The PS3® Grid-resource model

Martin Rehr and Brian Vinter
eScience center, University of Copenhagen, Copenhagen, Denmark

Abstract—*This paper introduces the PS3® Grid-resource model, which allows any Internet connected Playstation 3 to become a Grid Node without any software installation. The PS3® is an interesting Grid resource as each of the over 5 millions sold world wide contains a powerful heterogeneous multi core vector processor well suited for scientific computing. The PS3® Grid node provides a native Linux execution environment for scientific applications. Performance numbers show that the model is usable when the input and output data sets are small. The resulting system is in use today, and freely available to any research project.*

Keywords: Grid, Playstation 3, MiG

1. Introduction

The need for computation power is growing daily as an increasing number of scientific areas use computer modeling as a basis for their research. This evolution has led to a whole new research area called eScience. The increasing need of scientific computational power has been known for years and several attempts have been made to satisfy the growing demand. In the 90's the systems evolved from vector based supercomputers to cluster computers which is build of commodity hardware leading to a significant price reduction. In the late 90's a concept called Grid computing[7] was developed, which describes the idea of combining the different cluster installations into one powerful computation unit.

A huge computation potential beyond the scope of cluster computers is represented by machines located outside the academic perimeter. While traditional commodity machines are usually PC's based on the X86 architecture a whole new target has turned up with the development and release of the Sony Playstation 3 (PS3®). The heart of the PS3® is the Cell processor, The Cell Broadband Engine Architecture (Cell BE)[4] is a new microprocessor architecture developed in a joint venture between Sony, Toshiba and IBM, known as STI. Each company has their own purpose for the Cell processor. Toshiba uses it as a controller for their flat panel televisions, Sony uses it for the PS3®, and IBM uses it for their High Performance Computing (HPC) blades. The development of the Cell started out in the year 2000 and involved around 400 engineers for more than four years and consumed close to half a billion dollars. The result is a powerful heterogeneous multi core vector processor well suited for gaming and High Performance Computing (HPC)[8].

1.1. Motivation

The theoretical peak performance of the Cell processor in the PS3® is 153,6 GFLOPS in single precision and 10.98 GFLOPS in double precision[4][1]. According to the press more than 5 million PS3's have been sold worldwide at October 2007. This gives a theoretical peak performance of more than 768.0 peta-FLOPS in single precision and 54.9 peta-FLOPS in double precision, if one could combine them all in a Grid infrastructure. This paper describes two scenarios for transforming the PS3® into a Grid resource, firstly the Native Grid Node (NGN) where full control is obtained of the PS3®. Secondly the Sandboxed Grid Node (SGN) where several issues have to be considered to protect the PS3® from faulty code, as the machine is used for other purposes than Grid computing.

Folding@Home[6] is a scientific distributed application for folding proteins. The application has been embedded into the Sony GameOS of the PS3®, and is limited to protein folding. This makes it Public Resource Computing as opposed to our model which aims at Grid computing, providing a complete Linux execution environment aimed at all types of scientific applications.

2. The Playstation 3

The PS3® is interesting in a Grid context due to the powerful Cell BE processor and the fact that the game console has official support for other operating systems than the default Sony GameOS.

2.1. The Cell BE

The Cell processor is a heterogeneous multi core processor consisting of 9 cores, The Primary core is an IBM 64 bit power processor (PPC64) with 2 hardware threads. This core is the link between the operating system and the 8 powerful working cores, called the SPE's for Synergistic Processing Element. The power processor is called the PPE for Power Processing Element, figure 1 shows an overview of the Cell architecture. The cores are connected by an Element Interconnect Bus (EIB) capable of transferring up to 204 GB/s at 3.2 GHz. Each SPE

1. The PS3® Cell has 6 SPE's available for applications. Each SPE is running at 3.2 GHz and capable of performing 25.6 GFLOPS in single precision and 1.83 GFLOPS in double precision.

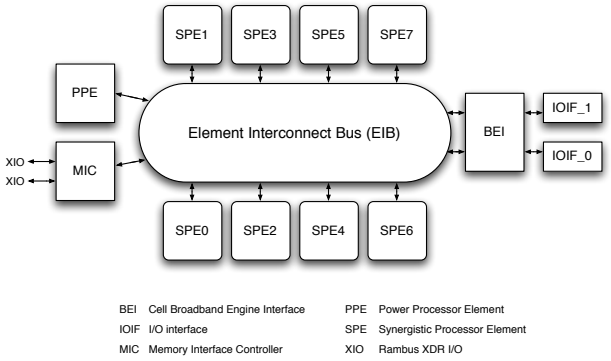


Figure 1. An overview of the Cell architecture

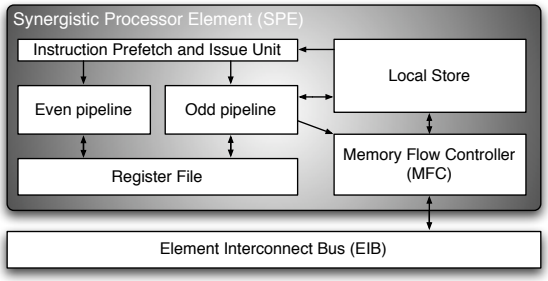


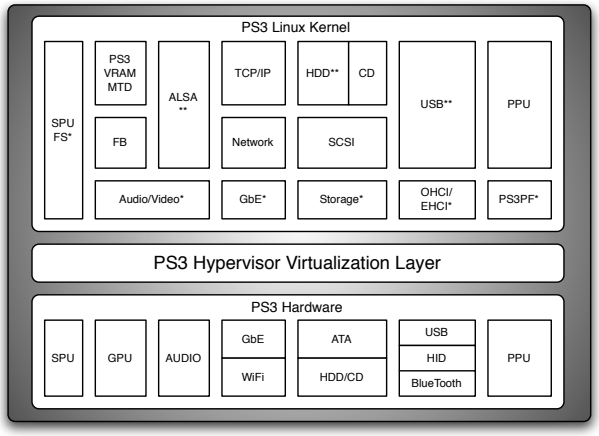
Figure 2. An overview of the SPE

is dual pipelined, has a 128x128 bit register file and 256 kB of on-chip memory called the local store. Data is transfered asynchronously between main memory and the local store through DMA calls handled by a dedicated Memory Flow Controller (MFC). An overview of the SPE is shown in figure 2.

By using the PPE as primary core, the Cell processor can be used out of the box, due to the fact that many existing operating systems support the PPC64 architecture. Thereby it's possible to boot a PPC64 operating system on the Cell processor, and execute PPC64 applications, however these will only use the PPE core. To use the SPE cores it's necessary to develop code specially for the SPE's, which includes setting up a memory communications scheme using DMA through the MFC.

2.2. The game console

Contrary to other game consoles, the PS3® officially supports alternative operating systems besides the default Sony Game OS. Even though other game consoles can be modified to boot alternative operating systems, this requires either an exploit of the default system or a replacement of the BIOS. Replacing the BIOS is intrusion at the highest level, expensive at a large volume and not usable beyond the academic perimeter. Security exploits are most likely to be patched within the next firmware update, which makes this solution unusable in any scenario. Beside the difficulties modifying other game consoles towards our purposes, the processors used by the game consoles currently on the market, except for the PS3®,



* PS3 Hypervisor Linux drivers provided by SONY
** Linux drivers NOT included on the PS3LIVE CD

Figure 3. An overview of the PS3® Hypervisor structure for the Grid-resource model

are not of any interest for scientific computing.

The fact that the PS3® is low priced from a HPC point of view, equipped with a high performance vector processor, and supports alternative operating systems, makes it interesting both as an NGN node and an SGN node. All sold PS3's can be transformed to a powerful Grid resource with a little effort from the owner of the console. Third party operating systems work on top of the Sony GameOS, which acts as a hypervisor for the guest operating system. See figure 3. The hypervisor controls which hardware components are accessible from the guest operating system. Unfortunately the GPU is not accessible by guest operating systems[2], which is a pity, as it in itself is a powerful vector computation unit with a theoretical peak performance of 1.8 tera-FLOPS in single precession. However 252 MB of the 256 MB GDDR3 ram located on the graphics card can be accessed through the hypervisor, The hypervisor reserves 32 MB of main memory and 1 of the 7 SPE's available in the PS3® version of the Cell processor[3]. This leaves 6 SPE's and 224 MB of main memory for guest operating systems. Lastly a hypervisor model always introduces a certain amount of performance decrease, as the guest operating system does not have direct access to the hardware.

3. The PS3® Grid resource

The PS3® supports alternative operating systems, making the transformation into a Grid resource rather trivial, as a suitable Linux distribution and an appropriate Grid client are the only requirements. However if you target a large amount of PS3's this becomes cumbersome. Furthermore if the PS3's

2. It is not clear whether it's to prevent games to be played outside Sony GameOS, due to DRM issues or due to the exposure of the GPU's register-level information

3. The Cell processor consists of 8 SPE's, but in the PS3® one is removed for yield purposes, if one is defective it is removed, if none is defective a good one is removed to assure that all PS3's have exactly 6 SPE's available for applications, to preserve architectural consistency

located beyond the academic perimeter are to be reached, minimal administrational work form the donator of the PS3® is a vital requirement. Our approach minimizes the workload required transforming a PS3® into a powerful Grid resource by using a LIVECD. Using this CD, the PS3® is booted directly into a Grid enabled Linux system. The NGN version of the LIVECD is targeted at PS3's used as dedicated Grid nodes, and uses all the available hardware of the PS3®, whereas the SGN version uses the machine without making any change[4] to it, and is targeted at PS3's used as entertainment devices as well as Grid nodes.

3.1. The PS3-LIVECD

Several requirements must be met by the Grid middleware to support the described LIVECD. First of all the Grid middleware must support resources which can only be accessed through a pull based model, which means that all communication is initiated by the resource, i.e. the PS3-LIVECD. This is required because the PS3's targeted by the LIVECD are most likely located behind a NAT router. Secondly, the Grid middleware needs a scheduling model where resources are able to request specific types of jobs, e.g. a resource can specify that only jobs which are targeted the PS3® hardware model can be executed.

In this work the Minimum intrusion Grid[11], MiG, is used as the Grid middleware. The MiG system is presented next, before presenting how the PS3-LIVECD and MiG work together.

3.2. Minimum intrusion Grid

MiG is a stand-alone Grid platform, which does not inherit code from any earlier Grid middlewares. The philosophy behind the MiG system is to provide a Grid infrastructure that imposes as few requirements on both users and resources as possible. The overall goal is to ensure that a user is only required to have a X.509 certificate which is signed by a source that is trusted by MiG, and a web browser that supports HTTP, HTTPS and X.509 certificates. A fully functional resource only needs to create a local MiG user on the system and to support inbound SSH. A sandboxed resource, the pull based model, only needs outbound HTTPS[1].

Because MiG keeps the Grid system disjoint from both users and resources, as shown in Figure 4, the Grid system appears as a centralized black box[11] to both users and resources. This allows all middleware upgrades and trouble shooting to be executed locally within the Grid without any intervention from neither users nor resource administrators. Thus, all functionality is placed in a physical Grid system that, though it appears as a centralized system in reality is distributed. The basic functionality in MiG starts by a user submitting a job to MiG and a resource sending a request for a job to execute. The resource then receives an appropriate job from MiG, executes the job, and sends the result to MiG that

4. One has to install a boot loader to be able to boot from CD's

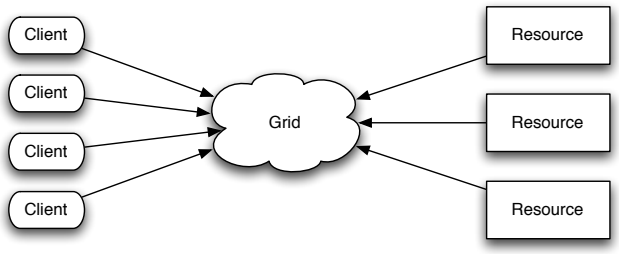


Figure 4. The abstract MiG model

can then inform the user of the job completion. Since the user and the resource are never in direct contact, MiG provides full anonymity for both users and resources, any complaints will have to be made to the MiG system that will then look at the logs that show the relationship between users and resources.

3.2.1. Scheduling. The centralized black box design of MiG makes it capable of strong scheduling, which implies full control of the jobs being executed and the resource executing them. Each job has an upper execution time limit, and when the execution time exceeds this time limit the job is rescheduled to another resource. This makes the MiG system very well suited to host SGN resources, as they by nature are very dynamic and frequently join and leave the Grid without notifying the Grid middleware.

4. The MiG PS3-LIVECD

The idea behind the LIVECD is booting the PS3® by inserting a CD, containing the Linux operating system and the appropriate Grid clients. Upon boot, the PS3® connects to the Grid and requests Grid jobs without any human interference. Several issues must be dealt with. First of all the PS3® must not be harmed by flaws in the Grid middleware nor exploits through the middleware, Secondly the Grid jobs may not harm the PS3® neither by intention nor by faulty jobs. This is especially true for SGN resources where an exploit may cause exposure of personal data.

4.1. Security

To keep faulty Grid middleware and jobs from harming the PS3®, both the NGN and SGN model use the operating system as a security layer. The Grid client software and the executed Grid jobs are both executed as a dedicated user, who does not have administrational rights of the operating system. The MiG system logs all relations between jobs and resources, thus providing the possibility to track down any job.

4.2. Sandboxing

The SGN version of the LIVECD operates in a sandboxed environment to protect the donated PS3® from faulty middleware and jobs. This is done by excluding the device driver for the PS3® HDD controller from the Linux kernel

used, and keeping the execution environment in memory instead. Furthermore, the support for loadable kernel modules is excluded, which prevents Grid jobs from loading modules into the kernel, even if the OS is compromised and root access is achieved.

4.3. File access

Enabling file access to the Grid client and jobs without having access to the PS3's hard drive is done by using the graphics card's VRAM as a block device. Main memory is a limited resource[5], therefore using the VRAM as a block device is a great advantage compared to the alternative of using a ram disk, which would decrease the amount of main memory available for the Grid jobs. However the total amount of VRAM is 252 MB and therefore Grid jobs requiring input/output files larger than 252 MB are forced to use a remote file access framework[2].

4.4. Memory management

The PS3® has 6 SPE cores and a PPE core all capable of accessing the main memory at the same time, through their MFC controllers. This results in a potential bottleneck in the TLB, as it in the worst case ends up thrashing, which is a known problem in multi core processor architectures. TLB thrashing can be eliminated by adjusting the page size to fit the TLB, which means that all pages have an entry in the TLB. This is called huge pages, as the page size grows significantly. The use of huge pages has several drawbacks, one of them is swapping. Swapping in/out a huge page results in a longer execution halt as a larger amount of data has to be moved between main memory and the hard drive.

The Linux operating system implements huge pages as a memory mapped file, this results in a static memory division of traditional pages and huge pages, using different memory allocators. The operating system and standard shared libraries use the traditional pages which means the memory footprint of the operating system and the shared libraries has to be estimated in order to allocate the right amount of memory for the huge pages. Opposite to a cluster setup where the execution environment and applications are customized to the specific cluster, this can't be achieved in a Grid context[6]. Therefore a generic way of addressing the memory is needed. Furthermore future SPE programming libraries will most likely use the default memory allocator. This and the fact that no performance measurement clarifying the actual gain of using huge pages could be found, led to the decision to skip huge pages for the PS3-LIVECD.

At last it's believed by the authors that the actual applications which could gain a performance increase by using huge pages is rather insignificant, as the the majority of applications will be able to hide the TLB misses by using double- or multi buffering, as memory transfers through the MFC are asynchronous.

5. The PS3® only has 224 MB of main memory for the OS and applications

6. Specially in MiG where the user and resources are anonymous to each other

5. The execution environment

The PS3-LIVECD is based on the Gentoo Linux[9] PPC64 distribution with a customized kernel[5] capable of communicating with the PS3® hypervisor. Gentoo catalyst[3] was used as build environment, this provides the possibility of configuring exactly which packages to include on the LIVECD, as well as providing the possibility to apply a custom made kernel and initrd script. The kernel was modified in different ways, firstly loadable modules support was disabled to prevent potential evil jobs, which manages to compromise the OS security, from modifying the kernel modules. Secondly the frame-buffer driver has been modified to make the VRAM appear as a memory technology device, MTD, which means that the VRAM can be used as a block device. The modification of the frame-buffer driver also included freeing 18 MB of main memory occupied by the frame-buffer used in the default kernel[7].

The modified kernel ended up consuming 7176 kB of the total 229376 kB main memory for code and internal data structures, leaving 222200 kB for the Grid client and jobs. Upon boot the modified initrd script detects the block device to be used as root file system[8] and formats the detected device with the ext2 filesystem, reserving 2580 kB for the superuser, leaving 251355 kB for the Grid client and jobs[9]. When the block device has been formatted, the initrd script sets up the root file system by coping writable directories and files from the CD to the root file system. Read-only directories, files, and binaries are left on the CD and linked symbolically to the root filesystem keeping as much of the root filesystem free for Grid jobs as possible. The result is that the root file system only consumes 1.6 MB of the total space provided by the used block device.

When the Linux system is booted the LIVECD initiates the communication with MiG through HTTPS. This is done by sending a unique key identifying the PS3® to the MiG system, if this is the first time the resource connects to the Grid a new profile is created dynamically. The response to the initial request is the Grid resource client scripts, these are generated dynamically upon the request. By using this method it's guaranteed that the resource always has the newest version of the Grid resource client scripts, disabling the need for downloading a new CD upon a Grid middleware update. When the Grid resource client script is executed the request of Grid jobs is initiated through HTTPS. Within that request a unique resource identifier is provided, giving the MiG scheduler the necessary information about the resource such as architecture, memory, disc space and an upper time-limit. Based on these parameters the MiG scheduler finds a job suited for the PS3® and places it in a job folder on the MiG system. From this location the PS3® is able to retrieve the job consisting of

7. As the hypervisor isolates the GPU from the operating system, the display is operated by having the frame-buffer writing the data to be displayed to an array in main memory, which is then copied to the GPU by the hypervisor

8. The SGN version uses VRAM, DGN version uses the real hard drive provided through the hypervisor

9. This is true for the SGN version, the NGN version uses the total disc space available, which is specified through the Sony Game OS

job description files, input-files, and executables. The location of these files is returned within the result of the job request, and is a HTTPS URL including a 32 character random string generated upon the job request and deleted when the job terminates. At job completion the result is delivered to the MiG system which verifies that it's the correct resource (by the unique resource key) which delivers the result of the job. If it's a false deliver[10] the result is discarded, otherwise it's accepted. And the PS3® resource requests a new job when the result of the previous one has been delivered.

6. Experiments

Testing the PS3® Grid-resource model was done establishing a controlled test scenario consisting of a MiG Grid server and 8 PS3's. The experiments performed included a model overhead check, a file system benchmark using VRAM as a block device, and application performance, using a protein folding and a ray tracing application.

6.1. Job overhead and file performance

The total overhead of the model was tested by submitting 1000 empty jobs to the Grid with only one PS3® connected. The 1000 jobs completed in 12366 seconds, which translates to an overhead of approximately 13 seconds per job. The performance of the VRAM used as a block device was tested by writing a 96 MB file sequentially. This was achieved in 1.5 seconds, resulting in a bandwidth of 64 MB/s. Reading the written file was achieved in 9.6 seconds, resulting in a bandwidth of 10 MB/s. This shows that writing to the VRAM is a factor of approximately 6.5 faster than reading from the VRAM, which was an expected result as the nature of VRAM is write from main memory to VRAM, not the other way around.

6.2. Protein folding

Protein folding is a compute intensive algorithm for folding proteins. It requires a small input and generates a small output, and is embarrassing parallel which makes it very suitable for Grid computing. In this experiment, a protein of length 27 was folded on one PS3® resulting in a total execution time of 57 minutes and 16 seconds. The search space was then divided into 17 different subspaces using standard divide and conquer techniques. The 17 different search spaces were then submitted as jobs to the Grid, which adds up to 4 jobs for each of the 4 nodes used in the experiment plus one extra job to ensure unbalanced execution. Equivalently, the 17 jobs were distributed among 8 nodes, yielding 2 jobs per node plus one extra job. The execution finished in 18 minutes and 50 seconds using 4 nodes giving a speedup of 3.04. The 8 node setup finished the execution in 10 minutes and 56 seconds giving a speedup of 5.23, this is shown in figure 5. These results are considered quite useful in a Grid setup as opposed to a cluster setup where this would be considered bad.

10. The resource keys doesn't match, the time limit has been violated, or another resource is executing the job, due to a rescheduling

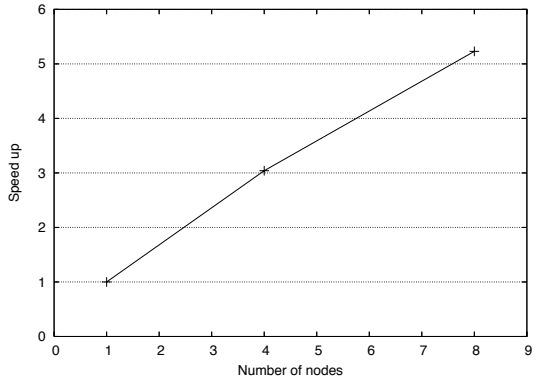


Figure 5. The speedup achieved using the PS3-LIVECD for protein folding with 4 and 8 nodes

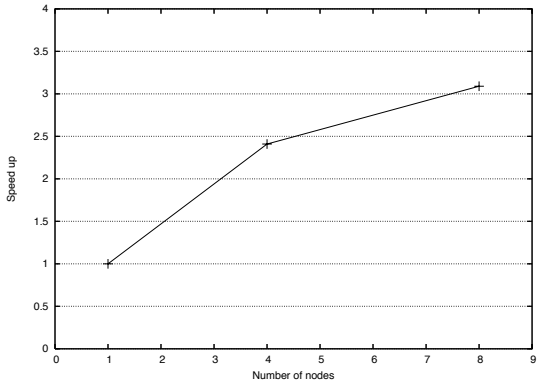


Figure 6. The speedup achieved using the PS3-LIVECD for ray tracing with 4 and 8 nodes

6.3. Ray tracing

Ray tracing is compute intensive, requires a small amount of input and generates a large amount of output. This experiment uses a Ray tracing code written by Eric Rollings[10], this has been modified from a real time ray tracer to a ray tracer which writes the rendered frames to files in a resolution of 1920x1080 (Full HD). The final images are jpeg compressed to reduce the size of the output. A total of 5000 frames were rendered in 78 minutes and 6 seconds on a single PS3®, the search space was then divided into 25 equally large subspaces. These were submitted as jobs to the Grid resulting in a total of 25 jobs, which adds up to 6 jobs per node plus one extra in the 4 node setup, and 3 jobs per node plus one extra in the 8 node setup. The execution time using 4 nodes was 32 minutes and 23 seconds giving a speedup of 2.41 and the execution time using 8 nodes was 25 minutes and 12 seconds giving a speedup of 3.09, this is sketched in figure 6. While the speedup achieved with 4 nodes is quite useful in a Grid context, the speedup gained using 8 nodes is quite disappointing. The authors

believe this is due to network congestion when the rendered frames are sent to the MiG storage upon job termination.

7. Conclusion

In this work we have demonstrated a way to use the Sony Playstation 3 as a Grid computing device, without the need to install any client software on the PS3®. The use of the Linux operating system provides a native execution environment suitable for the majority of scientific applications. The advantage of this is that existing Cell applications can be executed without any modifications. A sandboxed version of the execution environment has been presented which denies access to the hard drive of the PS3®. The advantage of this is that donated PS3's can't be compromised by faulty or evil jobs, the disadvantage is the lack of file access, which is solved by using the VRAM of the PS3 as block device.

The Minimum intrusion Grid supports the required pull-job model for retrieving and executing Grid jobs on a resource located behind a firewall without the need to open any incoming ports. By using the PS3-LIVECD approach any PS3® connected to the Internet can become a Grid resource by booting it with the LIVECD. When a Grid connected PS3® is shut down the MiG system will detect this event, by a timeout, and resubmit the job to another resource.

Experiments show that the ray tracing application doesn't scale well, due to the large amount of output data resulting in network congestion problems. Opposite to this, a considerable speedup is reached when folding proteins despite of the model overhead of 13 seconds applied to each job.

References

[1] Rasmus Andersen and Brian Vinter. Harvesting idle windows cpu cycles for grid computing. In Hamid R. Arabnia, editor, *GCA*, pages 121–126. CSREA Press, 2006.

[2] Rasmus Andersen and Brian Vinter. Transparent remote file access in the minimum intrusion grid. In *WETICE '05: Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*, pages 311–318, Washington, DC, USA, 2005. IEEE Computer Society.

[3] Gentoo Catalyst. http://www.gentoo.org/proj/en/releng/catalyst.

[4] Thomas Chen, Ram Raghavan, Jason Dale, and Eiji Iwata. Cell broadband engine architecture and its first implementation. *IBM developerWorks*, 2005. http://www.ibm.com/developerworks/power/library/pa-cellperf.

[5] PS3 Linux extensions. ftp://ftp.uk.linux.org/pub/linux/Sony-PS3.

[6] Folding@home. http://folding.stanford.edu.

[7] Ian Foster. The grid: A new infrastructure for 21st century science. *Physics Today*, 55(2):42–47, 2002.

[8] Mohammad Jowkar. Exploring the Potential of the Cell Processor for High Performance Computing. Master's thesis, University of Copenhagen, Denmark, August 2007.

[9] Gentoo Linux. http://www.gentoo.org.

[10] Eric Rollings. Ray tracer. http://eric_rollins.home.mindspring.com/ray/ray.html.

[11] Brian Vinter. The Architecture of the Minimum intrusion Grid (MiG). In *Communicating Process Architectures 2005*, sep 2005.