Evolving your HP-UX environments from PA-RISC to the Intel® Itanium® architecture

Technical white paper

Table of contents

June 2003

Executive summary

Why should your enterprise transition from PA-RISC to the Intel® Itanium® architecture?

There are many strong reasons to move your most demanding workloads to Itanium-based systems now. For many application implementations, HP servers with Intel Itanium 2 processors offer the best performance, scalability, and investment protection available today. The Intel Itanium 2 processor has the best 64-bit processor performance on the market today—as measured by the SPECint2000, SPECfp2000, and LINPACK benchmarks.

The 2- and 4-way Itanium 2-based servers have superior performance to every equivalent SMP RISC-based server. This comparison is based upon system-level performance as measured by system bus bandwidth, memory bus system speed/width, total I/O bandwidth, and SPECwebSSL. It is also based upon systems-level performance, as measured by SPECint2000, SPECfp2000, SPECintRATEbase, and SPECfpRATEbase. Performance of Itanium 2-based systems exceeds PA-8700 performance by as much as 48%. For customers looking to implement a 2- or 4-way HP-UX system, Itanium 2-based servers have superior performance and headroom over *all* PA-8700–based 2- and 4-way servers. HP continues to invest in HP-UX compilers on Itanium-based systems, which are improving application performance by 36% between 2002 and 2003 based upon compiler performance alone.

HP provides an excellent array of tools, guidance, and services to assist our customers in making a smooth transition to the Intel Itanium architecture (see www.hp.com/go/Itanium). Areas for your consideration include:

- HP/Intel Solution Centers, run by HP and Intel, to assist with testing and migration
- Software engineering services, offering experienced HP resources
- HP Support services
- HP Education, offering many levels of training
- Tools available in the HP-UX Software Transition Kit (STK)

HP binary compatibility

HP customers enjoy the benefit of excellent HP-UX binary compatibility between PA-RISC and Itanium-based systems. The HP strategy for binary compatibility of HP-UX applications between PA-RISC–based and Itanium-based systems is to offer customers choices when moving 32- and 64-bit HP-UX applications to Itanium-based systems, as follows:

On Itanium-based systems, well-behaved[1] PA-RISC applications can run in "emulation mode," via **dynamic code translation**, at up to 90% native PA-RISC performance. HP's Aries translator dynamically translates code and runs these natively with little degradation. The benefit this provides is that **an application can be taken off a PA-RISC server and run directly on an Itanium-based system without alteration**. Because this software translation is transparently invoked and performance is close to native levels, customers may not always realize that an application is running in emulation mode, which is a very advantageous for customers with PA-RISC HP 9000 servers.

- An application can be **recertified using the HP-UX Software Transition Kit (STK),** available online. This is a relatively simple process that involves scanning the source code for binary exceptions and altering the code by recompilation on PA-RISC so that the new executable runs on both the source and target platforms. While not necessarily gaining full native-level performance for an application, results obtained in this way will be fine for a large number of customers running applications that are not performance critical.
- Ultimately, almost all applications should be recompiled to take maximum advantage of the Intel Itanium architecture. Customers can recompile their applications easily with their HP-UX STK. In addition, HP provides tools, guidance, services, and porting centers to customers and ISVs who want to do **recompilation**.

[1] A well-behaved application is one that does not make specific assumptions about the machine it is operating on, does not use undocumented features, and does not use non-portable extensions.

Prioritizing your transition steps to the Intel Itanium architecture

Frequently, data centers have hundreds of applications to manage on many different platforms. A data center's application mix is typically a stack of utility, core business, and performance-critical applications. Because HP-UX 11i on Intel Itanium processors was built from the same source code as HP-UX 11i on PA-RISC processors, customers can migrate their hardware to Itanium-based systems now, and then prioritize the recompilation of their software environment in parallel. An enterprise customer may divide their software environment into three distinct segments, as follows:

- **Utility applications**—These are applications such as data formatting and display, and print and file servers—often homegrown. This type of application makes up 80–90% of the overall applications in most environments. These applications just need to run—they don't need to be optimized right away. These are the applications that take immediate advantage of HP binary compatibility and software emulation. Transition of these applications to native operation can happen over time. In fact, just running these applications in emulation or having them recertified may be completely acceptable.
- **Core business applications**—These applications are critical to a business but are not performance-critical. They need to run correctly, but they can be optimized later. These applications can also take advantage of binary compatibility and software emulation today. They represent perhaps 5–10% of the applications being supported. Transition of these applications should occur as performance needs become critical. Examples include security applications, performance and monitoring agents, and applications associated with performance-critical applications.
- **Performance-critical applications**—These applications are part of the fundamental business processes and are mostly from third-party software providers such as Oracle®, BEA, and SAP. The key with these applications is to work closely with ISV partners and HP to plan the move to Itanium-based systems when the time is right. A majority of the most popular HP-UX applications are already available or will be available shortly. Consult your HP sales representative for details on the status of third-party applications that are available on Itanium.

Architecture and performance

The performance levels of Itanium-based systems have shown strong, steady increases over time. The performance of Itanium 2-based servers introduced in mid-2002 showed excellent improvement over the Itanium-based servers introduced in 2001. The servers introduced in mid-2003 show an equally outstanding performance improvement. In any discussion of Itanium processor performance, the first things that must be examined are the features of the processor.

Intel Itanium 2 processors have the largest on-chip cache of any merchant processor available in the market today, allowing applications with large instruction or data blocks to take advantage of fewer cache misses and benefit from reduced latencies. The Itanium 2 processor contains on-board L1, L2, and L3 caches, an improvement over the original Itanium processor introduced in 2001, which had only on-board L1 and L2 caches. Future iterations of the Itanium processor are expected to have even more on-chip memory. Small on-chip caches (L1) backed by slower off-chip (L2 or L3) caches may perform adequately for small benchmarks and applications. But when faced with demanding applications such as data warehousing, online transaction processing (OLTP), or technical computing, designs with off-chip caches result in poor performance. **That is one of the key reasons that larger caches were built into the Intel Itanium processor.**

The advantage that large on-chip caches bring to application performance is that when a processor is executing instructions or looking for data, it will first go to the closest location—in the L1 cache. If it does not find the instruction or data it's looking for there (a cache miss) it goes to the next level of cache—L2. This keeps occurring until the processor goes to L3 cache (if it exists), to RAM, and eventually to the disk drive looking for instructions or data. The problem is there is always a clock cycle delay during a cache miss, and that impacts performance. If the program goes to the L1 cache it's a one-clock-cycle delay, if it needs to go to L2 cache it's 5 cycles or more for Intel Itanium processors, 15 cycles or more for SPARC processors, and so on. By the time the processor needs to go out to disk to find an instruction or some data, the delay may be hundreds or thousands of clock cycles. At best, this impacts performance, and at worst it impacts application reliability. Applications can be written to take advantage of larger caches, which is what HP anticipates will happen now that the Itanium architecture is moving into full commercial adoption.

Another feature of the Intel Itanium architecture is the massive number of registers. This is directly reflected in the floating point and integer performance of Itanium processors. The large number of registers enables greater parallelism used in the Itanium processor when executing instructions. As can be seen, Itanium processors have a high number of issue ports, a large number of execution units, a high sustained instruction issue rate, and a short instruction pipeline length. All of these features are related to the greater parallelism built into the Intel Itanium Processor Family. This use of parallelism for performance gain reinforces the need for a strong family of compilers, backed up by an equally strong compiler development roadmap. Compiler development and maturity play a much bigger role with Itanium architecture than with other architectures, since compilers have a greater impact on application performance with Itanium.

- As might be noted, clock speed becomes less a measure of chip performance when instructional parallelism, maturing compilers, and large on-chip caches are part of the processor design.

Table 1. Performance

Processor type	Intel Itanium	Intel Itanium 2	Intel Itanium 2 6M (mid-2003)	HP PA-8700+	Sun UltraSPARC III Cu	IBM POWER4+
Operating systems	HP-UX, Windows®, Linux	HP-UX, Windows, Linux	HP-UX, Windows, Linux	HP-UX	Solaris v9	AIX 5L
Clock speed	800 MHz	1 GHz	1.5 GHz	875 MHz	1.7 GHz	1.45 GHz
L1 cache (1+ clock cycle)	32 KB	32 KB	32 KB	750 KB (inst.) 1500 KB (data)	32 KB (inst.) 64 KB (data)	32 KB (inst.) 64 KB (data)
L2 cache (5+ clock cycle latency)	96 KB	256 KB	256 KB+	N/A	8 MB (off-chip, 15+ clock cycle latency)	1.5 MB (shared)
L3 cache (12+ clock cycle latency)	4 MB (off-chip)	3 MB (on-chip)	6 MB (on-chip)	N/A	N/A	N/A
Issue ports	9	11	11	4	4	8
Execution units	9	11	11	4	4	8
Sustained issue rate	6	8	8	4	4	5
Pipeline depth	10	8	8	7	14	12
Registers	264 (128 general purpose, 128 floating point, 72 specific function)	264	264	32	96	72

Performance	Intel Itanium	Intel Itanium 2	Intel Itanium 2 6M (mid-2003)	HP PA-8700+	Sun UltraSPARC III Cu	IBM POWER4+
SPECint2000 (Base)	379	810	1,000+	642	537 (1.05 GHz)	909 (1 CPU)
SPECfp2000 (Base)	715	1,427	1,600+	464	701 (1.05 GHz)	1,221 (1 CPU)
SpecINTRateBase	19	37	12.2	23.2	21.4	35.8
SpecFPRateBase	24	40	34	66	26.1	38.1

Note 1: The SPECint2000 and SPECfp2000 benchmarks can be found at www.spec.org. They represent integer and floating-point processor performance. SpecINTRateBase and SpecFPRateBase are throughput measurements for integer and floating-point applications (www.spec.org/cpu2000/).

Note 2: The benchmarks by IBM were run with 1 of 2 processors in a dual-core POWER4+ package. The dual-core package contains 1.5 MB of on-chip cache that is usually shared between the two processors. This benchmark reflects the corner case of having only one active processor using all of the shared memory. This would not typically be the case when handling actual workloads. Real commercial application performance is likely to be lower than Intel Itanium architecture performance.

Architectural improvements of Intel Itanium 2 processors over Itanium processors:

- Reduced latency—more L1, L2, L3 cache; L3 cache now on-chip
- Reduced execution path: greater issue rate, shorter pipeline
- Fully bypassed functional units
- Moreways to issue/execute per clock cycle

Branch prediction

Branch predication is the process of using instruction execution prediction to improve application performance. Zero-clock branch predication is a key benefit of the Intel Itanium architecture. The Itanium processor will execute multiple code branches, and as the predicted branch is taken the correct branch "guess" will be used, and the remaining unused paths discarded. The execution will continue, without a pause, along the predicted target branch ("0" cycle branch penalty). Other RISC architectures attempt to "guess" at which branch will be taken early in the process. If the correct branch is not guessed, a penalty is taken. Zero-clock branch predication indicates that the machine takes no extra clocks to handle the branch.

In the future, a new generation of applications will be written to take full advantage of the instructional parallelism of the Intel Itanium architecture.

Hardware investment protection

In the future, HP will implement the Intel Itanium processor broadly across its server product lines. For HP-UX customers, both PA-RISC-based HP 9000 servers and Itanium-based HP Integrity servers are currently available.

All current HP 9000 servers have in-chassis processor board upgrades to the Itanium processor. Also, HP will continue to produce HP 9000 servers for a number of years, to enable our customers to transition to HP Integrity servers at the time that's right for their business.

Ask your HP account manager if you have questions about evolution planning or HP's future product directions. He or she should be able to provide the most updated hardware roadmaps for both the Itanium-based HP Integrity servers and the HP 9000 PA-RISC server family.

HP Integrity server product information and related links are found at: www.hp.com/go/integrity

Planning and implementing a transition to the Intel Itanium architecture

A move from PA-RISC to the Intel Itanium architecture can be planned and implemented either by customers directly, by customers in conjunction with HP, or with the assistance of a partner. Information about HP planning services can be found at www.hp.com/products1/itanium/services/planning.html.

It makes sense for some customers to use the HP/Intel Solution Centers for their transition to the Itanium architecture. The HP/Intel Solution Centers are dedicated to assisting customers with proof-of-concept services as well as performance tuning, migration, benchmarking, and capacity planning. There are now three centers: one in Cupertino, California (co-located with the Capacity Planning Center); a second in Grenoble, France; and a third center in Shanghai, China. Each offers a complete Itanium 2-based platform data center environment with the latest in server, storage, and networking technologies across HP-UX, Linux, and Microsoft® Windows operating systems. For detailed data sheets, solution blueprints, and white papers describing their capabilities, visit the HP/Intel Solution Centers Web site at www.hpintelco.net.

For other customers, working with the HP Services will make the most sense. These teams offer customers the option of having assistance from HP in any or all aspects of software transition planning by offering expert engineering services to assist in the migration to the Intel Itanium architecture.

Taking a step-by-step approach with HP Services

To address the challenges and help reduce risk during the transition, HP Services professionals offer project-based porting and migration services with a fixed price, scope, and timeframe*—backed by the ability to rapidly mobilize resources for onsite or off-shore services delivery. This proven approach hinges on two key services:

- **Application survey**—To gain a clear understanding of a customer's current application environment, architecture, and functionality, HP submits a questionnaire to the customer organization that is designed to help HP get a comprehensive view of the current situation. The completed survey provides the basis for determining the desired future state and the alternative paths to be explored.

- **Porting and migration assessment**—The experts at HP Services then provide a detailed assessment of the customer's business applications, databases, data, and platforms. The information gathered is then shared through one or more workshops where the customer can take a closer look at business drivers and the impact of change. The outcome includes specifics on migration strategies, applications to be ported, an application integration plan, resources needed, gap analysis, and step-by-step deliverables.

*For more information, see www.hp.com/go/Itanium, or contact your local HP Services representative.

Proven methodology built on experience

HP has already performed more than 100 major migration projects involving millions of lines of code for customers with business-critical application environments. Every porting and migration project follows set steps to help ensure a successful delivery, according to requirements as defined in the detailed migration proposal. Tailored to each customer's unique needs, the steps include:

- Phase 1—planned and detailed assessment
- Phase 2—tools development and customization
- Phase 3—test migration
- Phase 4—application migration
- Phase 5—acceptance
- Phase 6—installation, warranty, and product support

Ask your HP Support representative for more information on HP migration services.

Taking advantage of the Intel Itanium architecture

Improved architectural features and compilers

The Intel Itanium architecture was designed to allow compilers explicit control over the execution resources of the processor, in order to maximize instruction-level parallelism (ILP). The advantage this brings to customers is that HP's investment in compilers means improved application performance independent of operating system or hardware improvements. Instruction-level parallelism is the concurrent execution of multiple instructions. The Intel Itanium Processor Family provides three key features that enable the compiler to maximize ILP: speculation, predication, and explicit parallelism. Customers moving their applications to HP-UX on Itanium-based systems should recompile their application and take advantage of ILP on the Itanium architecture.

The three architectural features that are relevant to application performance improvement are speculation, predication, and explicit parallelism. Speculation is the execution of an instruction or dependent instruction stream before it would normally be executed in the program order specified by the application developer.

Predication is the conditional execution of an instruction based on the setting of a Boolean (true or false) value contained in a predicate register. Without predication, parallelism would be impossible. Instead of waiting for each section of a complex calculation to finish, it is faster if the processor can predict the outcome and proceed on the basis of that prediction. These prediction points are called branches and current processors try to guess which branch to take. If it predicts correctly, the whole calculation is validated. If it predicts incorrectly, the string has to be thrown out and the calculation starts over. The Intel Itanium architecture provides 64 predicate registers that can be used to control the execution of nearly all instructions. Explicit parallelism allows the compiler to communicate dependence information to the hardware through explicit S bits (stops) between instruction groups.

Because the instruction-level parallelism on the Intel Itanium architecture is explicit, the role of the compiler is critically important in delivering application performance. Furthermore, assembly programming, already rendered complex by the introduction of RISC features such as delayed branching and exposed latencies, becomes even more challenging with the introduction of architectural features such as predication, speculation, and explicit parallelism.

Maturing compilers

The benefit of new optimization capabilities added with each compiler release varies depending on the application and the optimization options used to build it. Compilers built to use with the Intel Itanium architecture should see great performance gains over the next few years relative to the compilers built for older RISC-based architectures such as POWER, UltraSPARC, and PA-RISC. This will give Itanium-based performance significant improvements in performance as each succeeding processor and operating system iteration is released. For example, HP is expecting to see double-digit performance improvements in application performance over the next few years, based on the improvement in compilers alone.

More information on Intel Itanium compilers can be found in Carol Thompson's Interex paper at www.interex.org/pubcontent/enterprise/jan01/08thom.html.

Intel compiler Web site: www.intel.com/software/products/compilers/

Using Aries and the HP-UX 11i Software Transition Kit

The process for transitioning HP-UX applications from PA-RISC–based to Itanium-based systems can be found on the Software Transition Kit (STK) Web site at http://devrsrc1.external.hp.com/STK/index.html. All the information you need to take advantage of the HP-UX/Intel Itanium 2 architecture environment can be found there. Moving to the Itanium architecture can be done in one of two ways: **qualifying an application** or **porting an application**.

Qualifying a well-behaved application

This process provides a single executable that runs on both source and destination platforms. In a few cases, it may be necessary to do minor source-code changes and recompile on the source platform. Therefore, during the qualification process it scans only for binary compatibility impacts that might prevent source code from running on the destination platform.

Qualifying an application requires the following:

- Setting up your environment
- Finding binary compatibility impacts
- Finding impacts in scripts and makefiles
- Building and testing the application on the source platform
- Building and testing the application on the destination platform

If significant performance or binary compatibility impacts are discovered and fixing them causes the application to not run on either the source or destination platforms, the software cannot be qualified and must be ported instead.

Note: The term "well-behaved application" is used throughout the industry to describe, in general, an application or piece of code that is written to be transportable between hardware or operating system environments. Well-behaved applications do not make assumptions about run-time architecture, platform architecture, or object file format.

Porting an application

Porting an application involves recompiling it to run with the best possible performance on the target platform. The duration of this activity will depend upon the size of the code being moved and the amount of testing planned.

To port the application:

- Set up the user environment as described in Configuring the HP-UX STK Tools
- Scan the source files using scansummary and scandetail
- Resolve impacts in the source files
- Set up the new development and build environments
- Upgrade the compiler (if required)
- Build and test the application on the source platform
- Build and test the application on the destination platform

Operating in a mixed system environment

As enterprises begin to more fully integrate Itanium-based systems into their IT environments, customers will find that **most, perhaps all, of their PA-RISC applications will function correctly on Itanium-based platforms without recompilation**. This is primarily due to the great source code compatibility between HP-UX on Itanium-based systems

and PA-RISC–based systems. Customers may also find that some of their code is operating natively on the Intel Itanium architecture, and some is operating in translation mode (described later). HP has built its Aries dynamic translator with quality in mind, so customers should be confident that they can transition their hardware environment to the Itanium architecture now, and then transition their software environment at their own pace.

Binary translation on Itanium-based systems

The HP PA-RISC software emulator, **Aries**, is shipped with every version of HP-UX 11i for Itanium-based systems. Designed and created by HP, Aries is a **dynamic software translator that allows PA-RISC–based code to operate on Itanium-based systems**. Because it is invoked transparently when a PA-RISC binary is run on an Itanium-based server, it is important to understand its features and limitations. The basic operation of Aries is that, as it is invoked, **it translates the PA-RISC-based code into Itanium-based code on the fly**. If certain sections of code are repeated, Aries will translate and keep the repeated code for usage. What that means is that if code is being translated and executed on the fly, and reaches a point in the execution where a repeated code block exists in memory that was previously translated, Aries will understand this and execute the repeated code block as if it were native Itanium instructions. One can imagine that over time, as an application operates in binary translation, many repeated blocks of code will be saved and then used during execution. Thus, customers may experience binary translation that actually improves during execution time and approaches native performance levels.

(At the end of the execution process, the translated code is discarded).

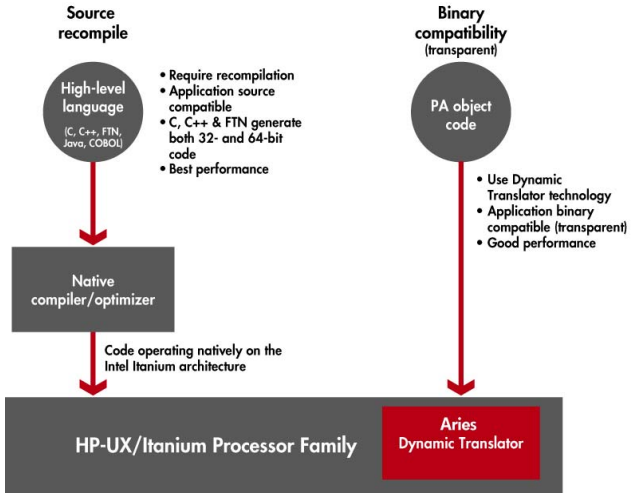
The major components of Aries are:

- The interpreter (flem), which emulates each RISC instruction
- The translator (dynt), which translates RISC instructions into 64-bit instructions
- The system call manager (scm), which emulates all the system calls made by the RISC application
- The exception manager (exc), which deals with all issues related to signals, such as setting up signal handlers, receiving signals, etc.
- The low- and high-level control systems, which handle the control flow across various components of Aries

As customers build software environments using Itanium-based systems, the ability to use a dynamic code translator to run applications during a period of transition is important, since not every application may be available to operate natively on the Intel Itanium architecture. The great advantage of Aries is that it allows you to use your Itanium-based hardware while doing a parallel software move to run natively on Itanium-based systems.

A block diagram of how native and translated applications operate together is shown in Figure 1.

Figure 1. HP-UX/Itanium



The Aries software translator allows PA-RISC–based HP-UX binaries to run transparently on Itanium-based systems.

Table 2. Important features of Aries

Feature	Description
Transparent operation	The HP-UX kernel automatically invokes Aries when an HP-UX/PA-RISC binary is launched. The invocation and usage model for HP-UX/PA-RISC applications on Itanium-based systems running HP-UX remains unchanged.
Hardware levels of reliability	Aries achieves hardware levels of reliability. This is the result of more than three years of HP investments in verification and validation methodologies for Aries.
Support for 32-bit and 64-bit addressing models	Aries fully supports applications built for the 32-bit and 64-bit addressing models. Aries is embodied as two separate shared libraries: one library emulates 32-bit applications and the other emulates 64-bit applications. The HP-UX kernel automatically invokes the appropriate shared library.
Built-in supportability	Aries makes it possible to debug a PA-RISC application that is running on Itanium-based systems. If such an application fails and generates a core file, Aries will generate a PA-RISC–style corefile, and is debuggable using gdb.
Reasonable performance	Aries offers levels of performance to make it a viable compatibility solution. Aries' performance varies depending on the nature of applications. For user-interactive/system-intensive applications, Aries performance is comparable to the native PA-RISC run of the applications. For compute-intensive applications, Aries performs at a very reasonable level: integer applications slow down by a factor of 2–4 and floating-point applications slow down by a factor of 3–6.

While the Aries is extremely robust and versatile, the complexity of the software interaction in many customer environments is extreme. Certain restrictions to the operating software translator need to be understood:

- Aries does not support emulation of privileged PA-RISC hardware instructions. Privileged instructions are used in operating system code to perform certain system management operations. This limitation will not affect user applications.
- Aries does not support HP-UX applications on Itanium-based systems that attempt to use HP-UX/PA-RISC shared libraries, or vice versa. Aries only supports pure HP-UX/PA-RISC applications. A pure HP-UX/PA-RISC application is one that consists of only HP-UX/PA-RISC libraries (archived or shared). This warning is for precautionary purposes. A PA-RISC-based application already written to correctly operate on PA-RISC hardware would not invoke any Intel Itanium Processor Family shared libraries. It would only be as a result of inconsistent coding techniques that a new PA-RISC or Itanium-based system application would attempt to use a cross-architecture shared library. If you are still unsure, it is advisable to run the following command:

  chatr library >> $PLAN_DIR/stk.libs

  It will determine the shared libraries used by the executable. (See "determining library dependencies" http://devrsrc1.external.hp.com/STK/libscan.html.)

- Aries may not support some implementations of ttrace(), ptrace(), or profil(). Aries emulates the ttrace system call in order to enable pa-gdb (the PA-RISC debugger) on Itanium-based systems (both 32- and 64-bit Aries). Aries does not support applications that use the ptrace() or profil() system calls. This limitation would impact only debugger programs, which are normally not portable between environments.
- Aries supports both fork() and vfork() system calls. However, Aries does not support applications that rely on differences between fork() and vfork(). Most applications that use the vfork() system call do use it with a purpose that is well known to the programmer. It is an extremely rare situation that a standard application would have any such reliance on the differences between fork() and vfork() calls.
- Aries does not support identical SigNaN and QNaN. Not-a-Number (NaN) is an IEEE floating-point bit pattern. Compaq Fortran identifies NaN values with the letters "NaN" in output statements. A NaN can be a signaling NaN or a quiet NaN. Note that this is an advanced issue that will only impact a limited number of floating-point implementations.
- The version of Aries that ships with HP-UX 11i v2.0 supports HP-UX PA-RISC applications that run on HP-UX 11i. Aries does not support applications compiled on HP-UX v8.x or earlier. However, such applications should work if they run on a contemporary version of HP-UX (that is, HP-UX 11i v2 and earlier).

Aries does not support privileged PA-RISC instructions. Hence, device drivers and loadable kernel modules are not supported.

- Aries consumes a small amount of an application's virtual memory address space. Therefore, Aries does not support applications that are nearly or completely out of virtual address space. Such applications, in practice, have been found to be extremely rare.
- When an emulated program makes any system call that returns the processor-related information, under emulation Aries returns information pertinent to a PA-RISC 2.0 processor even as the emulated application runs on an Itanium Processor Family system. For instance, a call made to sysconf(2) with SC_CPU_VERSION will return CPU_PA_RISC2_0. This is an Aries policy that an emulated program sees a complete PA-RISC environment on an Itanium-based system. If the application requires that it be able to determine that it is running on an Itanium-based system, then one method is for the application to use the system(3S) call and utilize the Itanium Processor Family–native command, getconf(1), to get the required fields API and ABI compatibility.

API and ABI compatibility

API compatibility

Rather than let applications access operating system resources directly, open systems such as UNIX® and Linux have a portability layer to which applications are coded. This is the source-code compatibility layer that all well-behaved applications should be written to. In general, this coding layer is the set of POSIX application programming interfaces (APIs).

There are UNIX-branded systems to ensure source-code compatibility for applications among UNIX systems, and there are LSB-branded systems to ensure binary compatibility for Linux applications. However, applications must be coded to use the APIs or application binary interfaces (ABIs) within the scope of the compatibility definition. Applications must not use private interfaces of the operating system or subvert the portability layer by accessing operating system resources directly.

ABI compatibility

An ABI is the interface through which an application gains access to the operating system upon which the application has been set up to execute. Most often the term ABI refers to a UNIX system ABI that is a specification of common header files, data structures, and system interfaces for UNIX and Linux implementations, which define binary compatibility for applications between compliant systems.

ABI compatibility is represented by the set of supported run-time interfaces available for applications to use (as opposed to the API, which is the set of build-time interfaces). The most important part of a system's ABI is the set of interfaces provided by its shared libraries.

Restricting development to the source API specification is not enough for binary compatibility because different releases and different systems have different versions of the libraries. To become binary compatible, you must develop to the ABI.

Validation of applications on Aries

Table 3. HP has verified the functionality of a large number of functions and applications on the PA-RISC dynamic code translator.

Category	Applications tested functionally on Aries (many of these applications may have native versions)	Native application examples
Web servers	Apache Web Server version 1.3.12 iPlanet Web Server Enterprise Edition version 4.1 sp2 Netscape Enterprise Server version 4.02/4.15 Squid Cache Server version 2.3.STABLE4 Netscape Proxy Server version 3.5x	Zeus Web server SeeBeyond e*gate
Middleware		BEA WebLogic Server 6.1, 7.0 BEA Tuxedo 6.5, 8.0 Borland Visibroker TIBCO Rendezvous
Tools, applications, and commands	xfig netglance echoping finder gnuplot gperf gzip/gunzip ls tar xtar Acrobat Reader xgas gslots	Rational Clearcase Service Control Manager for centralized management System Administration Manager (SAM) for HP-UX system administration HP-UX Kernel Configuration for self-optimizing kernel changes Ignite-UX for installation and deployment of the operating system Software Distributor-UX (SD-UX) for software and patch management System Inventory Manager for change and asset management Event Monitoring Service (EMS) for fault management Security Patch Check for protecting servers against new vulnerabilities Management Processor for powerful remote management over the Web
Development tools	softbench gcc	
Debugger	gdb	
Web browser	Netscape version 3.0, 4.7x, 6 Internet Explorer version 4, 5	
Search engines	Inktomi search software version 4.1.1	
File sharing servers	Samba (CIFS/9000 server-rev a.01.02 and client) version 2.0.7 Netscape Directory Server version 4	
Scientific applications	Many	Many, see http://isvweb.nsr.hp.com/ipf
Databases	Oracle version 8.x Sybase	Oracle9*i*
System administrator utilities	SD utilities (swinstall, swremove…) revision 5.1 SAM revision 73.2	
Multimedia	Real Server version 8 Real Player Basic version 8	
Performance tools	TOP 82.1.1.15 SAR 82.1.1.7 ITO OpenView node manager EMF HA Monitor SAS version 11e hpwebqos	Process Resource Manager (PRM) for workload management
Benchmarks	Alaska benchmark spec95, 2000 (int and fp) version 95, 2000 mth	
Shell	expect tcsh tcl hpterm	
Programming language	Perl (32, 64) version 5.x	

Java™	Java plug-ins	Java 1.3.1.8
Java applications	Forte version 4 JBuilder version 3.5 JProbe version 2.8.1 Together version 4.2 HPjmeter version 1.1.1	
Java benchmarks	CaffineMark version 3.0 SwingMark JVM version 98, 2000 Firehunter DiningPhilosopher JCK (with JVM 1.2.2.04H, 1.2.2.08, 1.3.0.00, 1.3.0.1) version 1.22, 1.3	SPECInt2000 SPECfp2000 SPECjAppServer2002 TPC-C SpecJBB
Editors	Emacs version 20.7.x XEmacs version 19.14 vi/vim version 5.5	
Security		RSA Security Bsafe Encryption

Early supporting information on the PA-RISC Dynamic Code Translator (Aries) can be found at:

www.cpus.hp.com/technical_references/ia_64_abstracts.shtml (paper abstracts)

Expected performance using dynamic translation

Once it is understood whether a given set of PA-RISC applications will function correctly in Aries, it is important to understand what the expected level of performance will be for the given application, especially if the application will be operating in a production environment. One thing is extremely important to understand: for many commercial applications, operation in translation may be good enough if performance is not a factor—recertification or recompilation may be unnecessary. In addition, for many customers operating on older PA-RISC hardware, running an application in "emulation mode" on an Itanium 2-based system may actually result in performance that is *better* than currently being experienced.

For customers with older PA-RISC systems, expect performance for PA-RISC applications operating on Itanium 2-based systems in binary translation mode to be in the range of PA-8600 to PA-8700 levels. For example, if you had one of the older HP 9000 N4000 systems (which use PA-8600 processors) and were operating a custom application on it, expect performance on an Itanium-based system in binary translation mode to be about the same, or better. For more detail on performance of applications in binary translation, see your HP sales representative.

Why is binary translation necessary? Most hardware architectures require some type of translation to enable binary compatibility of applications or to maintain performance levels between generations of processor implementations. The Sun and IBM architectures are no different across generations of processor technologies. In the case of the Itanium processor, the processor architecture is different from PA-RISC, but it was designed with HP-UX as its operating system. Therefore, as PA-RISC/HP-UX applications originally designed to work on PA-RISC begin to interact with the Intel Itanium microprocessor, they need to be given some "direction" on how to operate within the processor. Explanations of how dynamic code translation works can be found in the previous section. Once an application is recompiled, it can operate natively on a processor architecture and does not require any extra direction.

For system-intensive applications, such as Web server and GUI applications, applications with intensive disk I/O, and interactive applications, minimal performance degradation should be expected. Table 4 shows relative performance comparing an initial release Itanium-based server using dynamic code translation and native modes for the given application types. Performance for Itanium 2-based servers is expected to show the same relative performance.

Table 4. Aries performance

Application	Relative Aries performance on 733 MHz Itanium-based server (% of native performance)
Multimedia (SAM)	90%
Web browser (Netscape)	90%
Web server (Apache, Zeus)	85%
Application server (XEmacs)	85%

Compute-intensive applications, such as floating-point–intensive applications, applications with poor code locality, or applications that have undergone optimization using inlining, may have less optimal performance. Performance in binary translation mode will be, on average, close to that of a native PA-RISC application running on a PA-8600 or PA-8700 processor. It is recommended that if a compute-intensive application is part of a critical business process, it should be recompiled to run natively on Itanium-based systems. For more detail on running PA-RISC applications in binary translation mode, please contact your HP account representative.

System management and HP OpenView capabilities

HP server management tools are well integrated into HP OpenView and, when combined, result in the most manageable data center solutions available. The huge advantage that HP OpenView brings to existing PA-RISC customers looking to run Itanium-based systems in a mixed environment is that Itanium-based systems can be managed from an OpenView station the same way that HP 9000 systems are managed. This dramatically accelerates the integration of HP Integrity servers into your environment. The capabilities of OpenView on Itanium-based systems are growing very quickly; some examples are listed below.

- **OpenView Operations Agent**—collects and correlates OS and application events; allows Itanium-based servers to be managed from a central location
- **OpenView Performance Agent**—determines OS and application performance trends; provides performance management and tracking of Itanium-based servers
- **Glance**—shows real-time OS and application performance to diagnose problems
- **Data Protector (Omniback II)**—backs up and recovers data

In addition, the Network Node Manager (NNM) management station will run on Itanium-based servers running HP-UX. NNM automatically discovers, draws (maps), and monitors networks and the systems connected to them including Itanium-based servers from HP. All other OpenView management tools, including OpenView Operations, Service Desk, and Service Reporter, will be able to collect and process information from the agents running on Itanium-based servers that are running HP-UX.

The **importance of being able to incorporate Itanium-based systems into an OpenView environment cannot be overemphasized**. This is an outstanding customer benefit, and will accelerate adoption of the Intel Itanium architecture in many enterprises.

Intel Itanium architecture services and education

HP offers a Business Systems Evolution Web site that houses a growing array of reference resources, links, and information, including product upgrade and trade-in programs, presales assessments, financial options, training, and implementation services, to assist in planning a transition of your IT infrastructure to the Intel Itanium architecture (see www.hp.com/go/evolve.)

Porting and migration services

As discussed earlier, while many software programs run quite well in Itanium processor compatibility mode with minimal change, these applications run even better after being migrated to operate natively on HP Integrity servers. HP offers a flexible set of services to help customers through the optimization process. Select the degree of assistance you may need from the following:

- **Workshop:** a quick investigation of needs, followed by development of a high-level application migration plan
- **Consulting:** design services to architect a porting and migration strategy; an excellent choice for businesses trying to develop their strategy for how and when they will move to the Intel Itanium architecture
- **Detailed assessment:** an in-depth investigation that yields a detailed migration plan with specific recommendations. Not every application needs to be ported—HP can help a customer decide which ones to port and how to do it
- **Solution delivery:** HP works side by side with our customers to do the porting and migration of specific applications, including reengineering and integration with existing applications

Education and training

Effective training can be crucial in accelerating a transition to the Intel Itanium architecture. HP has tapped the knowledge of its Itanium processor experts in HP Labs, as well as in its partner base, to develop the best training available for this new technology. HP offers an online curriculum that covers overview and administration on all three operating systems: HP-UX, Windows, and Linux. HP can help customers make more informed decisions and move forward more quickly. Web-based classes include:

- Introduction to Itanium (H8361aae)
- New Features and Functions of HP-UX for Itanium (H8362aae)
- Linux on Itanium (H8363aae)
- Windows on Itanium (H8364aae)
- Migrating Applications to Itanium I (H8365aae)
- Migrating Applications to Itanium II (H8366aae)
- Software Functions and Algorithms on Itanium (H8367aae)

- HP-UX 11i on the Itanium 2 Architecture (U2391aae)
- Customizing Your Apache Web Server on HP-UX (H4291s)

Specific training for HP-UX 11i v1.6 can be found at:
http://cso.fc.hp.com/ssil/uxsk/hpux/hpux_releases/11.22/index.html

Further customer resources and collateral
HP offers collateral and documentation on transitioning HP-UX environments from PA-RISC–based to Itanium-based systems. Customer information can be found at the Business Systems Evolution Web site, www.hp.com/go/evolve.

HP-UX Software Transition Kit documents: http://devrsrc1.external.hp.com/STK/docs.html

Additional notes and resources
Software Licensing, Emulation, and Support (www.geocities.com/remklerosy1/index2.html): Questions have been raised that operating a third-party application in emulation mode might violate certain software licensing restrictions. While this is not intended to be a legal opinion, after review of current industry practices regarding enterprise software licensing, it is felt that, for the most part, customers should not have problems running applications in emulation mode on an Itanium-based system. From a support and troubleshooting perspective, however, each third-party software provider will handle their own application differently. It can be assumed that if a software problem occurs in emulation mode and the source of the problem is traced to third-party software, it is likely that troubleshooting support for the problem will be difficult to get. However, it is important in these situations that the customer read the licensing agreement closely and speak to an appropriate customer support representative.

5981-8773EN, 06/2003