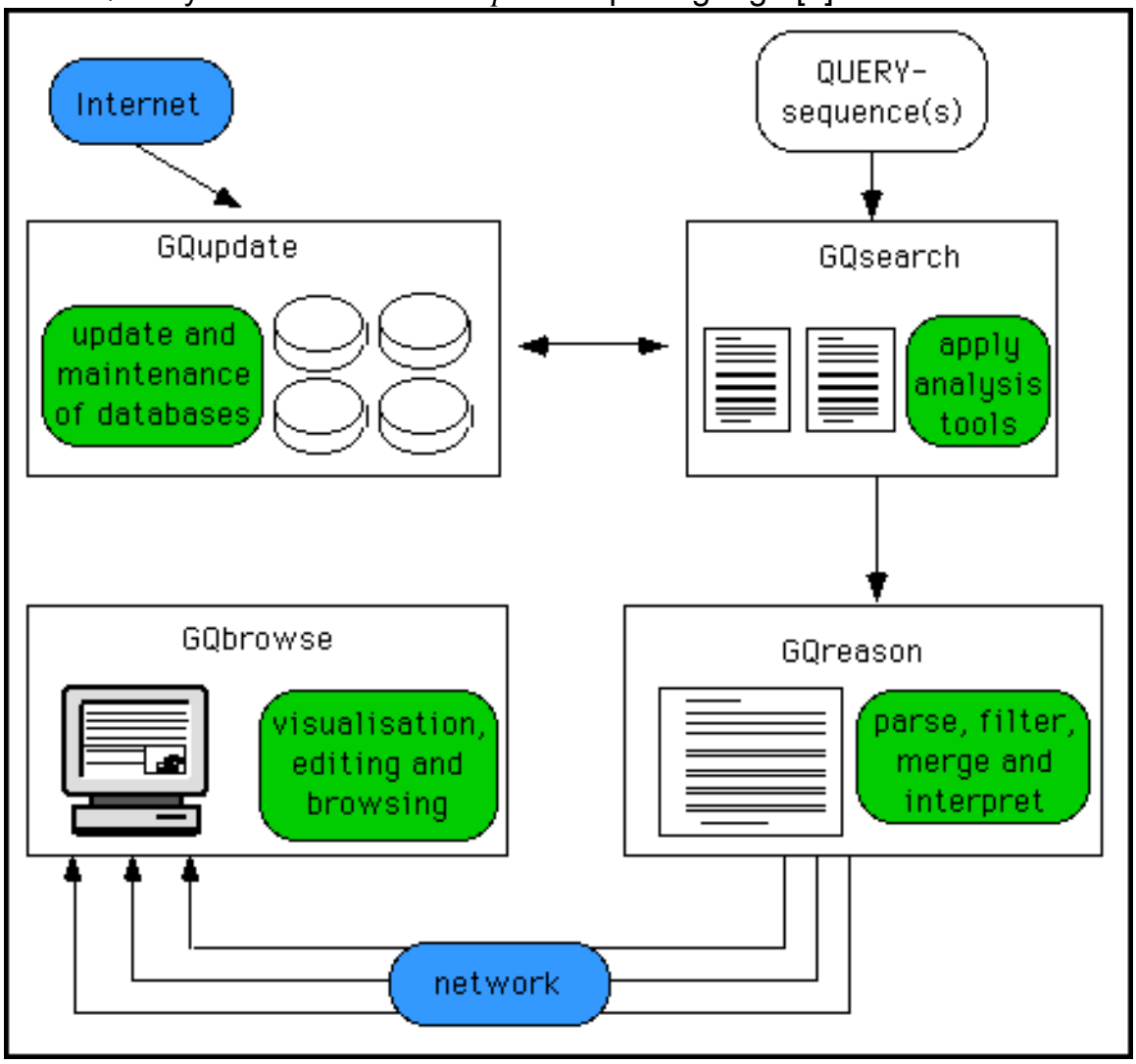


Figure 6: Schematic flow chart of the GeneQuiz system. Note: for the

analysis done during the GeneCrunch project the databases were frozen and no update was performed during the runs.

The compute-intensive part of GeneCrunch is the GQsearch part, which performs the automated database search and sequence analysis. To accelerate a first scanning of all databases in the most efficient way, a hierarchical model for database searches was implemented. First, searching with the fastest available tool, currently BLAST [7], allows the identification of clear homologous sequences from which a possible function can be inferred. The search is by default performed against a nonredundant database, which also includes the proteins translated from genes in the DNA databases.

Additional characteristics of newly sequenced ORFs are of interest, especially when function by homology cannot be predicted. For example, structural features, indication for the subcellular location, or previously described sequence patterns can be of extreme importance for a further understanding of protein function. The compute time for these analyses is negligible; therefore, they are performed for every input sequence. In addition to standard analyses, we use filters, pre- and post-processing tools, and additional methods for shorter and more meaningful output lists, multiple alignments, cluster analysis, and secondary structure prediction.

The GQsearch control program distributed the jobs to the individual nodes of the POWER CHALLENGEarray. Due to the fact that each single sequence analysis can be seen as an independent task, the problem becomes in principle parallel and allows for a straightforward load balancing scheme. The program contains a checkpointing facility to ensure the recovery of jobs after hardware and software problems.

The reasoning and function assignment based on the output of all the runs in GQsearch is processed by the GQreason tool. This tool builds a relational database (RDB) (developed by Walt Hobbs , RAND Corporation, Santa Monica, CA), structuring the results obtained. The generation of the RDB is not trivial, since the programs used in GeneQuiz provide a wide range of output formats, usually a compromise between machine and human readability. The lack of syntax and a standard has necessitated the implementation of a variety of dedicated parsers for the output. Parsers for all the database search programs and for most of the analysis tools have been implemented. In that respect, the system is independent of the search software.

Finally, in the last step, the extracted features are summarized at a higher

level into a comprehensive table of the results. At this level, rules are very strict, and we report only clear results. In this way, the user can trust the derived facts and is relieved from time-consuming interactive checking. Ambiguous assignments are marked as such and help the user to directly focus on those difficult cases that could not be automatically resolved with the given data.

The fourth module (GQbrowse) gives access to the result databases and allows for interactive evaluation and browsing of related sequences and other databases like bibliographic entries. The current solution is based on World Wide Web technology and dynamically provides HTML documents that can be displayed with most of the Web browsers (such as Netscape® or Mosaic). With this technology, it is straightforward to make the results-and the whole browsing capacity-available to any user connected to the Internet.

If the a homologous 3D structure is available in the database, an automatic model building procedure in the WHATIF program [8] is used to construct a 3D model of the protein. Using the VRML [9,10] 3D description, a Web browser can actually display the 3D structure of the molecules.

3.3. GeneCrunch Setup

The yeast genome sequence was completed and released in 1996 as a result of an international collaboration with teams from the European Union, North America, and Japan. The genome consists of about 12.5 million basepairs on 16 chromosomes. Its content is estimated to be about 6,300 open reading frames (ORFs). ORFs are regions of the genome identified as potentially coding for proteins. Complete analysis of the yeast genome represents a milestone in genomic research. It is the first eucaryotic genome fully sequenced and contains the complex subcellular organization typical of higher organisms.

While many parts of the yeast genome have been previously analyzed over the history of the sequencing effort, a complete reanalysis using the most up-to-date versions of the databases and the newest search and analysis methods was necessary. For the GeneCrunch project we extracted more than 6,000 yeast ORF sequences from the publicly available databases. The analysis of these ORFs represents a large computational effort that would require months to complete on standard workstations or servers.

To redress this computational bottleneck, the analysis was performed on a

SGI POWER CHALLENGEarray providing 23.04 GFLOPS of compute power on 4 POWER CHALLENGE nodes with 16 R8000 90MHz CPUs and 2GB RAM each. POWER CHALLENGE uses the concept of shared resource parallelism, which permits applications with different memory, I/O, compute, and visualization needs to run while the hardware and operating system ensure data consistency among the parallel threads and dynamically allocate the resources among the different programs. This concept makes it very easy to run the GeneQuiz system, in which applications vary greatly from each other in computational resource needs.

A HIPPI Internet providing 100MB/sec data transfer speeds with a sustained bisection bandwidth of 200Mb/sec connected the 4 nodes. The sustained bandwidth within each node is 1.2GB/sec. A separate CHALLENGE® acted as a file server for the array, with 20GB of data stored in one XFS® volume covering five 4GB physical disks. One node of the array also served as the repository for 13GB of data generated during the computation and provided internal Web service on the SGI intranet.(See Figure 7.)

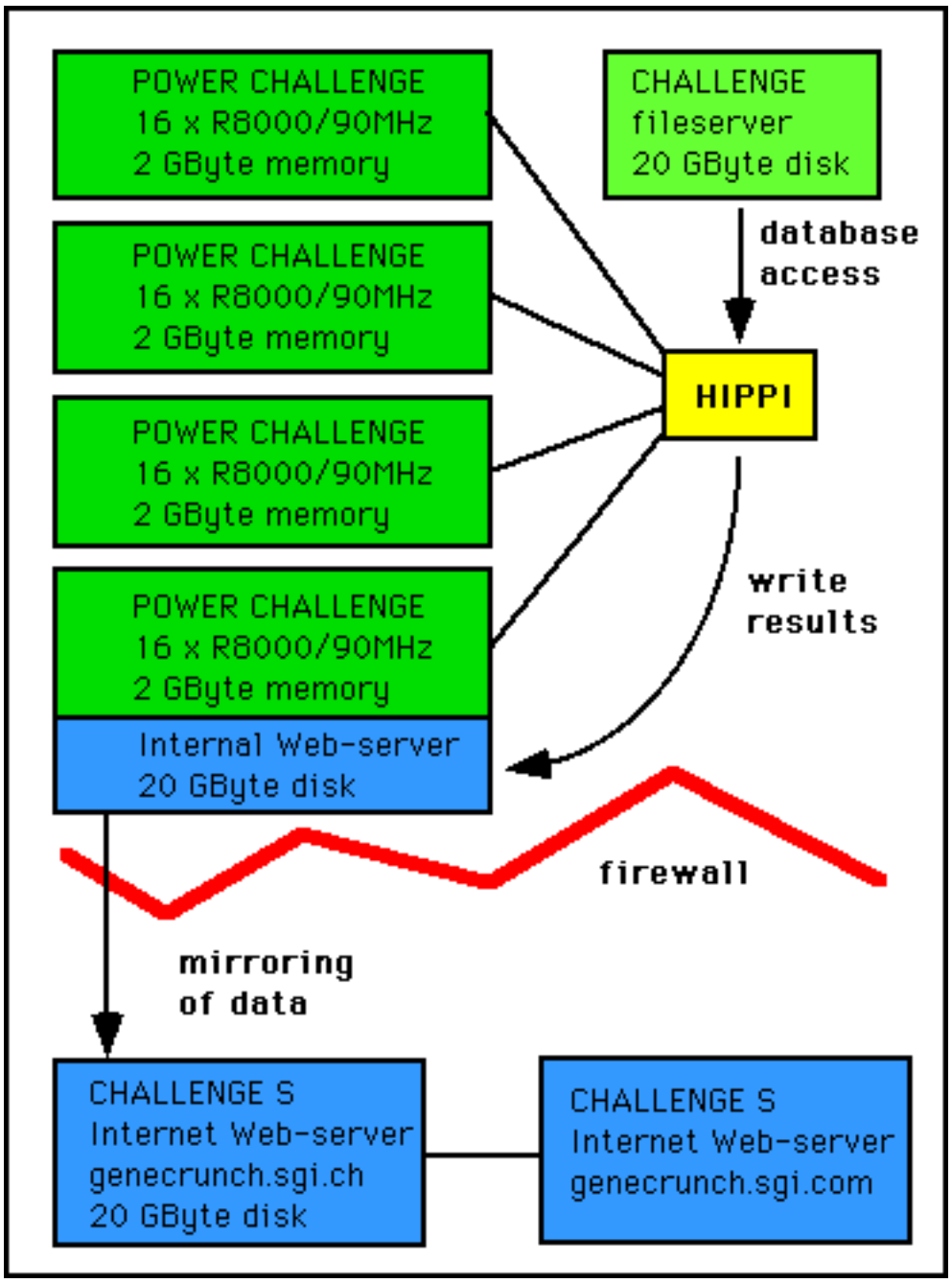
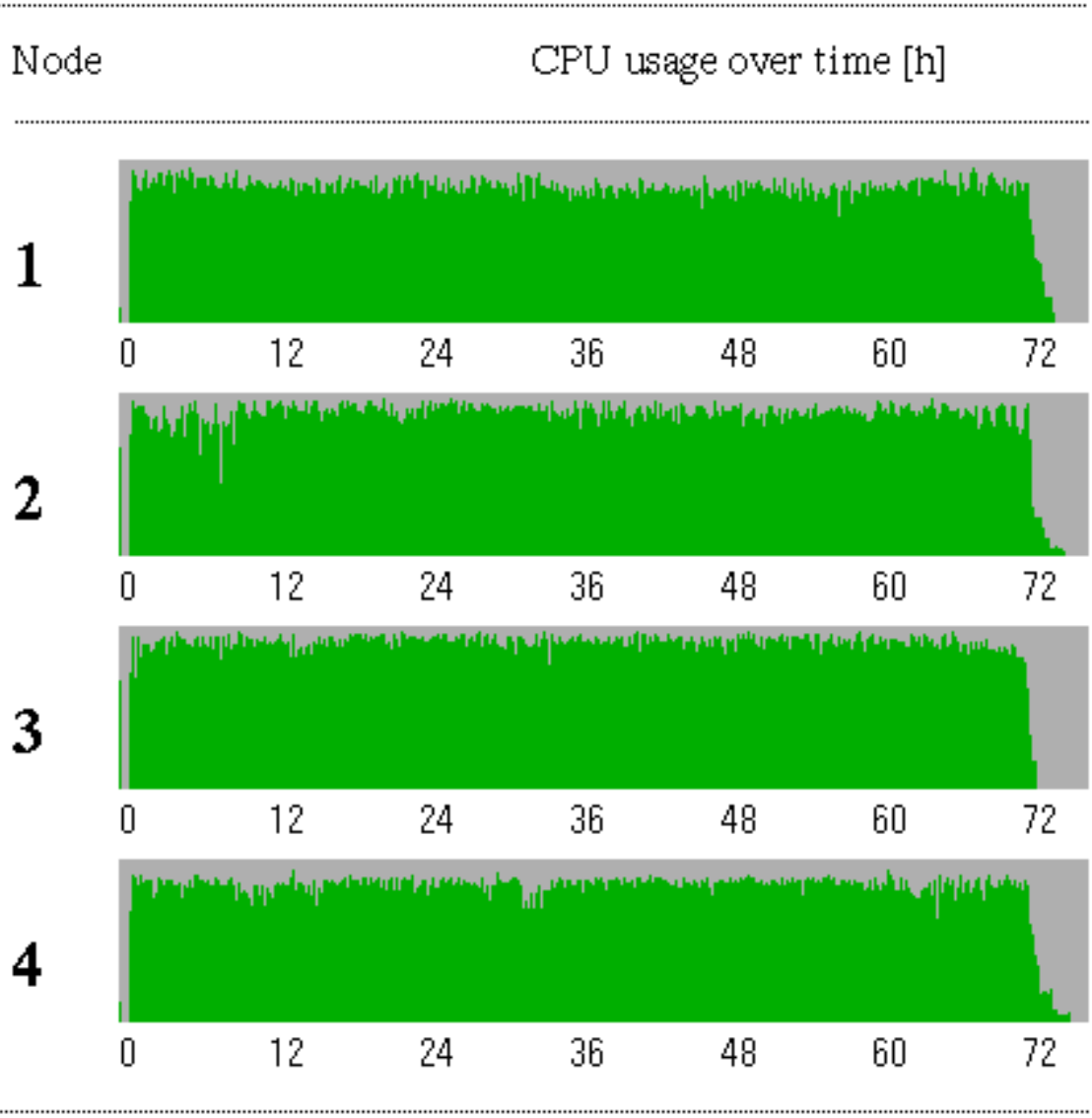


Figure 7 : Schematic drawing of the setup for the GeneCrunch project.

The external Web sites were provided with two CHALLENGE S systems located outside the firewall to provide Internet connections located in Switzerland and California. As it was generated, data was copied across the firewall to a 20GB RAID system on the Switzerland server and was NFS mounted to the server for the U.S. site. While both machines were physically located in Switzerland, the U.S. server was connected to the SGI Mountain View site through a 128K dedicated line and thereby provided external Internet access with good response times in the

U.S.(See Figure 8 for CPU usage.)

Figure 8: CPU usage



over the three days of run time of the GeneCrunch project. Each node represents one POWER CHALLENGE with 16 R8000 at 90MHz.

3.4 The Live Event

The analysis was performed March 4-7 on the POWER CHALLENGEarray placed at the SGI European Supercomputing Technology Center in Cortaillod, Switzerland. The results of the analysis were made available via two Web servers while the computations were running. This procedure--quite unusual in science, where data are not shared prior to publication in journals--was one of the novel features of the GeneCrunch project. The

results are available at the following Web address: columba.ebi.ac.uk:8765/ext-genequiz/

Given the extremely heterogeneous production environment for the GeneCrunch project, the POWER CHALLENGEarray was stable during the live event with no single hardware or system software problem (see also Figure 4). The only problem during the project was a disk crash on the outside Web server, which caused a backlog on the results for a few hours.

With the biological sequence databases used the analysis required more than 1.90 x 10^{10} sequence comparisons just for the database scanning. The complete analysis could be completed in 72 hours, which is the equivalent of more than 73,000 sequence comparisons per second (Table 1). In addition to the raw database scans, the GeneQuiz system fired off all the other tools and analysis programs to produce multiple sequence alignments for protein families, predictions of structural features up to the generation of full 3D atomic coordinate sets for model structures. In total, more than 4,200 multiple sequence alignments were performed and a few hundred 3D models were generated.

6,613 yeast sequences analyzed
185,688 entries in NR-DB (nonredundant protein database)
54,435,055 amino acids
438,305 entries in NR-EST (nonredundant EST database)
151,663,018 bases
13,226 BLAST runs
2,221 additional FASTA runs against NR-DB
1.90x10^{10} number of sequence comparisons
72 hours run time
~73,700 sequence comparisons / second
~13GB of results

Table 1 : Short summary of the computational effort for the yeast genome analysis. Note: the number of sequence comparisons given here represent only the comparisons during the database scan, not included here are the additional comparisons done for multiple sequence alignments and for internal repeat searches.

During the three-day event, scientists located at more than 1,000 sites worldwide immediately accessed the gene analysis results using the World Wide Web servers. An additional 1,000 more sites accessed our server during the following week.

The results represent a unique consistent snapshot of the function prediction of yeast protein sequences, which will take a few months to analyze in detail. A first rough analysis, however, already showed new functional predictions for a few hundred proteins.

The use of powerful supercomputers in genome analysis permits the processing of huge amounts of raw data in days instead of months. It also allows scientists to keep up with the current information growth by performing frequent reanalysis. The sheer compute power necessary to do all this is enormous. And it will continue to grow exponentially as the databases containing the raw data are doubling in size every 6-12 months.

4. Conclusion

With the Europort and GeneCrunch projects we were able to outline the capabilities of the POWER CHALLENGEarray and its usability in industrial and academic environments. The hierarchical approach allows applications to scale well beyond the limit of processors in a shared memory system. On the other hand, it provides a tightly coupled framework for programs written with shared memory parallelism or applications that require low latency and high bandwidth between the threads.

The POWER CHALLENGEarray represent an ideal machine if high throughput and turnaround time have to be optimized, since you have powerful parallel nodes. These nodes allow single or parallel programs access to a large (up to 16GB) memory, a fast file system, a fast network, and, if required, high-end visualization capabilities.

The Europort project demonstrated good scalability of the machine beyond the shared memory limit of processors showing impressive speed-ups to 64 CPUs. Also, in a mixed mode of multiple parallel jobs it demonstrated superior performance.

The GeneCrunch project demonstrated the throughput capabilities of the POWER CHALLENGEarray of applications with very different requirements for memory, processor performance, and disk performance. There is no necessity to direct specific application to specific "fat" nodes to fulfill the resource requirements, but rather the shared resource environment of the nodes provides the application with the desired resources dynamically.

The use of powerful supercomputers in genome analysis permits the processing of huge amounts of raw data in days instead of months. It also allows scientists to keep up with the current information growth by performing frequent reanalysis. The sheer compute power necessary to do all this is enormous. And it will continue to grow exponentially as the databases containing the raw data are doubling in size every 12 months.

References

[1] R. Ahlrichs and M. v. Arnim, "TURBOMOLE: Parallel Implementation" in Methods and Techniques in Computational Chemistry: METECC-95, E. Clementi editor, p. 509.

[2] F. Brakhagen and P. Lauwers, "An Efficient Paralleleigensolver for Large Matrices and a Moderate Number of Processors" in Methods and Techniques in Computational Chemistry: METECC-95, E. Clementi editor, p. 556.

[3]C. Sander and R. Schneider, "Database of Homology-Derived Protein Structures and the Structural Meaning of Sequence Alignment," Proteins, 9, 56-69, 1991.

[4] G. Casari, M. A. Andrade, P. Bork, J. Boyle, A. Daruvar, C. Ouzounis, R. Schneider, J. Tamanes, A. Valencia, and C. Sander, "Challenging Times for Bioinformatics," Nature, vol. 376, pp. 647-648, 1995 .

[5] M. Scharf, R. Schneider, G. Casari, P. Bork, A. Valencia, C. Ouzounis, and C. Sander, "GeneQuiz: A Workbench for Sequence Analysis," presented at ISMB-94 Second International Conference on Intelligent Systems in Molecular Biology, Stanford, California, 1994.

[6] L. Wall and R. L. Schwartz, "Programming perl ". Sebastopol, CA: O'Reilly & Associates, Inc., 1990.

[7] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local Alignment Search Tool," J. Mol. Biol., vol. 215, pp. 403-410, 1990 .

[8] G. Vriend, "WHAT IF : A Molecular Modeling and Drug Design Program," J. Mol. Graphics, vol. 8, pp. 52-56, 1990.

[9] VRML Architecture Group, URL: www.web3d.org/vag .

[10] H. Vollhardt, C. Henn, G. Moeckel, M. Teschner, and J. Brickmann, "Virtual Reality Modeling Language in Chemistry" , J. Mol. Graphics, vol. 13, pp. 368-372, 1995.