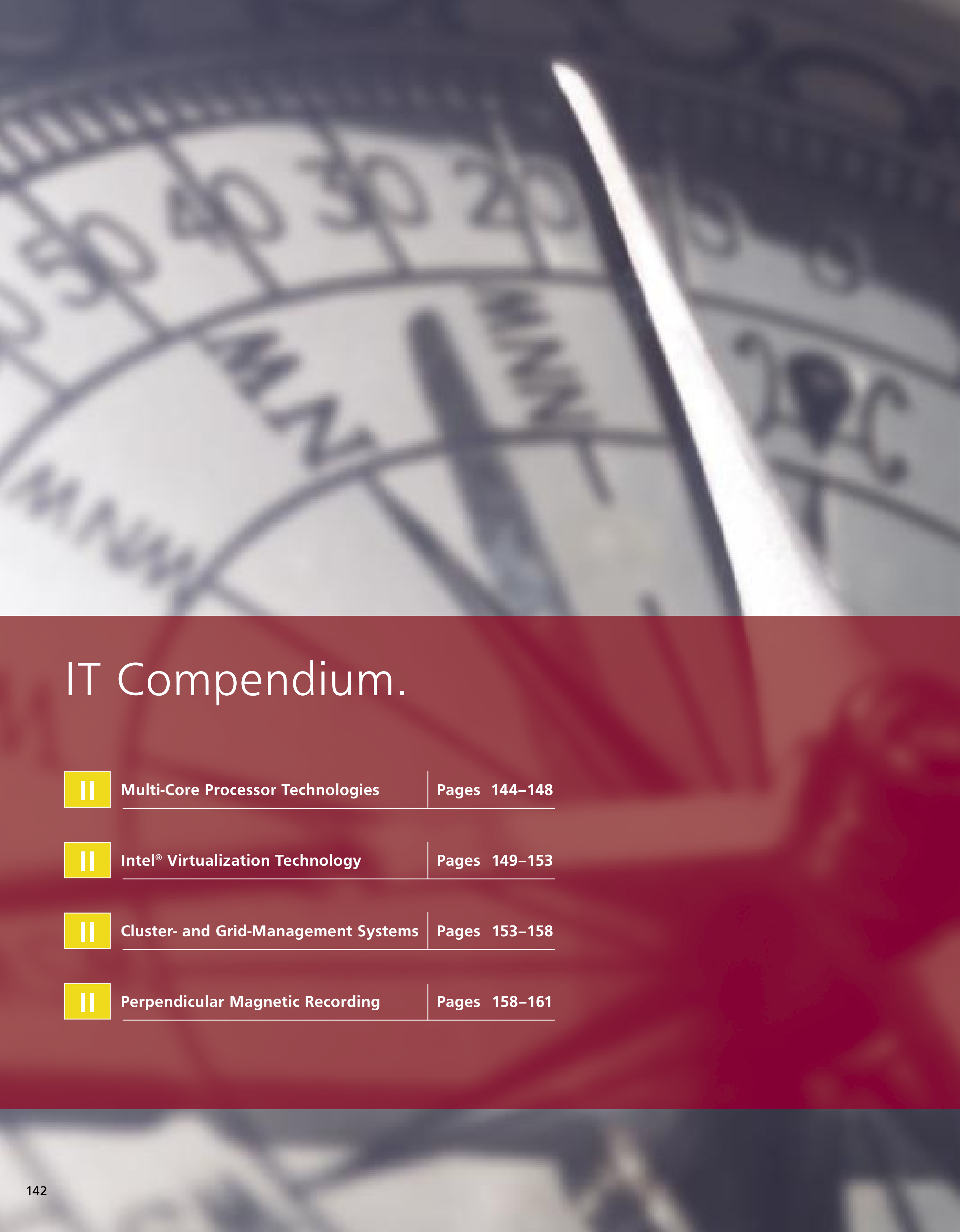
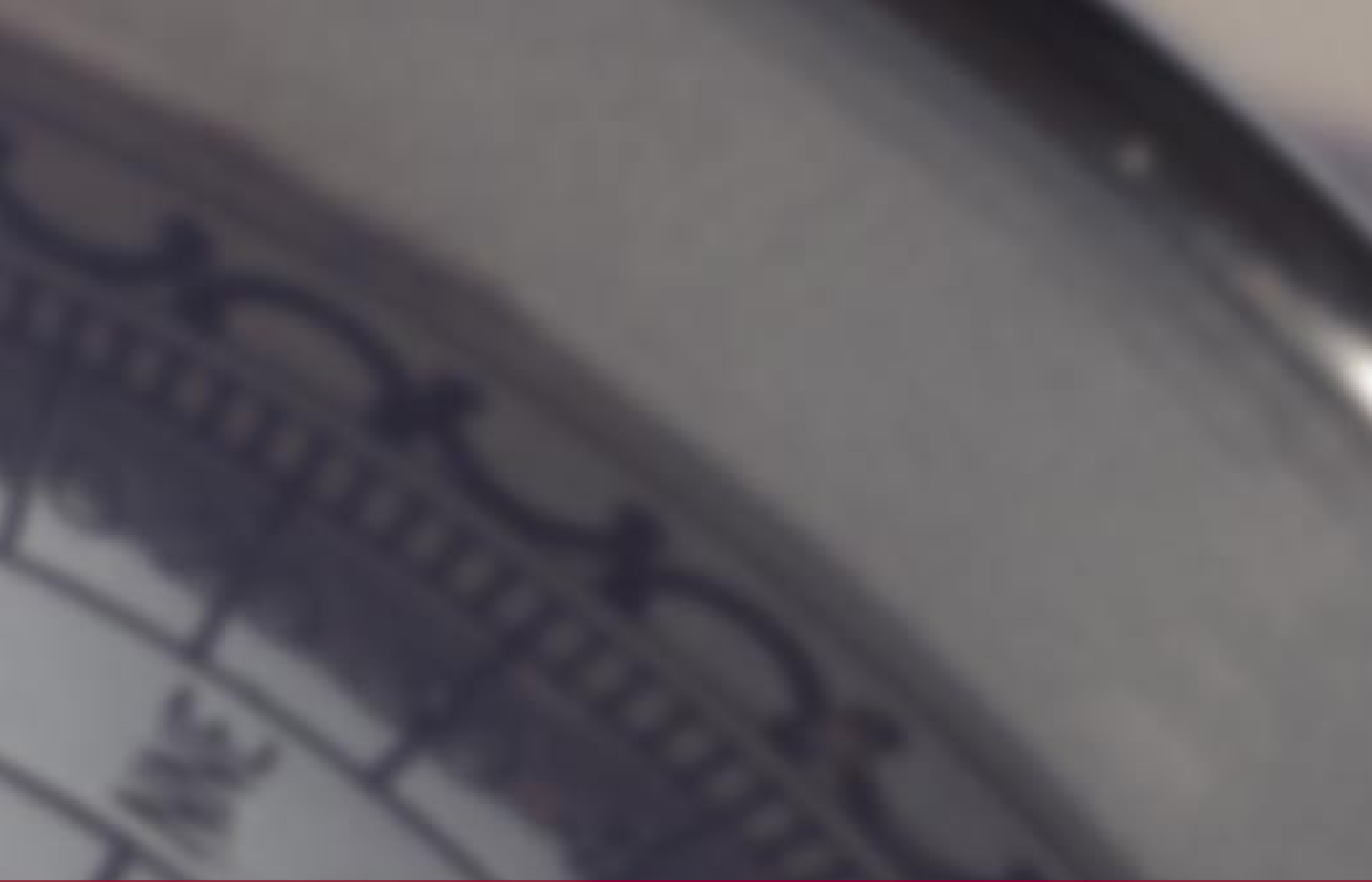


IT Compendium.



⬜ in the IT Compendium on pages 144 – 161

🟨 online on our website:
www.transtec.co.uk., www.ttec.nl, www.ttec.be,
www.transtec.ch

1. Computer Architectures

1.1.9 Multi-Core processor technologies

The introduction of multi-core processors in 2006 made this a very important year for processor development – one has to look back quite a long time for the introduction of a new technology that has had a similar impact on CPU performance. It was this development that opened up the way to parallel computing.

1.1.9.1 Some terminology

Single-core CPUs (Central Processing Units)

Prior to the introduction of dual-core CPUs by AMD and Intel® in 2004, a CPU could be said to consist of the following units:

- ALU (Arithmetic and Logic Unit): The processor unit responsible for executing machine instructions.
- FPU (Floating Point Unit): A coprocessor performing floating point arithmetic.
- L1 (Level 1) cache: A fast memory bank used for intermediate storage of instructions and data. Ideally, the L1 cache should be clocked at the same speed as the CPU.
- L2 (Level 2) cache: Intermediate memory that feeds the L1 cache using a look-ahead algorithm. Both L1 and L2 caches increase the CPU instruction throughput.

Other CPU functional units such as memory interfaces, instruction decoders and the like will not be treated here in any detail.

Cores

A core has in principle the same architecture as a complete CPU. Since the memory interface is not part of the core all cores must share it in order to communicate with main memory. The same applies to other communication interfaces such as the HyperTransport bus.

There are different strategies for implementing the interface between cores and L2 cache. In the case of dual-core Intel® Core™2 Duo processors (Conroe, Merom, Woodcrest) the L2 cache is shared by the cores. On the other hand, each core of AMD Opteron™ dual-core processors and the Athlon™ 64 X2 processor have their own L2 cache.

1.1.9.2 Why multi-core?

Before the introduction of the first AMD Opteron™ dual-core CPUs in 2005, performance gains were achieved primarily through faster clock speeds and improvements in architecture.

Continuous improvements in die fabrication processes made it relatively easy to achieve faster clock speeds. 1 GHz was normal in 2001 and was found, for example, in the Intel® Pentium® 4 Willamette Core and the AMD Athlon™ Thunderbird, both of which were fabricated with 180 nm processes.

The introduction of 90 nm and 65 nm processes allowed clock speeds to be increased to 3.5 GHz by 2005.

The optimistic predictions that CPU clock speeds of 5 GHz and higher could be achieved proved illusory. The reason is the dependence of power consumption P on the CPU core voltage U and the clock speed f given by

$$\mathbf{P = C_{CPU}U^2f}$$

This relationship is a fundamental property of CMOS chips, which form the basis for current processor technology. C_{CPU} is a constant of proportionality which depends on the CPU architecture and feature width employed – in general, the smaller the process, the smaller the C_{CPU} value. Further improvements can be expected from the transition from 65 nm to 45 nm processes.

The typical power consumption of a CPU clocked at 1 GHz is 60 W while the higher-clocked CPUs introduced in 2005 sometimes consumed over 100 W. The thermal penalty of such Wattage is quite high – CPU temperatures of 70°C were not unusual. The ensuing problems, including the resulting rise in case temperature, are difficult to remedy through fans and improved airflow.

Side effects such as electromigration are compounded by higher CPU temperatures (the law of Arrhenius) and lead to decreased CPU life expectancy.

Since faster clock speeds require higher core voltage U, the U^2 dependence will at some point lead to unacceptably high power consumption P. In a more rigorous treatment the dependence of U^2 on f would also have to be taken into account.

Chip manufacturers developed CPUs with two or more cores in response to the above limitations: since multi-core processors are run at relatively low clock speeds, power consumption can be reduced to acceptable levels.

1.1.9.3 Fabrication and architecture of multi-core CPUs

Modern 65 nm processes make it feasible to employ dual-core technology in mobile computers. High-end desktop and server CPUs from Intel® already have four cores.

The performance edge of quad-core CPUs over even the fastest single-core processors represents a dramatic breakthrough of the likes not seen in many years.

Current 65 nm and 90 nm processes allow the manufacturing of dual-core processors. Intel® quad-core Core™2 Extreme QX6700 and Xeon® 5300 processors have two dual-core chips in an LGA package. Although this technology requires complex, cost-intensive fabrication techniques, it is nevertheless employed by manufacturers to get their quad-core designs on the market as soon as possible.

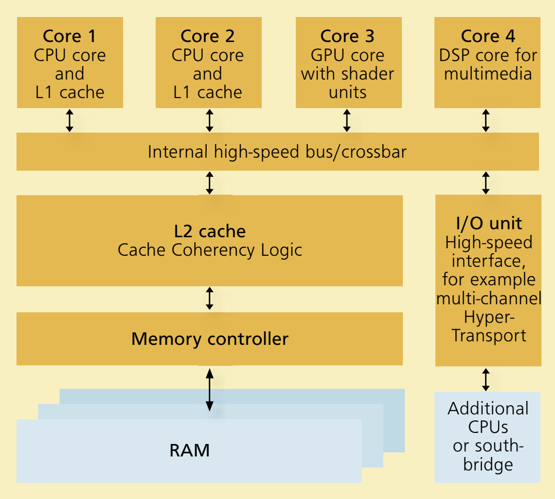
The only long-term multi-core solution that will reduce the costs of complex, error-prone bondings is one chip per package.

Manufacturers have taken different approaches to the L2-cache architectures used in multi-core processors. Intel® makes use of a single L2 cache that is shared by two cores. This technique is advantageous for programs that consist of a single thread since in this case a single core has exclusive access to a large cache. AMD follows the opposite approach, giving each core its own small L2 cache. This has the advantage of eliminating the need for complex cache coherency strategies. The problems associated with the co-ordination of shared L2 caches can be expected to become even more serious with future multi-core processors.

Intel® uses the Front Side Bus for communication between cores. On the other hand, AMD multi-core CPUs employ a crossbar switch as core interconnect. This approach will prevail in the long run since no degradation in external bus system performance results and because it allows the crossbar switch to be clocked at a higher speed than the external bus.

AMD's future quad-core Barcelona integrates all cores in a single chip package. Here, too, the crossbar switch is used as core interconnect. The Barcelona has a novel feature, the use of an L3 cache.

It should be pointed out that neither AMD nor Intel® invented multi-core technology: a three-core chip was developed in 1997 at the Massachusetts Institute of Technology.


Core 1
CPU core and L1 cache
Core 2
CPU core and L1 cache
Core 3
GPU core with shader units
Core 4
DSP core for multimedia
Internal high-speed bus/crossbar
L2 cache
Cache Coherency Logic
Memory controller
I/O unit
High-speed interface, for example multi-channel Hyper-Transport
RAM
Additional CPUs or south-bridge

A future Multi-Core CPU with two specialised cores for graphic and media-streaming, for example HDTV-decoding/-encoding by both cores simultaneously

1.1.9.4 Software and multi-core CPUs

Single-core CPUs are based on the "von Neumann architecture" and the same is true for a single core of a multi-core CPU. A von Neumann machine executes instructions sequentially. The most direct way to increase its performance is to use higher clock speeds and wider registers, for example 54-bit width instead of 32-bit.

It is also possible to reduce the number of clock cycles per instruction through improved architecture. RISC processors are one example; they achieve performance boosts through a small instruction set allowing faster instruction execution. Another technique involves the use of longer instruction pipelines to reduce instruction latency by allowing better utilisation of CPU logic.

VLIW (Very-Long Instruction Word) and similar technologies have been under development for several years now but still have to reach full maturity.

However, as stated earlier, the ability of this technique to deliver additional performance boosts is limited by currently available technology.

Multi-core technology uses two or more cores in a single CPU, each built around a von Neumann architecture. While in principle the performance gain so achieved is proportional to the number of cores, there are practical limitations that stand in the way of ideal performance scaling. The two most important are:

Memory interface bandwidth limitations
A multi-core CPU cannot provide each core with its own memory interface. The memory interface can be a source of bottlenecks for memory-intensive applications such as CAE (Computer Aided Engineering) and aero- and fluid-dynamic simulations where memory requirements can reach several gigabytes.

In such applications a computer with two dual-core processors can deliver better performance than one with a single quad-core CPU since each core in the two processor solution has its own memory banks.

A larger L2 cache brings benefits but also occupies considerably more chip area. An L2 cache uses SRAM (Static RAM) where four or six transistors per bit are used, depending on the design. This puts practical limits on the size of an L2 cache.

Multithreading
For proper core utilisation, multi-core processors require that programmes be divided into several threads, that is subprogrammes that execute independently. A programme with two threads can benefit from execution on a dual-core CPU; however two threads will not lead to increased performance on a quad-core processor.

Programmers are currently making efforts to improve the multithreading capabilities of diverse programmes. This is a difficult undertaking and progress has been slow. The unsuitability for multithreading of the algorithms used in some programs poses a problem – 3D CAD software, which is based on techniques drawn from constructive solid geometry, is a notable example and many such programmes are still single-threaded.

Photorealistic rendering programmes, on the other hand, can take full advantage of multithreading on multi-core processors. Since these programmes use ray tracing algorithms to simulate illumination and since light rays can be superimposed without mutual interaction, it is in principle possible to simultaneously calculate all pixels in an image.

One example is Povray, an open-source ray tracing programme. Since the programme core occupies only a few hundred kilobytes it can be easily stored in the L2 cache of all current multi-core CPUs. Scenery data is based on constructive solid geometry and therefore has only very modest memory demands; therefore the memory interface is not the bottleneck.

Povray’s performance is proportional to the number of cores employed and the speed at which they are clocked. It is therefore ideally suited to multithreading applications.

1.1.9.5 Operating system issues

The operating system scheduler has the responsibility of distributing processes to individual cores in such a way as to achieve the highest-possible core utilisation. A process is an instance of an executing programme, memory pages, variables, protection attributes and CPU register state. If the scheduler has to handle a large number of processes it may not be able to distribute processes over cores in an optimal fashion. The scheduler’s task is complicated by the fact that individual processes may be multithreaded.

Therefore Windows® XP exhibited difficulties in using first generation dual-core processors. Future multi-core CPUs provide the operating system with 32 and more cores; this can be achieved using four 8-core CPUs and can be expected to find application in the server area as soon as 2009.

This development will confront operating system developers with even greater challenges in achieving optimal core utilisation. It can only be hoped that developments in software keep pace with those in hardware.

1.1.9.6 Other multi-core architectures

Cell Broadband Engine

Cell Broadband Engine (CBE), the result of a joint effort of IBM®, Toshiba® and Sony®, is a promising new development in multi-core technology. It is used in Sony's PlayStation® 3 and an upgrade board allowing its use on PCs is already available. Such boards make use of CBE's high floating point performance, as does Computer Aided Engineering software.

Intel® and AMD multi-core designs depart in several respects from CBE architecture. The differences can be traced to the fact that in the case of CBE one is not dealing with functionally equivalent cores.

CBE is made up of the following functional units:

- PPE (Power Processor Element), a PowerPC®-based CPU
- Eight SPE (Synergistic Processing Element) units for cores optimised for integer and floating point arithmetic
- EIB (Element Interconnect Bus), a fast bus system used for internal communication between SPEs and PPE
- A memory controller based on Rambus XDRAM technology.

The PPE is a complete PowerPC® CPU because it functions as the interface to the operating system. Arithmetic commands are passed to the eight SPEs, which together achieve up to 200 GFLOP single-precision performance and a very respectable 14 GFLOPS with double precision, making the CBE a powerful FPU accelerator. IBM® has plans to create CBE-based blades, which are intended for use in the supercomputing area.

SPEs resemble vector processors, used, for example, in Cray supercomputers. They can deliver full performance only when programme code is tailored to their specific architecture.

The next CBE generation will be fabricated with 65 nm technology and will have SPEs optimised for double-precision arithmetic, making it easier to design very powerful FPU accelerators. For most numerical applications simple precision is just not adequate.

NVIDIA GeForce® 8800 GPU

Current graphic cards employ several shader units. Such units may be considered as special-purpose FPUs (Floating Point Unit). In the case of the NVIDIA® GeForce® 8800 GPU (Graphics Processing Unit) there are 128 shader units which are not restricted to pure vertex or pixel shading.

Each shader unit can perform single-precision arithmetical calculations. There exist C routines, which already can achieve very good results with parallelisation of numerical algorithms with those shader units, for example the FFT (Fast Fourier Transformation), used in image processing. When executing high-performance BLAS3 matrix operation routines the GeForce® 8800 GPU is capable of achieving around 100 GFLOPS.

Shader units are no match for conventional CPUs in terms of flexibility. Their very limited instruction set impedes programme development. Programmes with extended branching and many conditional transfers are not suitable for execution on shader units. While present shader units are limited to single-precision arithmetic, this drawback will be overcome in next-generation double-precision GPUs.

1.1.9.7 Outlook

The introduction of multi-core CPUs is undoubtedly a revolutionary development. In the course of 2007 dual-core architectures will gain broad acceptance in entry-level PCs, this coinciding with widespread use of quad-core workstations and servers.

65 nm technology will be implemented in 2007 and Intel®'s roadmap calls for the transition to the 45 nm process in 2008. This will lead to server chips with up to a billion transistors. Fabs for manufacturing CPUs with 65 nm technology cost over two billion dollars and are affordable by only the largest semiconductor producers or joint ventures.

The most important consequences of these developments are:

- Further miniaturisation of transistors will result in significant increases in leakage current, which is already on the same order of magnitude as the dynamic switching power.
- Today's CPUs have core voltages of 1.1 V. Further decreases in core voltage will be difficult to realise – 0.7V is the lowest feasible value. According to the $P = C_{CPU} U^2 f$ law, this means that there is a definite limit on how far power consumption can be reduced by lowering the core voltage.

In the past, a reduction in wafer thickness allowed clock speed to be increased while at the same time lowering power consumption. Physical constraints will prevent this trend from continuing: CPUs are fast hitting the so-called "power wall".

Further improvement in multi-core technology is the only available option left to CPU developers. With the advent of 32 nm technology the number of cores per CPU should rise to between 50 and 100. Power consumption issues are likely to stand in the way of significant increases in clock speed. The physical limitation on the number of pins of a CPU package means that only modest improvements in the CPU-RAM interface are possible; this will place an upper limit on the performance of future multi-core CPUs. Conventional cache technology is likewise a limiting factor since it cannot feed cores with data fast enough.

Software and operating system developers are entering a period in which parallel programming is more or less mandatory in order that the computing load be equally distributed over multiple cores. The development of massive multithreaded programmes is required, a task that is significantly more complex and cost-intensive than programming single-threaded software since it requires the development of new algorithms. Progress in this area lags behind developments in CPU technology.

Stream programming might provide a solution. Using a special compiler, this technique attempts to minimise communication between individual functional units of a CPU chip and loads required code in a local cache before code execution begins. This avoids "cache misses" which result in an interruption of the flow of data to a core. GPUs already benefit from stream programming.

The availability of the appropriate software has a decisive influence on the performance boost achievable by multi-core CPUs. For computer simulation applications, well-parallised software is already available and therefore science and engineering will be the first to benefit from multi-core technology.

Heterogeneous CPU core architecture is also conceivable. AMD's Fusion technology is based on such a strategy. It permits GPU cores to be integrated in a CPU package, allowing graphics cards to be unburdened or even dispensed with altogether.

Some examples of future cores are:

- DSP (Digital Signal Processing) cores which can be used for real-time audio-video processing.
- FPU (Floating Point Unit) cores for computational-intensive tasks. These may be modelled on CBE SPEs.
- TCP/IP offload engines, which unburden CPU cores by optimising throughput in high-speed networking, in particular iSCSI implementations.

In addition, certain cores may function as a "sea-of-gates", an architecture used in current FPGAs. Individual gates can be consolidated per software into a core, thereby achieving unprecedented flexibility. Current FPGAs are limited to 1 GHz clock speeds; however, optimising gate layout to fit the application can nevertheless result in very high performance.

1.1.10 Intel® Virtualization Technology

Intel Virtualization Technology, sometimes referred to by its code name "Vanderpool", is a term referring to virtualization functionality implemented within an Intel processor. Thanks to Intel® VT, virtualization software can be made more efficient and robust since central virtualization functions are now performed by hardware. This hardware-assisted approach to server virtualization results in greater speed and flexibility and overcomes the limitations of software-based approaches which require virtualized operating systems to be modified. Intel® VT allows, for example, an unmodified Windows® System to run as a guest operating system on a Linux server.

1.1.10.1 Server Virtualization

Why Virtualization?

Virtualization, particularly server virtualization, can create many distinct benefits for businesses. To achieve optimal performance and trouble-free use, the required hardware and software components should be complementary.

Server virtualization can help businesses cut costs in many areas by leveraging hardware utilisation and by lowering administration, hardware and operating overheads.

By isolating operating systems from their underlying hardware, server virtualization transforms the business IT environment into a more flexible IT infrastructure with less risk. Server virtualization allows the guest operating system to be considered as an independent and isolated object that can be transferred between servers, or be duplicated and quickly restored, allowing, for example, backups to be made with greater ease. This permits businesses to respond more quickly to new challenges and provides a platform which permits the creation of better evaluation and development environments, allowing greater flexibility in hardware configuration, as well as faster deployment of server systems and their applications. The result is increased business agility.

How Server Virtualization Works

Server virtualization solutions enable the host computer to have individually isolated partitions or containers, so-called virtual machines. In such solutions a software application layer usually provides the environment for virtual machines. Depending on system architecture, this layer may run directly on the hardware or as a software application under the operating system. This layer, usually referred to as a virtual machine monitor (VMM) or hypervisor, enables the hardware resources of a single computer to be shared among different operating systems. VMM software allows host system resources such as RAM, processors, I/O and DMA to be transparently shared among guest operating systems.

A VMM should meet the following criteria:

- Isolation: In sharing physical resources, virtual machines must be securely isolated from one another with regard to the following:
  - Data security and integrity: VMs should not have mutual read/write access to data.
  - Stability: A malfunctioning virtual machines; should not have adverse effects on other virtual machines, for example, it should not result in their instability or failure.
- Efficiency: Virtual machines should not hog resources. In practice this means that a significant portion of machine instructions should be executed without direct VMM control.

Developing virtualization software and related administrative tools for virtual machines is a challenging task in which many obstacles must be overcome if good performance is to be achieved.

A number of difficulties result from the inadequacies of the x86 processor. This "one computer, one operating system" platform was, of course, not designed with virtualization applications in mind. Hardware-assisted virtualization software became a reality only with the advent of the "virtualizing" VT processor generation in 2006.

Virtualization without hardware support

Hardware-based virtualization, traditionally used in the mainframe domain, has been employed for over forty years. However, thanks to ever more economical and higher-performance hardware along with the availability of powerful virtualization software, virtualization has become an affordable technology. Its benefits are making it indispensable in modern IT infrastructures.

Intel® VT addresses the inadequacies of older hardware architectures, especially the x86 architecture which had its roots in the desktop PC. Typically, desktop PCs require only modest processing power since they host only a single operating system.

However, the x86 processor architecture has a number of characteristics which impede the virtualization.

To date, virtualization is entirely software-implemented. This approach is by nature a compromise and either results in performance penalties or is difficult to implement, for example because of the need for modifying the guest OS kernel.

Where the challenges lie

x86 processors run in protected mode using a ring architecture (privilege levels), which permits the operating system kernel and system applications to operate in distinct, mutually isolated domains. For example, most kernel code executes with maximum privileges in ring 0, while user processes run with restricted access and privileges in ring 3.

In virtualized systems, to protect hypervisor processes from running in ring 0, virtual machines execute code in a ring other than ring 0. A typical hypervisor, for instance Xen™, follows the architecture depicted in Figure 1.

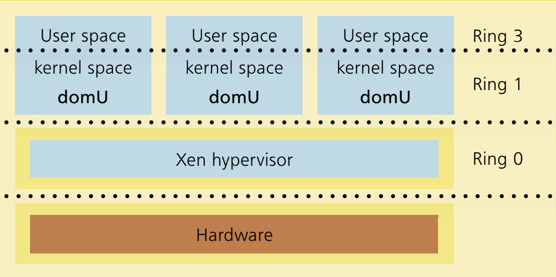


Figure 1: Ring architecture

The guest operating systems are run in ring 1 to prevent them from executing privileged instructions. If guests nevertheless try, as they inevitably will, to execute privileged commands, then the processor issues an exception (an internal error code), which can then be handled by the hypervisor. The hypervisor may, for example, choose to terminate the guest or to emulate the command.

From the standpoint of the memory management unit (MMU), which recognises only the two privilege levels user and system, both the guest, in ring 1, and the hypervisor, in ring 0, run with system privileges. In this scheme the guest may access memory belonging to the hypervisor. To protect against such memory access, many hypervisors use memory segments, setting their limits to prevent guest access to the hypervisor area.

Since the guests run with restricted privileges in ring 1, the hypervisor cannot execute all instructions without generating a processor exception. Such exceptions must be handled by the VMM and are CPU- and resource-intensive.

The x86 instruction set contains 17 instructions which require "creative" handling by the hypervisor. Among these are those not requiring ring 0 privileges. Such instructions are therefore executable by the guest, but they have a problematic behaviour. Some of the difficulties that may arise are:

- Instructions may fail to return the desired result (example: when not executed in ring 0, popf ignores the Interrupt flag)
- Instructions may return information differing from that of their privileged counterparts
- Instructions may return information that should not be available to user.

In such cases the hypervisor must take action commensurate with its role as final arbiter. VMware® (Server, Workstation, GSX), for example, filters out these instructions – a resource-intensive approach resulting in a performance degradation of up to 40%. The paravirtualization approach followed by Xen™ uses software interrupts – so-called hypercalls. While paravirtualization achieves better performance, it requires modification of the guest OS kernel.

1.1.10.2 Server virtualization architectures

The lack of hardware-side virtualization support has led to the development of a great variety of models and techniques for server virtualization over the years. Most software solutions depend on a "privileged" operating system which has control over hardware resources and distributes these among the guest operating systems.

A look at the virtualization software in most widespread use reveals three basic categories:

- **Full virtualization** through virtualization of a hardware environment. The VMM (virtual machine monitor) runs on an OS.
- **Virtualization of the operating system** with a single OS-image or OS-kernel ("single kernel image" or "SKI").
- **Paravirtualization:** The hypervisor (VMM) runs directly on the hardware.

Full virtualization

In full virtualization (often also referred to as "partitioning"), virtualisation is completely guest-OS-transparent. The virtual machine monitor (VMM) runs within the host-OS environment – usually as an application in user-space – and acts as an intermediary between the guest OS and its host.

The best-known representatives of this category are VMware® (Workstation, Server, GSX) and Microsoft® Virtual PC/Server.

OS-level virtualization

In OS-level virtualization several instances of one and the same OS (a single kernel, hence "single-kernel image" or "SKI") run virtually. Here it is the OS which is virtualized, not the hardware. The host OS creates several instances of itself by grouping user processes and applications in so-called "resource containers." All virtual machines are of exactly the same type and release version as the host OS.

The best-known examples of OS-level virtualization are found in OpenVZ/Virtuozzo, Linux Vserver, BSD jails and OpenSolaris zones.

Paravirtualization

Paravirtualization is a recent form of virtualization representing a compromise between SKI and full virtualization. Mutually-independent virtual machines are controlled by the hypervisor. They access shared hardware resources through their own OS via an API.

The most notable feature of this approach is the lack of guest-OS-transparency; here the guest operating system indicates its intent to the hypervisor through the interaction of a hardware abstraction level. Paravirtualization requires that the guest OS be modified.

The host OS comprises only a specialised kernel (hypervisor) and a privileged OS for management purposes.

XenTM is the best-known representative of this category.

1.1.10.3 Intel® VT

Intel® Virtualization Technology (VT) is an important technology in the mainstream server area and opens up possibilities for future desktop-virtualization. Virtualization software can be significantly optimised through hardware-assisted virtualization, which also offers increased robustness: Intel® VT enables VMMs to run guest OSs without modification and also increases the degree of isolation between virtual machines.

Intel® VT offers virtualization support for the following hardware components:

- CPU
- RAM
- I/O (based on IOMMU = I/O Memory Management Unit; in preparation)

Support for virtualization is accomplished by eliminating problematic instructions from the architecture's instruction set and through various extensions in processor architecture, including:

- An extended CPU instruction set
- A new exclusive processor operating mode for the VMM/hypervisor
- An interface (for transitions, handoffs) between the VMM and the guest OS via hardware-implemented methods
- RAM protection, in which the processor reserves RAM exclusively for each guest OS.

The new VMX processor generation includes hardware support for virtualization. VMX processors feature VMX root operation and VMX non-root operation. The former is a privilege-level reserved for the hypervisor. A schematic representation of this architecture is shown in Figure 2 below.

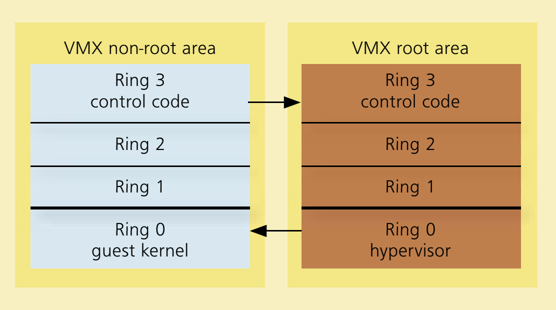


Figure 2: VMX root vs. VMX non-root operation

As can be seen from Figure 2, the new processors recognise three non-root area rings in which guest-OS code is executed and three root area rings of similar construction used for VMM operations. This architecture has new control structures which create individual address spaces for each guest OS and which help avoid conflicts between the guest OS and the hypervisor.

VMX root operation is intended for use by a VMM, and its behaviour is very similar to that of a normal processor without VT support. VMX non-root operation provides an alternative IA-32 environment controlled by a hypervisor.

Both areas support all four privilege levels, allowing guest-OSs to run at their intended privilege levels.

VT currently comprises:

- VT-x: IA-32 architecture, for example, Xeon® processors
- VT-i: Itanium® architectures
- VT-d: I/O virtualization

VT-x defines two new transitions: a transition from VMX root operation to VMX non-root operation called VM entry, and a transition from non-root to root operation called VM exit. A virtual-machine control structure (VMCS) is defined to manage VM entries and exits. The VMCS is divided into logical sections, two of which are the guest-state area and the host-state area. These areas contain fields corresponding to different components of processor state.

VMX mode is initiated by a call to VMXON. VM entries (VT instruction VMLAUNCH) loads processor status information from the guest area and control is transferred to the guest. VM exits (VT instruction VMRESUME) save processor status information in the guest area and load the state of the host area (see Figure 3). A call to VMXOFF causes the processor to return to non-root mode and control is returned to the VMM.

Briefly, guest systems run unmodified, isolated in the VMX non-root area, and in parallel with the well-protected VMM, while the VMM, via new instructions, can transfer control to or revoke control from virtual machines.

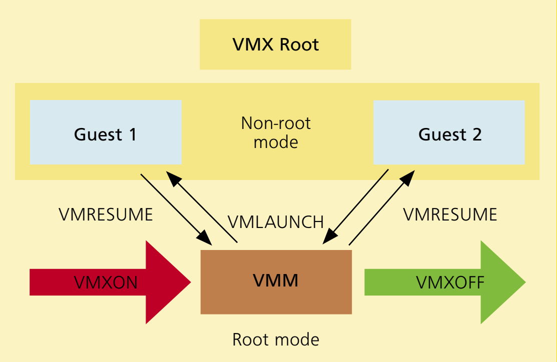


Figure 3: VMX mode

VT-i consists of extensions to the Itanium® processor hardware and the Processor Abstraction Layer (PAL) firmware. PAL interfaces can be used by the VMM (or hypervisor) to create and manage VMs. A Virtual Processor Descriptor (VPD) can be defined by the hypervisor to represent the resources of a virtual processor – logical processors, for example.

A new processor status register bit (PSR.vm) has been added to define a new processor operating mode. The VMM runs with this bit cleared while the guest OS runs with it set. If this bit is set then privileged instructions that may reveal the true operating state of the processor trigger virtualization faults, which can then be trapped and handled by the VMM.

VT-d (d for directed I/O) is an upcoming technology which complements existing VT solutions by hardware-implemented I/O-device virtualization.

The benefits of VT

The new approaches discussed here significantly improve server virtualization technology and offer a range of benefits, including:

- More secure operation thanks to smaller, simpler hypervisors
- Cost-reduction through the use of "normal" unmodified guest OSs
- Better security through the CPU's isolation of virtual machines
- Simplified (further) development of future hypervisor software.

Using VT

The following virtualization software is Intel® VT-ready:

- KVM: Linux kernel interface (interacts directly with the Intel® VT interface to run unmodified guests)
- Parallels Workstation
- TRANGO real-time embedded hypervisor
- VMware® Workstation 5.5 (Intel® VT is required for 64-bit guests)
- Virtual Iron 3.0 (a Xen™ derivative)
- Xen™ 3.x

The following software has been announced:

- Microsoft® Virtual PC 2007 (beta)
- Microsoft® Virtual Server (will support VT in 2007)
- VMware® ESX

About the author

Andrej Radonic is Managing Director of interSales AG Internet Commerce, Cologne. He has worked professionally with internet technologies, open-source software, server-based computing and server virtualization for the past twelve years. Radonic is the author of Xen 3 (Franzis' Verlag, October 2006), which provides in-depth coverage of open source approaches to virtualization from a practical perspective.

3. Clusters

3.5 Cluster- and grid-management systems

Reduced to its essentials, a cluster system consists of two types of computers: one or more servers and several clients which exchange data over a fast network connection.

The basis design of a cluster system is generally an asymmetric one in which the master node handles administrative work and receives tasks and parcels them out to clients. The consolidation of several heterogeneous cluster systems into a large, regionally dispersed system is called a grid. The name was coined to emphasise the analogy with an electric power grid – grid computing makes computing resources available in the same way in which a power grid provides electricity over a large geographic area.

Essential to the operation of a cluster system is a special software arrangement called a cluster management system. It is this software that allow users to submit their jobs to the cluster in a transparent manner.

Many providers of cluster management systems concentrate on programming user interface tools intended to simplify cluster use. Usually, such software runs only on hardware produced by the manufacturer and its use is subject to a licensing agreement.

In order to create a cluster system out of several computers, complex management software is frequently needed. User administration and adding and removing compute nodes are only two of the many uses of such software – it can also administer and carry out on a selective basis complex updating of software and configuration scripts. Management systems employed with cluster and grid systems are designed to make the complex inner workings and configuration of the cluster system transparent to the user.

The terms "cluster management system" and "grid management system" will now be defined more precisely. Both refer to software packages that provide the functions needed to managed cluster and grid systems. These functions are quite diverse and go far beyond user administration and capacity utilisation monitoring. Cluster management systems and grid management systems are responsible for:

- Installation
- Set-up/configuration
- Support of certain hardware
- System control and monitoring
- Maintenance and updating (hardware and software installations)
- User administration
- Allocation of relevant software (parallel-programming environments, resource management systems, etc.)

A number of powerful tools for administering computers have arisen from such humble beginnings as scripts and small utilities.

Many of these software packages are open-source. Some of this software is strongly orientated towards use in networks, since they are so ubiquitous in the world of commerce. These software packages contain a number of tools that are useful in managing cluster- and grid systems. In addition, there is management software specially designed for use in the cluster and grid computing area.

3.5.1 Primary functions of management systems

Management systems belong to the next development level of computing systems. While ad-hoc management is quite feasible In the case of small networks with only a few computers, in the case of large networks consisting of several hundred or even thousand computers it would require days to carry out the same tasks. Besides simple jobs such as the installation of new programmes there are more complex ones such as monitoring essential functions. While rack-mounted cluster systems have a definite advantage in terms of spatial separation over grid systems, this advantage is negated by the large number of computers involved.

There is no generally accepted minimum requirement for a management system. The type of support provided and which tools are desirable depend on the application at hand. Up to a certain degree it is possible to incorporate existing scripts and programmes as an integral part of the management system, allowing the management system to be adapted to the conditions prevailing at a specific site.

It is, however, possible to make a wish list of all tools currently available for managing clusters and grid systems. A survey of Linux-based systems presently sold on the market and on the Web is sufficient for this purpose. Table 1 below lists the features of current management systems that are relevant for the operation of a cluster or grid system.

Management features
Monitoring
Hardware inventory
Configuration management
Repository
Lights-Out-Management (LOM)
Health check
User administration

Parallel-programming environments

Resource management systems

Supported distributed file systems (cluster file systems)

Other selection criteria
Software licensing model
Support
Supported hardware
Supported operating systems

Table 1 – Management system functions.

3.5.1.1 Management functions

This section considers the essential management functions of cluster- and grid management solutions. These functions involve the analysis and evaluation of running tasks, as well the prevention of problems.

Monitoring

Monitoring is one of the essential operations for distributed computers. It provides administrators with information on storage utilisation, temperatures and other system parameters. Such information can be stored and evaluated, and used as the basis for corrective measures. Some of it can be made accessible to users, allowing them to choose the best time and place for submitting their jobs.

Hardware inventory

Hardware inventory is important mainly in large client-server networks. Clients return information on things like the type of processors used and their clock speeds, memory allocation, network configuration and the capacity of server hard drives. Such information could later serve as the basis for selecting the appropriate computer pool for running certain jobs.

Configuration management

Configuration management serves the function of restoring specific configurations. In the case of large distributed systems, configuration management is useful in installing identical software and configuration scripts on different groups of computers.

Repository

Servers with repository support can manage specific software packages. When the need arises, this software can be updated by a higher-level server. A scheduler is used to install software on clients at specified times. Software can also be deinstalled in a similar fashion.

Lights-Out-Management (LOM)

Lights-Out-Management features include hardware reset, rebooting and monitoring and readout of temperature, fan RPM, and other sensors. Of course, the software functions must be tailored to the corresponding hardware. One of most widely used management standards is IPMI (INTEL® Intelligent Platform Management Interface). A special Out-Of-Band chip is required to implement hardware reset and rebooting. This intelligent microcontroller (Baseboard Management Controller, BMC for short), which collects system information and is capable of reacting to events, lies at the heart of LOM and is responsible for the “well-being” of the system.

Health checking

Health checking involves the periodic monitoring of the status of services which have been registered in the health-check module. The health check function can also undertake the monitoring of critical resource metrics such as processor utilisation and disk capacity and issue warning messages if predefined thresholds are exceeded.

User administration

User administration plays a central role in large clusters and grid systems. Since rights are user-specific, manually managing user accounts for all computers in a cluster system would be unduly time consuming.

3.5.1.2 Parallel programming environments

Parallel programming environments form an essential element of cluster and grid systems. Process parallelisation involves the simultaneous execution of tasks on multiple CPUs. Parallelisation can be built into software by its programmer by distributing a process over several threads. Otherwise parallelisation is implemented automatically by the creation of causally independent (parallel) processes. Parallelisation can be implemented by the compiler.

The success of cluster systems has been influenced by the development of message passing libraries. PVM and MPI are the most successful of these.

Many cluster management systems come with one or more out-of-the-box parallel programming environments, allowing an easy, trouble-free implementation of parallel programming.

3.5.1.3 Resource management systems

Just as in the case of a public transport system, low utilisation of cluster or grid systems has a negative impact on cost efficiency. The ultimate goal of resource management systems such as schedulers and batch queuing systems is to achieve 100 percent utilisation.

In combination with a parallel programming environment, a resource management system can create a cluster out of a computer network.

Resource management systems differ in terms of their hierarchies and the degree of granularity supported. Most system have linear hierarchies in which the master node is the final arbiter, receiving tasks and when required parcelling them out compute nodes. This is distinguished from a flat hierarchy where each node is allowed to decide if, and to which other nodes, it will distribute tasks.

A process can consist of one or more daughter processes known as threads. A scheduler with a given granularity has the responsibility of spreading individual threads or even entire processes.

The three most important resource management systems are:

- OpenPBS
- Sun Grid Engine (SGE)
- Globus

3.5.1.4 Distributed cluster file systems

Distributed cluster file systems allow the consolidation of a collection of physically dispersed file system resources into a single logical structure transparent to users. Data on such a file structure appears to compute nodes as local data.

Cluster file systems provide cluster nodes with concurrent access to shared storage. They are frequently used in high-availability clustering environments to provide and manage access to shared storage. In high-performance cluster systems such as those used at CERN, the world's largest particle physics laboratory, distributed file systems allow the rapid transmission of high volumes of data.

If fast access and high bandwidth are of overriding importance, hardware solutions can provide the answer. NAS (Network Attached Storage) is such a hardware technique. It offers good performance through the use of a dedicated file server which makes network storage available to clients. A more complex solution involves the use of a SAN (Storage Area Network), which provides block-level access to network storage resources. Both NAS and SAN solutions come with rather high price tags. Software solutions are a more economical alternative.

3.5.2 More considerations

The choice of a cluster management system can be influenced by further considerations such as licensing models and the availability and type of support. Both can have a strong influence on the total cost of ownership.

3.5.2.1 Software licensing models

The software licensing models under which software is marketed or distributed are decided upon by the author or manufacturer and differ according to whether the software is open-source or commercial. There are specific licensing models for cluster management systems, again depending on whether they are commercial or open source. In the former case the licensing is done on a per-node basis with a separate additional license for the master node.

3.5.2.2 Support

The availability and type of support for a cluster- or grid system should be given careful attention. A small fraction of the Top 500 list of the world's fastest supercomputers are described as "self-made" and it can be assumed that they are maintained and managed by a dedicated team of administrators. Only in exceptional cases involving smaller systems is hardware support carried out by a local team – normally both hardware support and administration is provided by the same company that is responsible for managing system software. Before procurement it must be decided which of the above support alternatives is more economical over the life span of the system.

3.5.2.3 Supported operating systems (master/clients)

The choice of a management system often hinges on the type of operating system support offered. Support issues are far more relevant in the case of management systems for client networks than for cluster- and grid systems. In most cases the operating system is selected according to its suitability for specific tasks.

3.5.1.5 Supported hardware

Even the best cluster- and grid management software is worthless if it does not support the installed hardware. In the case of single-source management solutions the type of hardware has a decisive influence on both the ability to implement a system and the overall costs.

Networking technology is just as important as processor architecture, especially for cluster applications where network latency is often a source of bottlenecks. For this reason, many distributed computing systems employ advanced, low-latency technology.

In addition to opening up a number of new possibilities, developments in supercomputer technology, ranging from SMP (Symmetric Multiprocessing) to clustering, have also led to some complications. For example, while in the case of an SMP system all processors can access system memory over a fast bus, for cluster systems this is done over the network with its own specific protocols.

The creation of a virtual SMP system, one of the basic techniques of today’s cluster technology, is hampered by a critical bottleneck: the cluster system network, which assumes the functions of the bus used in SMP systems. The disadvantages of this are higher network protocol overhead and network latency. The more computers there are in a virtual SMP system, the greater the discrepancy between the ideal and realised boost in computing power. This divergence is the result of internal cluster communication. Here the decisive factor is latency; bandwidth plays a subordinate role.

In current network technologies efforts are made to achieve low latency so as to achieve the best possible system scalability (speedup). While throughput is of secondary importance for optimising speedup, it is important in many applications.

Network technologies in current use are:

- Ethernet (10/100/1000/10.000 MBit/s)
- InfiniBand
- Myrinet
- Quadrics
- SCI Dolphin

3.5.3 Conclusion

The commercial products s.cluster with scVENUS, Scali Manage and CSM have advantages in the areas of support and monolithic structure. In the final analysis, their cost must be added to the procurement cost, effectively lowering the performance/price ratio.

The table below contains a short summary of presently available cluster management systems.

Management system	Summary description/ distinctive features
CLUTO	CLUTO is versatile but requires specialised knowledge for installation. It ability to be adapted to various platforms and architectures permits the use of a broad range of middleware and applications.
OSCAR	This management package from the Open Cluster Group delivers what it promises: easy installation and configuration of Linux clusters.
Rocks Linux	Rocks provides an elegant approach for managing cluster hardware and software installation and configuration with Rolls. However, its extremely monolithic design can impede or even thwart installation.
s.cluster/ scVENUS	The commercial products s.cluster and scVENUS provide a variety of tools for managing network computers. ScVENUS is a ideal solution for organisations that have a large number of heterogeneous computers on a network or in a cluster.
Scali Manage	Scali Manage is available from transtec as an option. The integration of Scali MPI Connect in Scali Manage provides an MPI implementation with commercial support. Ditto for batch queuing systems with Scali Manage with integrated PBS.

Table 2 – Summary of evaluated cluster and grid management systems.

By way of summary:

- A transparent cluster installation and configuration procedure is easier and saves time; however, it complicates diagnostics and error remedying.
- Management systems built on top of a Linux distribution provide more versatility and better hardware independence.
- The choice of application software depends on the intended use and is also highly relevant for the choice of a management system. Commercial products have a mature software administration and simplify the installation and configuration of middleware and applications.

- The choice of the cluster management solution depends on cluster size and configuration. Heterogeneity management and architecture support can increase flexibility.
- The availability and type of support for cluster installation and management and for the set-up and running application software is a decisive factor affecting cost.
- The expenditures for the purchase or leasing of management software must be weighed against the personnel costs that accrue for an ad hoc management solution. The trend in most cases is that application costs outweigh the expenditures for procuring cluster hardware and management solutions.

Management solutions for cluster and grid systems can be classified according to applications into three types:

Type/Application of cluster or grid systems	Appropriate management system
Homogenous clusters with up to 64 nodes	CLUTO
Stand-alone user administration.	
Individual configuration desired	
Homogenous clusters with more than 64 nodes	Scali Manage
Advanced user administration	
Availability of commercial support with warranty	
Heterogeneous clusters with several (> 64) nodes	s.cluster with scVENUS
Integration of multiple platforms and architectures	
The option of managing additional computers over network	
Complex user administration Availability of commercial support with warranty	

Table 3 – Classification of cluster and grid management solutions according to application.

5. Hard disks and RAIDs

5.7 Perpendicular Magnetic Recording Technology

5.7.1 Introduction

An exciting new magnetic recording technology has been introduced into hard drive storage. Perpendicular magnetic recording (PMR) offers the customer higher capacities, improved reliability and robustness, and a very positive outlook for future growth in capacity and performance. Drive development incorporated a year-long field test that successfully demonstrated the viability of this new advanced design. Core technologies include second-generation trailing-shield heads and advanced granular cobalt-chromium-platinum (CoCrPt) oxide media. Jointly optimised head and media designs, as well as advanced system integration, resulted in very high performance with sharp write field gradients, resistance to stray fields, and excellent media magnetic and mechanical stability.

5.7.1.1 Hard Disk Drive (HDD) Storage Basics

All HDDs store the data as tiny areas of either positive or negative magnetisation on the surfaces of the disks. Each tiny area represents a ‘bit’ of information. The bits are written closely-spaced to form circular ‘tracks’ on the rotating disk surface. Many such concentric tracks cover the surfaces of the disks. There are millions of bits on each track and many tens of thousands of tracks on each disk surface. The total storage capacity of a HDD depends directly on how small we can make the area needed to represent one bit of information: the smaller the bits, the greater the capacity.

5.7.1.2 Areal Density, Technology Growth, Thermal Limit

The product of bits per inch along the track, multiplied by tracks per inch radially on the disk equals areal density in bits per square inch. Areal density growth-rate is a frequently quoted measure of the rate of advance of the technology. In recent years the growth-rate has slowed because of a fundamental limit in magnetic recording. This limit relates to the fact that the magnetic material on the disk surface is necessarily composed of small grains. Because of the randomness of the grain shapes and sizes, each bit written on the disk must cover about 100 grains to ensure that the information is reliably stored. Unfortunately there is a lower limit to the size of a grain. Below this limit, there is a risk that the magnetisation may spontaneously reverse just due to excitation by the thermal energy that is universally present in the environment, even at room temperature.

5.7.1.3 Perpendicular Recording Technology

Perpendicular recording addresses this ”thermal“ limit and allows continued advances in areal density. In conventional ”longitudinal“ magnetic recording (LMR), the magnetization in the bits is directed circumferentially along the track direction. In perpendicular recording, the ‘magnetic bits’ point up or down perpendicular to the disk surface. Figure 1 contrasts how the recording media, the write head, and the read head are configured for a longitudinal and for a perpendicular recording system.

The unique feature seen in the perpendicular system is the ”soft magnetic underlayer” incorporated into the disk. This underlayer conducts magnetic flux very readily. When the write head is energised, flux concentrates under the small pole-tip and generates an intense magnetic field in the short gap between the pole-tip and the soft underlayer. The recording layer that stores the data is directly in this gap where the field is most intense. Higher fields allow ”higher coercivity“ media to be used. Such media require higher fields to set the magnetization, but once set, the magnetization is inherently more stable.

The presence of the soft underlayer also strengthens the readback signals and helps decrease interference from adjacent tracks. Although the read head itself does not need to be changed very much, the waveforms that come out of the head are totally different and require new signal processing techniques in order to gain the most benefit.

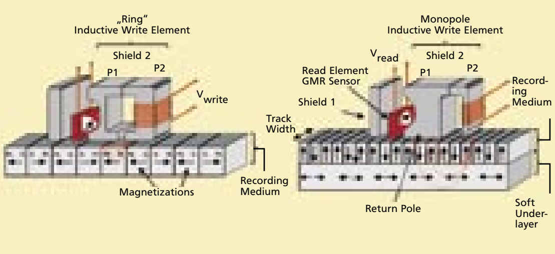


Figure 1: Longitudinal recording diagram (top) and perpendicular recording diagram (bottom).

5.7.1.4 Second-Generation Perpendicular Technology

Figure 1 refers to what we might call first-generation technology. While this technology proved advantageous and reliable, even more benefit can be gained by further refinements to the head, media, and electronics. This is second-generation technology (shipped in the Hitachi Travelstar 5K160). Second-generation technology involved changes to the write head, the recording medium, and the read/write electronics.

The write-head is modified by placing a ”trailing-shield” spaced closely to the trailing-edge of the pole-tip where the data is recorded, as in Figure 2. This can impact the field-strength slightly but has a big advantage in that the fields die away very rapidly as the medium moves from under the pole-tip to under the shield. This rapid gradient in field means that the bits that are written can be much more sharply defined.

For ease of implementation, the first-generation media was created as a single uniform layer. However, it is very advantageous to tailor the properties differently through the thickness of the media. These are properties such as the magnetic moment (magnetization per unit volume), the anisotropy (the strength with which the magnetization likes to align along a given direction), and the exchange (the level of atomic coupling between adjacent grains that tends to make the magnetization of adjacent grains point the same direction). These magnetic properties are a complex function of the materials used and the conditions under which the media is laid down.

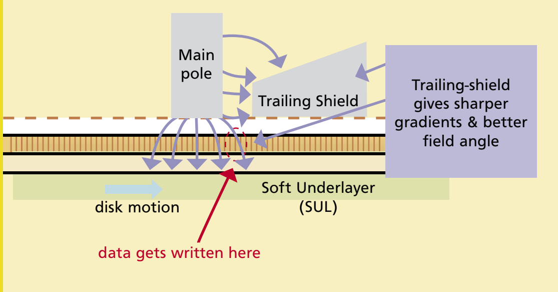


Figure 2: Cross-sectional schematic of a ”trailing shield“ head – a thin magnetic shield placed in the proximity of the trailing edge of the writing pole.

5.7.2 Head Technologies

In a conventional LMR head, the magnetic field for writing is generated from a thin non-magnetic gap in the ring head, and it has a higher longitudinal component than perpendicular component. In PMR, the media has magnetization orientated in the up-down direction. To achieve efficient writing, a PMR write head needs to generate fields having the perpendicular components higher than the longitudinal components. As illustrated in Fig. 1, a "single pole" PMR head combined with a soft-under-layer (SUL) offers a strong perpendicular write field, while the longitudinal component is much reduced. Rather than being generated from the gap, the field from a PMR write-head is generated from the pole surface and collected by the SUL. Fig. 3 shows that the corners of a rectangular pole will cut into neighbouring tracks when the head is operating at a skew angle. In modern drives, the head has a skew angle with respect to the track direction when the head operates at inner or outer radii. Fabricating a narrow trapezoidal pole with a well-controlled bevel angle is essential to prevent the fields from the pole surface erasing data in neighbouring tracks.

The Hitachi second-generation PMR "trailing-shield" head has tight controls on the shield thickness and the gap between the trailing edge of the pole to intricately balance the interaction between the trailing shield and the main pole. Media matching for trailing shield heads is critical in order to take advantage of the high field gradient and more optimal field angle, and to tailor to the modified field strength. When writing on media with matching characteristics, trailing-shield heads write sharper bits. As a result, the drive delivers better bit-error rate performance and therefore a better reliability margin.

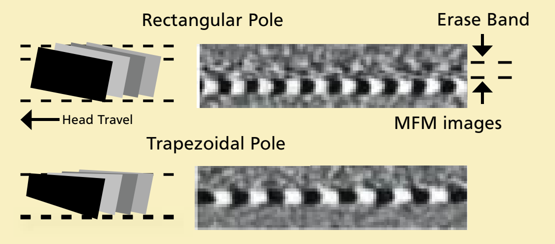


Figure 3: Trapezoidal pole shape in a PMR write head to avoid data erasure in neighbouring tracks when the head is operated at a skew angle, e.g. at inner or outer radius.

Besides the optimisation of the head design for recording performance, there are other challenges that are also unique to perpendicular heads, such as "pole erasure" and "stray field erasure". Both issues can lead to data corruption if not treated properly. Pole erasure refers to the phenomenon where the writing pole continues to emit magnetic fields even when the write current is set to zero at the end of the write cycle. As a result, the data can be erased unintentionally. This phenomenon arises from the extremely small dimensions and the ferromagnetic nature of the perpendicular writing poles. Special efforts in thin film magnetic material selection and processes are necessary to ensure a pole-erasure free write head.

One of the earliest concerns with PMR was an inherent increased susceptibility to external magnetic fields when compared with longitudinal technology. The increased sensitivity to stray fields originated from the interaction between the recording head and SUL. These external fields are a particular concern for mobile products, where, for example, a magnetic bracelet on someone's wrist can easily come within a few centimetres of the HDD in a laptop. Without special head designs, the external fields can greatly distort data writing and read-back signals and cause error events. In some extreme cases, the external field can even cause unrecoverable data erasure. With careful head and media designs, Hitachi has been able to bring the robustness of external fields up to a level equivalent to or better than current longitudinal drives.

5.7.3 Media Technology

As longitudinal media approaches its lower limit for thermally stable bit size, the industry has been motivated to resolve the historically complex issues surrounding perpendicular media manufacturing. The fundamental media structure developed for 2006 products is a type of "granular" media, comprised of magnetic alloys containing Cobalt, Chromium, and Platinum (CoCrPt) and an oxide grain-boundary segregant, as shown in Figure 4. By using a Hitachi-unique alloy combination and layer-deposition process, recording properties are graded through the media thickness to optimise signal-to-noise ratio while providing excellent writing characteristics and high mechanical quality. The media and head were co-developed to take advantage of features inherent in the trailing-shield write head design, the media soft magnetic underlayers, and the media hard magnetic layers.

Before product development started in earnest, perpendicular media had lower mechanical reliability than longitudinal media. It took a concerted and co-ordinated effort by Hitachi's vast resources to understand and improve the reliability to the level of Hitachi's quality standards, while also advancing the magnetic performance and achieving production-level cycle times and yield. Perpendicular media development required a rethinking of reliability metrics and anticipation of new potential failure mechanisms to help ensure the highest levels of corrosion resistance and mechanical robustness. Furthermore, manufacturing of PMR media technology required a new sputter tooling paradigm with more deposition stations, more process capabilities and higher throughput.

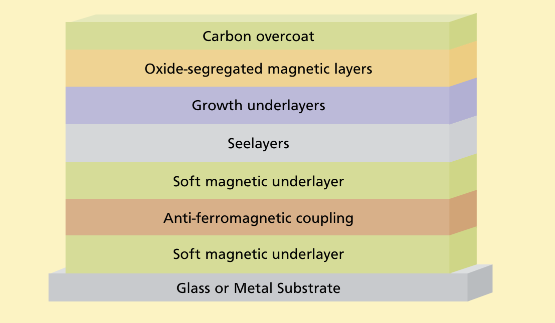


Figure 4: Hitachi Perpendicular Media Structure

5.7.4 Read/Write Electronics

The signals from a perpendicular recording system look dramatically different than those from a conventional longitudinal system (Figure 5). Every frequency component gets shifted by 90 degrees in phase (corresponding to the 90 degree rotation of magnetization from longitudinal to perpendicular). This totally alters the appearance of the waveforms. The signal processing in the Read/Write electronics must be modified to accommodate these waveforms. In addition to the phase-shift, there is also a lot more signal energy at low frequencies.

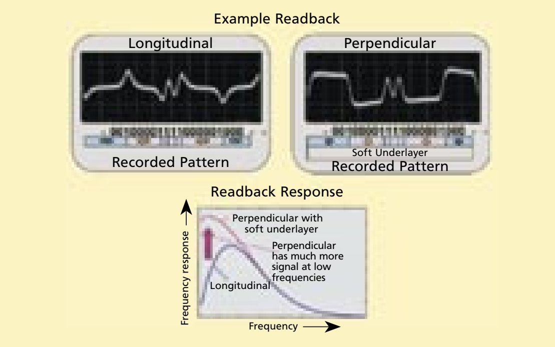


Figure 5: The readback waveforms change dramatically between longitudinal and perpendicular recording. The signal processing in the R/W channel must be able to accommodate these new waveforms and appropriately include some of the extra signal energy available at low frequencies.

Source: *Hitachi Global Storage Technologies*