Cellular IRIX™ 6.4 Technical Report

Table of Contents

Cellular IRIX 6.4 Technical Report

1.0 Introducing the C*ellular IRIX* Architecture and Cellular IRIX 6.4

Since the advent of the microprocessor, computer system users and designers have dreamed about being able to configure large powerful systems by connecting many smaller less powerful systems together. Ideally, a modular system would be immensely scalable and resilient, yet have the manageability, efficiency, and throughput associated with mainframes and supercomputers. Many vendors have tried to deliver this vision, but the resulting systems do not behave like a single system with common shared resources.

With the announcement of the *Cellular IRIX* architecture and the first systems based on the Origin2000 architecture, Silicon Graphics begins to deliver the dream. *Cellular IRIX* is Silicon Graphics' distributed software architecture for scalability and resilience. Cellular IRIX 6.4, the initial release of *Cellular IRIX*, introduces major scalability, performance, throughput, and availability enhancements to IRIX, the industry's richest UNIX® operating environment. The first software product from the merger of Cray Research and Silicon Graphics, Cellular IRIX 6.4 puts industry-leading supercomputing/datacenter software functionality on scalable cost-effective and high-performance Silicon Graphics hardware.

1.0.1 *The Cellular IRIX Architecture*

Silicon Graphics has been a leader in scalable UNIX systems from the day the company shipped its first symmetric multiprocessing systems in 1988.The company continues to lead with exceptional (NFSv2) LADDIS, TPC-C, and HIPPI throughput results on 64-bit CHALLENGE and POWER CHALLENGE servers running IRIX 5.3, 6.1, and 6.2.

Cellular IRIX supports modular computing, providing availability and throughput on small, one-to-four processor systems and scalability, performance, resilience, and throughput on large systems with tens or hundreds of processors and hundreds of gigabytes of memory. *Cellular IRIX* takes traditional symmetric multiprocessing systems to the next level of scalability, and marks a new era in computing: modular, scalable systems which behave like a highly-resilient mainframe or supercomputer because their operating system is both distributed and unified.

1.0.2 Cellular IRIX 6.4

The first release of *Cellular IRIX* immediately delivers on the promises of the software architecture. Cellular IRIX 6.4 is *scalable:* expandable from 1 to 32 processors (soon, to 128 processors). Cellular IRIX 6.4 is *compatible:* building on the features, functions, and interfaces of 64-bit IRIX 6.2. Cellular IRIX 6.4 provides *high performance* through innovative dynamic, explicit, and implicit data placement functionality. Cellular IRIX 6.4 provides *high throughput* via an advanced scheduler that insures fairness for interactive users, and via supercomputing software from Silicon Graphics and Cray Research.

Cellular IRIX 6.4 builds on the exceptional heritage of IRIX, one of the industry's first fully functional 64-bit UNIX implementations. With exceptional real-time capabilities, Cellular IRIX 6.4 presents virtually unmatched NFSv2 and NFSv3 network performance and data capacities, and offers an exceptional suite of supercomputing software. Cellular IRIX 6.4 is a leader in open systems with X/Open XPG4 Base 95 Profile and broad POSIX standards compliance. Government and other security conscious customers can take advantage of standard C2 security, or choose Silicon Graphics' Commercial Security Pack (which adds kerberos, ACLs, and extended capabilities to C2 security without requiring a full upgrade to a future B1 release).

Cellular IRIX 6.4 is supported on Silicon Graphics' Origin2000, Origin200™, and Onyx2™ Impact systems. Future releases of *Cellular IRIX* will be supported on a broad range of Silicon Graphics systems.

1.0.3 Cellular IRIX and UNICOS™

Silicon Graphics and Cray Research came together as one company in the first half of 1996, creating an opportunity to merge Silicon Graphics' premier supercomputing visualization and Cray's highest-performance supercomputing simulation environments. Cray's approach historically embraced large physical memory and vector processing; Silicon Graphics emphasized virtual memory and scalar multiprocessing. Both companies share a common interest in dealing with the world's toughest computing challenges and managing processing performance and resources on a vary large scale, while maintaining agressive standards support as well as compatibility with our (now) common past. Looking ahead, the merged companies see the challenge of ever-increasing CPU counts, processor performance, memory demands, I/O capacities and bandwidth, and a predictable, efficient operating environment to tie it all together. Already, a common software environment is starting to appear with NQE™ availability on Cellular IRIX, which will be soon followed by the integration of Cray's Data Migration Facility (DMF™) into Cellular IRIX and Silicon Graphics' XFS™ filesystem into UNICOS.

Back to top

1.1 *Cellular IRIX*

1.1.1 The Vision

Cellular IRIX brings together the best of the centralized mainframe and the distributed desktop worlds. The mainframes of the 1960s and superminicomputers of the 1970s were a single shared resource to their user communities: relatively easy to administer and easily optimized for maximum throughput. In the 1980s advances in semiconductor technology brought down the cost of computing; individual desktop workstations and personal computers proliferated. With desktops dedicated to individuals, the scalability of the environment increased. Compute cycles could be added incrementally and relatively inexpensively. Availability also increased. Any desktop could fail without impacting another, and if mail or files were on a separate server, the user of the broken desktop could move to another desktop and continue working. An environment of desktops was not, however, nirvana. Desktop administrators were frustrated as they performed the same tasks over and over from one system to the next, with end results substantially less integrated than in mainframe environments. As time passed, researchers and vendors proliferated solutions (e.g., DCE/DFS™, Tivoli™) attempting to manage workstations like a mainframe.

Today, the distributed operating system can become a reality. One operating system can scale across many incremental and modular units of computing (following the desktop price/performance curve), while at the same time providing the robustness and availability of a distributed environment. The vendors with extensive knowledge of the datacenter (including Silicon Graphics and Cray) have optimized the new scalable shared resources for high throughput.

The modern distributed operating system combines the virtues of the datacenter environment and the distributed workstation environment:

- Scalability

- Compatablity
- Availability
- Throughput

1.1.2 Scalability

Cellular IRIX rearchitects Silicon Graphics' highly-successful, advanced 64-bit multiprocessing IRIX operating system for increased scalability and enhanced availability. The *Cellular IRIX* architecture modularizes IRIX, distributing global kernel text and data into optimized SMP-sized chunks or "cells". Cells represent a control domain consisting of one or more machine modules, where each module consists of processors, memory, and I/O. Applications running on these cells rely extensively on a full set of local operating system services, including local copies of operating system text and kernel data structures. Contention for locks is largely limited to the local cell, providing application scalability and performance equal to that observed on small-scale SMP systems.

Inter-cell coordination allows application images to directly and transparently utilize processing, memory and I/O resources from other cells without incurring the overhead of data copies or extra context switches. Thus, the operating system scales as the machine size and workload increases, while at the same time allowing all of the machine's computing resources to be globally allocated and managed for maximum overall efficiency.

The Cellular IRIX architecture will be rolled out in phases. Cellular IRIX 6.4 is the first release, and provides substantial infrastructure for scaling, availability, and application performance, with scalability to 32 processors. The second phase of Cellular IRIX on Origin2000 will provide scalability to 128 processors. Later releases will support much larger numbers processors and provide other functional enhancements.

1.1.3 Compatibility with IRIX 6.2

Cellular IRIX is compatible: the vast majority of IRIX 5 and IRIX 6 applications will run on *Cellular IRIX* without change. The highly-functional user and developer interfaces from IRIX 6.2 are the user and developer interfaces on the first *Cellular IRIX* release. The administrative interfaces for Cellular IRIX have been enhanced since IRIX 6.2. Shared memory programming was the main programming model for IRIX 5 and IRIX 6. *Cellular IRIX* also retains this model. Message-passing programs run well on POWER CHALLENGE™ systems and arrays, and will run even more swiftly with *Cellular IRIX* over CrayLink with shared memory.

Cellular IRIX is revolution via evolution. Radical improvement in scalability within the kernel is masked from and gradual to users, developers, and administrators. Since IRIX 5 and IRIX 6 applications run on *Cellular IRIX* without change, ISVs and other developers can optimize applications for the hardware architecture or recertify existing binaries, preserving ISV and customer investments in software. *Cellular IRIX* developers will interact with the familiar IRIX 6.2 environment. Users and administrators will see a single-system image: every user process sees the same state for the same system call, while system administrators have one set of files and one configuration to manage.

Users, developers, and system administrators also retain the industry-leading functionality of IRIX 6.2:

- The industry's highest-capability 64-bit operating system:

    - 1-TB virtual memory addressing
    - 9-million TB files and 9-million TB filesystems (using Silicon Graphics advanced journaled XFS filesystem)
    - Compatible development and run-time environments for new and existing 32- and 64-bit applications
- Standards leadership
    - X/Open XPG4 Base 95 Branding
    - FIPS 151-2 certification
    - POSIX 1003.1, 1003.1b, 1003.1c, and 1003.2 compliance
- Advanced real-time functionality
- A complete suite of supercomputing software for predictable, efficient throughput on complex workloads:
    - Parallel languages and tools
    - Hard and soft affinity
    - Support for array sessions
    - Extended project and array accounting
    - Network Queueing Environment (NQE)
- Availability software:
    - IRIS FailSafe™
    - Integral checkpoint-restart

1.1.4 Availability

Traditional UNIX systems offer very crude recovery mechanisms. Applications interrupted by maintenance shutdowns or system crashes are typically restarted at the begining. Users unfortunate enough to be logged-in usually lost at least some of any work in progress.

The previous release of IRIX (IRIX 6.2) provided increasing levels of improved *system* availability. Standard availability offerings in IRIX 6.2 included XFS journaling for fast reboot and *availmon* for fast monitoring and communication of downtime events, duration, and symptoms. Optional system availability software included disk mirroring and RAID offerings, IRIS FailSafe for systems application failover, and Hibernator II™ Checkpoint-Restart.

As the Cellular IRIX architecture evolves, system availability will be greatly enhanced. Multiple cells in a forthcoming release will allow portions of a system to be taken off-line administratively without an interruption of service. In later releases, operating system services will be continually available because cells will contain random hardware and software failures within a single cell.

Though the OS will be continously available, applications running on a failed cell will still need to be restarted after a failure. To provide application recovery today and in the future, a new implementation of checkpoint-restart is an integral component of Cellular IRIX 6.4. With checkpoint-restart, administrators and privileged users can *snapshot* important jobs regularly in mid-execution. *Cellular IRIX* can use the most recent image to return an application to the state it was in at the time of the snapshot. Checkpoint-restart allows users with long-running jobs to regularly capture their progress, protecting themselves against unexpected downtime events. Administrators can perform regular maintenance or parts replacement, distribute a computing load across multiple systems, and manage the load on a single system at different points in time, without losing application state or forward progress.

Cellular IRIX 6.4 checkpoint-restart is kernel-based and application transparent. Users can checkpoint existing (ELF-linked) binaries without modification. The command line interface and core functionality are based on the POSIX 1003.1a Draft 11 standard. Checkpoint-restart supports MPI sessions as well as regular sessions, processes, and process groups. Open files, including pipes, are supported. Checkpoint-restart is also fully compatible with Cellular IRIX 6.4 accounting. Future releases of Silicon Graphics checkpoint-restart are expected to provide additional support for network, database, and high availability applications through support of queues, sockets, raw disks, X-windows, graphics, and special devices. Integral checkpoint-restart in Cellular IRIX 6.4 replaces the (optional) Softway, Ltd. Hibernator II™ product supported on earlier IRIX 6 releases.

Back to top

1.2 Cellular IRIX 6.4

The first release of the new software architecture, Cellular IRIX 6.4, initially supports up to 32 processors, with planned enhancements to 128 processors. The system has been tuned for computationally-intense and general-purpose computing. Cellular IRIX 6.4 builds the foundation for scalability with new facilities for data management, scheduling, and I/O. The memory and thread management features of Cellular IRIX 6.4, combined with the high-bandwidth and low latencies of Origin2000, establish Silicon Graphics as the leader in functionality and performance for distributed shared memory systems. No competitors provide the same levels of performance and functionality.

X1.2.1 New Memory Management Features

VM Scalability

The Cellular IRIX 6.4 VM subsystem has been enhanced for performance and scalability. Specific instances include:

- The granularity of synchronization for memory objects has been reduced to cover relatively small and local objects. For example, instead of a huge, global, highly-contended page cache, Cellular IRIX 6.4 has one small page cache per memory object.

- All large global data structures have been converted into per-node data structures with local synchronization. Cellular IRIX 6.4 has one free memory list per node, each with its own locks.

Upcoming releases of *Cellular IRIX* will build on these VM enhancements to provide increasingly scalable performance.

Memory Placement

The vast majority of existing IRIX 5 and IRIX 6 applications will run on Cellular IRIX 6.4 without change.

Cellular IRIX 6.4 manages system resources and applications while fully considering the topological characteristics of the underlying architecture, minimizing any latencies that might appear while sharing data. Though the vast majority of applications will not require operating system assistance to run quickly, the system can make use of underlying architectural mechanisms to implement latency minimization techniques, including:

- Dynamic Memory Replication

  When the shared memory page is read-only, the hardware will detect the sharing and the memory management subsystem in Cellular IRIX 6.4 will intervene to either replicate the specific memory page being shared--placing a copy in the memory location closest to each of the distributed processors--or identify an existing nearby copy that the application can reference.

- Dynamic Memory Migration

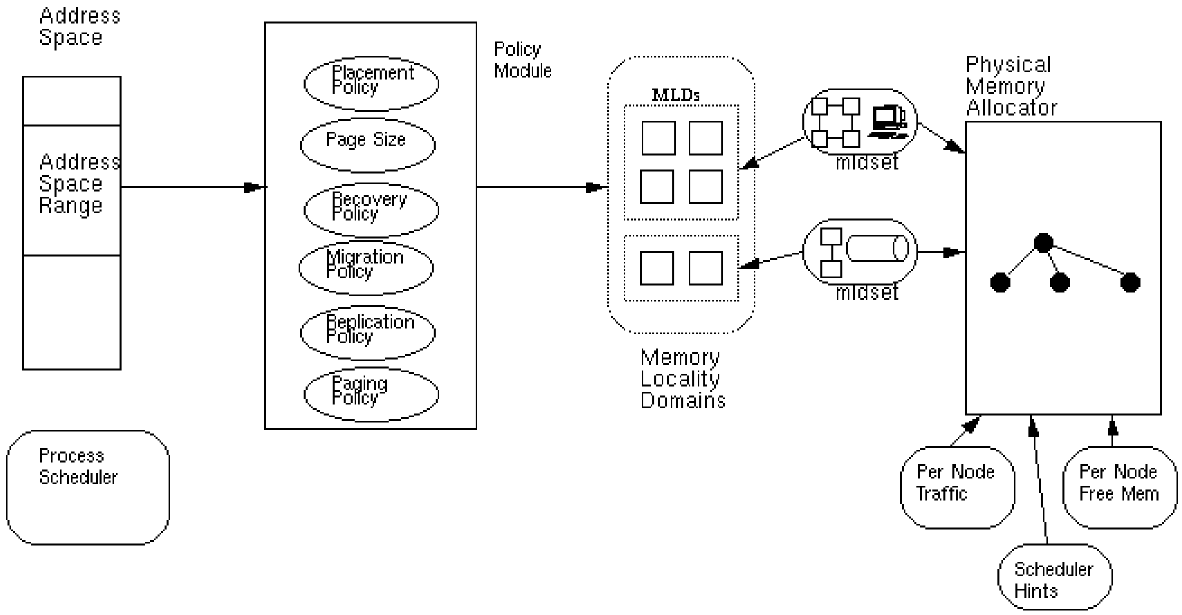
  When the single shared memory page is being accessed, counters in the hardware will detect the off-node references. When the number or proportion of the references passes some pre-defined relative or absolute threshold, the Cellular IRIX 6.4 memory management subsystem will migrate the common individual memory page close to the other sharing process for better locality, increasing the likelihood that related pages and threads will also migrate and maximizing the performance of all the sharing threads. Cellular IRIX 6.4 includes mechanisms to monitor and control subsequent memory behavior, with features for bounce detection, freezing and melting memory page locations, and delaying page migration as needed.

For applications which need to draw the last ounce of performance out of the underlying hardware, Cellular IRIX 6.4 also provides explicit data and thread management through compiler directives and application-driveable memory policies.

Compiler directives are one form of explicit management build into the Cellular IRIX 6.4 software environment. With a coming release of the Silcon Graphics compilers, developers can define memory and thread behavior in Fortran on a "virtual architecture" which allows placing memory pages within a specified number of "hops" of other pages, mapping threads to pages, and locating threads a specific distance from virtual hardware resources including graphics and I/O.

With memory policies, individual memory address spaces can be explicitly placed in a class so that all applications running in that address space will run similarly. The administrator defines class behavior once; new address spaces placed in the class will then exhibit the desired memory behavior. Two new memory management mechanisms in Cellular IRIX 6.4 support this functionality:

- Memory Locality Domains (MLDs) provide a mechanism to establish *where* the virtual address space should be placed (how closely memory pages must be placed together on potentially non-contiguous physical memory and where memory and threads should be located on the system in relation to graphics and I/O resources). Two or more MLDs can be grouped together in an MLDset.
- Policy Modules are used in conjunction with MLDs and MLDsets to determine *how* the virtual address space will behave on the hardware topology (paging policy, replication, migration, etc.)



Implicit data placement tools are also available:

- *dprof* analyzes thread VM access patterns to determine the best initial placement for data and threads.
- *dplace* uses the placement data from *dprof* to envoke initial placement on the hardware topology.

Cellular IRIX 6.4 provides multiple page size support. A process can use the Policy Module mechanism to map portions of its address space with different memory page sizes. Users can also specify page sizes using *dplace*. Without this feature all the memory pages used by a process are of the same page size. If a process has a large working set, it will incur high TLB miss overhead while mapping a large set of pages. This is especially true of applications like databases that run on high-end Silicon Graphic platforms. By using larger page sizes, the processes can reduce TLB miss overhead and provide a significant performance boost. This feature is available only for R4400/R10000-based systems. All the page sizes supported by the processors (16K, 64K, 256K, 1MB, 4MB, 16MB) are supported.

Back to top

1.2.2 New Scheduler Features

The scheduler in Cellular IRIX 6.4 has been enhanced to better support large numbers of processors and user processes. Enhancements include process-to-memory affinity and currency-based space-sharing for interactive users.

Process-to-memory affinity lets the scheduler interact with the new memory management subsystem to give short-term priority to processes and threads whose pages are still largely memory (rather than just cache) resident.

The new currency-based space-sharing scheduler changes the timesharing/interactive job scheduling paradigm significantly, insuring long-term fairness for all users while enabling short-term decision making for optimal application performance. Traditional UNIX scheduling assigns a priority to each user. Newly spawned processes inherit the same priority, which degrades as the process runs to allow lower-priority processes an opportunity to run. With this scheme, a single user with a relatively high priority can spawn a number of threads with the same priority, monopolizing the system unfairly. A user with 10 processes would receive ten times the CPU time of a user with one equivalent process and an equal priority.

With the new Cellular IRIX 6.4 scheduler, unfair monopolization of the system is much less likely. With currency-based scheduling, the system administrator assigns each user a proportion or share of total CPU time ("weight"). Users with multiple threads or processes must indicate the relative importance of each by allocating each a proportion of the user's total currency. For example, assume there are two users on the system, one with a weight of 100 and the other with 50. The first user has two jobs: one urgent and one less urgent. The second user has two jobs of equal importance. Both users would allocate their currencies as follows and receive the appropriate share of the system over time:

User #	Weight	Alloc. Job 1	Alloc. Job 2	Sys. % Job 1	Sys. % Job 2
User 1	100	75	25	1/2	1/6
User 2	50	25	25	1/6	1/6

If there are more processes than processors, the processes with higher currency are scheduled ahead of the processes with lower currency and are charged for their usage as it occurs. Conversely, processes with lower currency are not running, but earn CPU time by having weight, by being ready to run, and by waiting on busy run queues. Eventually (over microseconds of computer time) these trends will interact to provide the waiting processes with their share of system time. The long-term assurance of fairness allows the scheduler to make short-term optimizations (e.g., taking advantage of cache or memory affinity), knowing that the earnings-rate will eventually dominate and assure each user their share of system time.

Space-sharing scheduling recognizes that it is often more efficient to run multiple parallel jobs on fewer CPUs. Insisting on more CPUs with space-sharing will not effect the CPU time received: a four-CPU job receives one-quarter the time of an equal priority job with one CPU.

Back to top

1.2.3 New I/O Features

Cellular IRIX 6.4 provides increased I/O support for scalability in Origin2000 and Origin200 systems:

- Kernel threads (covered later in Section 9.0, "Symmetric Multiprocessing and Multitasking")
- Hardware graphs
- I/O bus provider layering

Provided for the first time in Cellular IRIX 6.4, the generalized hardware graph facility provides a mechanism to name, access and manage hardware resources (CPUs, memory, nodes, and particularly I/O devices) on a large or topologically-complex system. Traditionally, I/O devices have been represented by a character or block special file named by the administrator and created with the *mknod()* command. The administrator also specified the device's major number (device driver) and minor number (device specific information, such as partitions for disks and specific media handling requirements for tape drives). This traditional mechanism (*/dev* entries) is still available in conjunction with hardware graphs on Cellular IRIX 6.4.

With the hardware graph facility, Cellular IRIX 6.4 defines an internal graph of the system hardware and its relationships to device drivers. The graph includes a vertex for each hardware entity, and edges connecting the vertices that represent relationships. The result is systematic and persistent names for all devices, though the system administrator can create simplified aliases for each device, if desired.

I/O bus provider layering simplifies the interfaces presented to the device driver writer. Bus types used to be merged into single set of interfaces, and device driver APIs stressed portability without providing much opportunity for performance optimization. I/O bus provider layering moves towards bus-specific interface APIs for portability and performance-oriented device drivers. With I/O bus provider layering, driver code sharing across platforms with the same bus type is substantially easier.

Back to top

1.3 A Common Direction: Bringing Together Cellular IRIX and UNICOS

The software engineers at Silicon Graphics and Cray have found that software environments from the now-merged companies have much in common. They share a common heritage of technical innovation, extensive compliance with UNIX standards, and long records of success in demanding technical markets.

1.3.1 Standards

UNICOS and Cellular IRIX are rooted in UNIX System V, and the respective operating environments were the first two in the computing industry to be X/Open XPG4 Base 95 profile branded. Both UNICOS and Cellular IRIX support key POSIX operating system standards (POSIX 1003.1, 1003.2, and 1003.1c) as well as FIPS 151-1 and 151-2. Silicon Graphics and Cray have both invested heavily in keeping up with the latest language efforts: both provide robust Fortran 90 and up-to-date implementations of C and C++. UNICOS also supports excellent implementations of the PVM and MPI message-passing interfaces, aimed at the highly-scalable T3E. Silicon Graphics' implementations of the same interfaces have been tuned for POWER CHALLENGE arrays.

1.3.2 Common Technical Heritage

Silicon Graphics and Cray have both been industry leaders in high-performance computing, and share an interest in high-performance networking and I/O. Both companies have been 64-bit computing leaders, providing very large memory, file, and filesystem addressing to advanced technical and commercial applications. The companies offer mechanisms for raw, unbuffered I/O that scale with the number of processors. Both Cellular IRIX and UNICOS support advanced volume management including high-performance disk striping and logical multi-volume virtual devices. Both systems support mature implementations of ATM and serial HiPPI with leading-edge performance as well as a range of networking software options including NFSv2, NFSv3, AFS, and DCE/DFS.

1.3.3 Resource Management

Advanced technical users working on the frontiers of science and industry frequently want to wring the last ounce of performance out of their system. Workload management tools enhance these efforts. Both Cellular IRIX and UNICOS support advanced features for system performance optimization, including:

- Gang scheduling: related threads can be run as a group
- Soft affinity: processes can be scheduled where cache (and memory) data is re-usable
- Enhanced accounting: track jobs, users, projects
- NQE

With Cellular IRIX 6.4, Silicon Graphics adds currency-based scheduling, a feature previously available only on leading-edge supercomputing systems.

1.3.4 Scalability: Today and Tomorrow

Before coming together, Cray and Silicon Graphics charted similar courses: developing scalable hardware and scalable distributed operating systems. UNICOS/mk™ and *Cellular IRIX* are complementary approaches. With each, the UNIX kernel is modularized and distributed into local and global services. With UNICOS/mk, every processor sports a microkernel. Global services, running on their own processors, are added as the processor count climbs to 2048. Applications directly manage the system's distributed memory via message passing.

With *Cellular IRIX*, the operating system provides full shared memory support via distributed shared memory across 1 to 32 CPUs, and will support larger numbers of processors in subsequent releases. While *Cellular IRIX* technology is evolving to support thousands of processors later in the decade, UNICOS/mk provides globally addressable memory today on thousands of processors without hardware cache coherence. Applications developed on the Cray T3E using explicit message passing languages such as MPI and PVM will run on future implementations of *Cellular IRIX* with a simple recompilation.

1.3.5 Datacenter and Supercomputing Center Capabilities

Silicon Graphics has been tremendously successful in the departmental supercomputing market with the POWER CHALLENGE and 64-bit IRIX systems, but, with some exceptions, has not sold extensively into the very largest datacenters. Cray has dominated those same datacenters for more than 20 years. A world-class datacenter operating system must provide the following:

- Support for thousands of users and processes
- Dynamic tuning for batch and interactive response
- Explicit user and administrative control over application performance
- Very low communication and synchronization overhead
- Very high single-system availability
- Very high performance I/O and very large filesystems

1.3.6 Functionality Migrating Soon Between IRIX and UNICOS

Silicon Graphics and Cray plan to introduce a combined UNICOS and Cellular IRIX operating environment on a common future-generation hardware platform. In the meantime, UNICOS and Cellular IRIX will coexist as the sole operating systems running on Cray and Silicon Graphics hardware. Each will be updated and enhanced with releases over the next few years. During that time, value-added functionality from Cray's UNICOS and UNICOS/mk will be ported to Cellular IRIX, as important Cellular IRIX functionality will be migrated into UNICOS/mk and UNICOS. Examples of this cross-pollination of important software functionality include (but are not limited to):

- Application Source (API) Compatibility

  UNICOS API compatibility will be phased into future versions of *Cellular IRIX*. Language compilers, including CF90 and Fortran 90, will share common front ends. Cray's TotalView debugger, performance and source code analysis tools for SMP systems, and the MPP Apprentice Performance Tool will be integrated into the the *Cellular IRIX* development environment. Cellular IRIX will support Cray's high-performance library implementations, including the scientific library *(libsci)*, the math library *(libm)*, Fortran run-time libraries *(libf, libfi)* and the utility library *(libu)*.

- Cray Network Queueing Environment integrated with Cellular IRIX

  NQE is a robust network-based workload management system that automatically load balances tasks across heterogenous systems in a network. NQE provides automatic job routing, event dependency, and guaranteed file transfer. NQE runs each job submitted in a heterogenous environment as efficiently as possible on the resources available, without adding additional hardware. NQE provides faster turn-around times and automatic load balancing to assure effective resource utilization. NQE is supported on UNICOS™, UNICOS/mk™, Cellular IRIX™, AIX™, HP-UX™, SunOS™, Solaris™, DEC UNIX™, Windows™, and Macintosh™ platforms. UNICOS and Cellular IRIX implementations of NQE will be more tightly integrated with their host environments than NQE implementations on other platforms.

- Cray Data Migration Facility ported to Cellular IRIX

  DMF, an advanced data management package, monitors disk usage and transparently moves files between disks and secondary storage to maintain an administrator-specified amount of free disk space. DMF promotes efficient use of disk space, ensuring that the most costly disk blocks contain the system's most active files.

- Silicon Graphics' XFS ported to UNICOS and UNICOS/mk

  Silicon Graphics' XFS journaled filesystem with advanced volume management includes striping, disk concatenation, and optional disk mirroring functionality. XFS supports extents and guaranteed-rate I/O for predictable throughput. The journaling feature ensures fast system reboots, regardless of the size of user filesystems.

Back to top

2.0 Standards

2.1 X/Open

IRIX 6.2 received X/Open Base 95 Profile branding from the X/Open Company Ltd in January 1996. Many vendors have already embraced the X/Open effort to evolve the UNIX system and provide a single API for assured portability.

2.2 SVID

Cellular IRIX 6.4 provides the API described in the System V Interface Definition, Issue 3 (SVID3), the defining document for System V Release 4 (SVR4). Compliance with the SVR4 API is tested using a variety of commercial and public domain test packages. One of the most comprehensive of these is the Generic ABI Test Suite (gABI) developed by UNISOFT. The gABI currently includes more than 7,800 tests for interface compliance.

2.3 MIPS ABI

IRIX 6.2 was the reference platform for the MIPS ABI 1.2.1 and is compliant with the MIPS ABI 1.1. IRIX 5.3 remains the reference platform for the MIPS ABI 1.1. The MIPS ABI is defined in the Black Book (MIPS ABI Conformance Guide), and is determined by the member companies.

Previous releases of IRIX have been selected as the reference platform by the MIPS ABI group. The member companies--all of the vendors that offer SVR4 on MIPS processors--agreed that a suitably-compiled binary from IRIX would operate correctly on their platforms.

2.4 POSIX

Cellular IRIX 6.4 meets the standards established by the POSIX 1003.1 Specification. IRIX has been compliant with the FIPS 151-1 interpretation of P1003.1 since IRIX 4.0.5, and has been certified as compliant with the 1990 version, corresponding to FIPS 151-2, since IRIX 5.2.

Cellular IRIX 6.4 incorporates SVR4 commands and utilities that have been brought up to full compliance with the POSIX 1003.2 Specifications. Compliance was certified by the X/Open Base 95 Profile branding.

Cellular IRIX 6.4 provides the interfaces specified by the IEEE POSIX 1003.1b-1993 (formerly 1003.4 Draft 14) standard for real time. Cellular IRIX 6.4 contains kernel enhancements for the support of real-time programming and guaranteed interrupt latencies. POSIX 1003.1c threads (pthreads) the industry standard for multithreaded programming, are supported in Cellular IRIX 6.4.

2.5 Single UNIX Specification (SPEC 1170)

Silicon Graphics provides most of the Application Programming Interfaces (APIs) defined in the X/Open "Single UNIX Specification" UNIX 95 (formerly SPEC 1170). Silicon Graphics is committed to complying fully with this X/Open standard in a forthcoming IRIX release, enabling software developers to produce products portable to any version of the UNIX operating system that supports the specification.

Back to top

3.0 User Interface and Graphics

Silicon Graphics provides a single, advanced user interface across its product line, plus a broad range of standard and optional graphics and visualization development tools.

3.1 X11 and Motif™

Leading-edge technology continues to be the focus for the developer user interface libraries for Cellular IRIX 6.4. Cellular IRIX 6.4 supports Release 6 of the X11 Window System for both client and server, and IRIX IM, Silicon Graphics' enhanced version of OSF Motif 1.2.4. The X11R6 libraries and IRIX IM libraries have been updated to support 32-bit and 64-bit addressing. In addition, there is support for CID keyed fonts, the X double buffering extension, and enhanced font installation and management.

3.2 Indigo Magic™ User Environment

On systems with graphics, Cellular IRIX 6.4 supports Indigo Magic, the enhanced Silicon Graphics desktop for visual computing. Indigo Magic (now fully integrated into the World Wide Web) offers far more benefits than generic common desktops. Indigo Magic offers major value in the areas of Web integration, system administration, network awareness, and multimedia support. Implemented using the visually-enhanced Silicon Graphics version of the Motif toolkit rendered using the X Window System, this tightly-integrated environment includes:

- Icons on background for applications, documents, and local/remote devices
- Icon views allow browsing and copying from HTTP and FTP sites, as well as from directories
- Built-in Media Viewers for HTTP, movie, text, sound, VRML docs
- The ability to publish documents from your desktop to an automatically created home page
- Task-based graphical Personal Administration Tool (including modem support)
- Searchable Web-based documentation and help delivery
- Window overview/multiple desks for reducing screen clutter
- Proactive system monitoring, to alert you to system problems
- Bundled multimedia, collaborative, and Web-based tools
- Built-in access to Mac/PC file systems, including reading floppies
- SoftWindows for accessing PC applications and documents

3.3 Bundled Multimedia Products

A broad spectrum of especially useful multimedia products are bundled with each system:

- Digital Media Tools
- Netscape Mail™
- IRIS Showcase™
- InPerson™
- Mindshare™ OutBox
- WebMagic™
- Netscape Navigator™
- WebSpace™ Navigator
™

- Adobe Acrobat™ Reader Version 2.1

3.4 OpenGL®

The industry-standard OpenGL graphics development environment has been embraced by more than 30 companies including the major leaders in the PC and workstation industries. OpenGL is a cross-platform, portable API that enables 2D, 3D, and imaging applications to be developed once for deployment to a variety of hardware platforms with different operating systems and windowing environments. All OpenGL platforms conform to the OpenGL standard, insuring application portability across heterogeneous platforms.

3.5 Open Inventor™

The Open Inventor object-oriented 3D graphics toolkit offers a comprehensive solution to interactive graphics programming problems. It presents a programming model based on a 3D scene database that dramatically simplifies graphics programming. Open Inventor offers a rich set of objects that shortens programming time and extends 3D programming capabilities beyond OpenGL, X11, and Motif.

Following in the footsteps of OpenGL, Open Inventor is on its way to becoming an industry-standard 3D graphics toolkit. Like OpenGL, Open Inventor is a cross-platform API that has been ported to all major PC and workstation platforms. The Open Inventor end-user environment is distributed with Cellular IRIX 6.4.

3.6 ImageVision Library®

The ImageVision Library (IL) is an extensible, layered, object-oriented toolkit for creating, processing, displaying, and managing images on all Silicon Graphics workstations. The ImageVision Library was designed to reduce complexity for development of image processing applications. It provides a consistent interface regardless of changes or new releases of hardware or foundation software. The ImageVision Library end-user environment is bundled with Cellular IRIX 6.4.

3.7 IRIS Performer™

The IRIS Performer toolkit gives developers the means to achieve a fast, consistent frame rate for applications such as visual simulation, interactive entertainment, and simulation-based design. The IRIS Performer libraries are highly tuned for Indigo^2 IMPACT and build on the foundations of OpenGL and Cellular IRIX 6.4. IRIS Performer has the intelligence to evaluate underlying hardware and optimize performance accordingly. The key developer benefits include simplified management of the visual database and maximum performance for scene rendering on all Silicon Graphics workstations.

3.8 Shells

Cellular IRIX 6.4 provides the following shells:

- C Shell *(csh)*
- Korn Shell *(ksh, sh)*
- *tcsh*

All shells are programmable and allow for a tailorable character-based user environment.

Back to top

4.0 Filesystems

Cellular IRIX 6.4 provides an advanced filesystem with leading-edge features. It also supports a range of alternate filesystems for distributed and standards-based data access.

4.1 File Services

The Cellular IRIX file subsystem supports multiple physical disks and gives them the appearance of a single, logical filesystem with a hierarchical arrangement. Additionally, XFS filesystems may include plexed and non-plexed components.

The Virtual Filesystem interface facilitates the inclusion of several filesystem types into Cellular IRIX 6.4, including:

- EFS
- XFS
- Network Filesystem (NFS)
- ISO 9660 (CDFS)
- DOS, Macintosh, and HFS
- *swap*
- */proc* Filesystem

4.2 EFS

Ten years ago, Silicon Graphics replaced the standard UNIX filesystem with EFS, a higher-performance and higher-capacity filesystem needed by many of the company's customers. EFS increased throughput by supporting 64KB extents instead of the 8KB or smaller blocks in some UNIX implementations. EFS also implemented 8GB filesystems instead of the 2GB filesystems found on traditional UNIX systems.

Cellular IRIX 6.4 continues to support EFS for compatibility, but Silicon Graphics recommends rapid conversion to XFS, one of the most advanced filesystems available from any vendor, and a standard part of IRIX since IRIX 6.2.

4.3 XFS™

Designed from the ground up, XFS is a 64-bit journaled filesystem with a number of advanced features including very high throughput rates and guaranteed rate I/O.

XFS supports full 64-bit files and filesystems on machines running 64-bit versions of IRIX and Cellular IRIX. Systems that have a 32-bit virtual address space (for example, Indy workstations and certain versions of Indigo^2 workstations) are limited to one terabyte files and filesystems.

XFS uses database journaling technology to provide very fast filesystem recovery. No filesystem checking utility is needed--the system quickly references a small log of recently updated filesystem transactions after a system crash. Logging filesystems scale as disks and CPUs are added; traditional UNIX filesystem checking utilities do not.

The XFS filesystem supports logical block sizes ranging from 512 bytes to 64KB. Filesystem extents enable data contained in an extent to be placed on disk in contiguous filesystem blocks. The largest extent possible is about 1 million (2^{20}) blocks. Use of extents in XFS greatly increases I/O throughput by reducing disk seeks and rotational delays when data is read off or written to the disk.

Aggregate read/write performance of more than 500MB per second has been demonstrated; single-descriptor performance of more than 400MB per second has also been observed.

The XFS guaranteed rate I/O option makes XFS the only filesystem that allows applications to reserve specific bandwidth to or from the filesystem. This functionality is tremendously valuable to applications that perform real-time data capture or delivery.

The XFS filesystem has a full range of graphical system administration tools to create, delete, mount, export, and modify filesystems.

4.4 NFS™

Distributed filesystems simplify the way data is accessed by making remote files appear local. NFS, with an installed base of more than 3 million seats, is the de facto UNIX standard for distributed filesystems.

Until recently, virtually all NFS implementations used the NFS Version 2 protocol (NFSv2). But NFSv2 was not designed with today's high-performance networks or large file sizes in mind.

Silicon Graphics implemented NFS Version 3 (NFSv3) for the first time in IRIX 5.3. The Version 3 protocol also allows 64-bit file addressing, which makes it possible to use NFS to transfer files larger than 2GB over the network for the first time. Since NFSv2 compatibility is preserved in ONC+ (in a more efficient implementation), customers get the best of both worlds: the performance of NFSv3 with the backward-compatibility of NFSv2.

To enhance file sharing performance and reduce network load, NFS clients can use the Cache Filesystem (CFS) function, retaining a local copy of frequently accessed remote files and updating the cached copy only when the original has been changed.

The Network Information System (NIS) is provided for centralized filesystem management. The automounter service automatically mounts and unmounts NFS filesystems. The NFS locking service, *lockd*, allows locks to be used with remotely mounted files.

4.5 ISO 9660

ISO 9660 is a filesystem type used to mount CD-ROM discs in the High Sierra or ISO 9660 formats (with or without the Rock Ridge extensions).

4.6 DOS, Macintosh Floppies (and other media)

The DOS filesystem driver supports 5.25-inch floppy drives in three standard formats when used with the freestanding SCSI floppy drive. The standard single- and double-density, dual-sided 3.5-inch drive and the 3.5-inch 20.1MB floptical drive are also supported, as are HFS/MAC formats.

4.7 *swap*

swap allows either a file or block device to be used as a swap resource.

4.8 */proc*

The */proc* filesystem allows running processes to be accessed and manipulated as files by ordinary system calls such as *open, close, read, write, seek,* and *ioctl*.

Back to top

5.0 Volume Management

The XFS volume manager *xlv* is a superset of the *lv* volume manager. It supports striping, concatenation, and, optionally, disk mirroring. The system administrator can make online dynamic changes to volumes, including extending the size of a mounted filesystem, with a set of graphical system administration tools.

Back to top

6.0 Networking Solutions and Standards

Silicon Graphics offers a complete set of high-performance networking products, both local and wide area, with the fastest IP connectivity in the industry. Ultra-high-performance tuning and optimization for Silicon Graphics platforms plus support for Internet standards result in the fastest, most flexible networking standards available anywhere.

XTALK networking boards can be used on Origin2000 systems, and, with an expansion box, on Origin200 systems. PCI networking cards are available on all models.

Cellular IRIX 6.4 supports the following high-performance networking solutions:

- HIPPI-Serial (XTALK): 800Mb/sec with optional support for proprietary low-latency HIPPI-Bypass
- Multiport OC3c (XTALK): four-port ATM
- 100Base-T (XTALK): four-port Fast Ethernet
- FDDI (PCI): single or dual-attach fiber and single-attach copper configurations

Cellular IRIX 6.4 also supports (or will support) the following networking software options:

- Network Filesystem (NFS): Versions 2 and 3, Cache Filesystem (CFS), and AutoFS
- Bulk Data Service (BDS)

- DCE/DFS Version 1.1 (eventually)

Cellular IRIX 6.4 implements the following Internet RFC and non-RFC standards, among others:

Table 1-1 RFC Standards

RFC	Protocol	Name
678		Standard File Format
768	UDP	User Datagram Protocol
791	IP	Internet Protocol
792	ICMP	Internet Control Message Protocol
793	TCP	Transmission Control Protocol
821	SMTP	Simple Mail Transfer Protocol
822	MAIL	Format of Electronic Mail Messages
826	ARP	Address Resolution Protocol
854	TELNET	Telnet Protocol
959	FTP	File Transfer Protocol
1014	XDR	External Data Representation
1042	IP-IEEE	Internet Protocol for IEEE 802
1055	SLIP	Serial Line Internet Protocol
1057		Portmapper
1094	Sun-NFS	Network Filesystem Protocol
1122	TCP/IP	Internet Hosts Communication Layers (Interoperability to mandatory level only)
1123	TCP/IP	Internet Hosts Communications Layers (Interoperability to mandatory level only)
1156	MIB	Management Information Base
1157	SNMP	SNMP
1213	MIB-II	Management Information Base II
1323	TCP-HIPER	Transmission Control Protocol: Large windows, etc.
1532	DHCP	Dynamic Host Configuration Protocol
1533	DHCP	Dynamic Host Configuration Protocol
1534	DHCP	Dynamic Host Configuration Protocol
1541	DHCP	BOOTP Protocol
1813	NFSv3	NFS Version 3 Protocol Specifications
1945	HTTP/1.0	Hypertext Transfer Protocol

Table 1-2 Non-RFC Standards

Non-RFC	Name
r-commands	4.3 "r-commands" (rsh, rlogin, rcp, rexec)
ONC/ONC+	Sun ONC+ 1.2 and Sun ONC 4.2 but not NIS+
HiPPI	HiPPI-PH (X3.183-1991)
HiPPI	HiPPI-SC (X3.222-1983)
HiPPI	HiPPI-FP (X3.210-1992)
HiPPI	HiPPI-LE (X3.218-1993)
ATM	ATM UNI 3.0 ATM and 3.1 ATM
DCE	OSF/DCE 1.1 Core Services (DCE 1.1 with DFS)
AFS	AFS 3.4 via Transarc Corp. (Q1 1996)

Back to top

7.0 Memory Management

Memory management in *Cellular IRIX* is a sophisticated demand paging VM system designed to meet the challenges of the broadest possible spectrum of applications and the broadest range of machines from uniprocessor desktop graphics systems to the largest multiprocessor supercomputers. The Cellular IRIX VM system has been extensively tuned and optimized to provide the functionality and performance required to support everything from the world's busiest Web servers, real-time graphics, desktop applications, video-on-demand systems, the largest database servers, to the most demanding supercomputing applications.

The Cellular IRIX VM system is the result of ten years of evolution and technology investment by Silicon Graphics to produce a memory management system that has the internal infrastructure and the external interfaces required to meet the demands of evolving multiprocessor computer systems and leading-edge applications. The Cellular IRIX VM system has been designed to scale to very large physical memories, address spaces, numbers of processors, and numbers of processes, without sacrificing performance in small desktop systems. The release of *Cellular IRIX* on Origin2000 marks the third full generation of industry-leading MP hardware supported by the IRIX VM system, starting with the release of the original PowerSeries MP systems in 1988.

The *Cellular IRIX* VM system has the following features:

- Full support for the SVR4 API, including memory mapped files and devices *(mmap)* and per page protection attributes *(mprotect)*
- Full support for the SVR4 MIPS ABI
- 64-bit user virtual addresses, allowing individual processes with address spaces spanning terabytes (on appropriate platforms)
- Dynamic loading of libraries (at run-time, a process can dynamically bind and unbind libraries using DSOs (Dynamic Shared Objects))
- Efficient process creation (fork) by using lazy replication of data pages (copy-on-write)
- Efficient support for parallel programming by sharing address space between multiple processes or threads
- Multiple swap devices, including swapping to regular files in local or remote (NFS) file systems (swap devices and swap files can be dynamically added and removed without interrupting system operation or rebooting; swap to regular files in local or remote (NFS) filesystems)
- Multiple page sizes within a single process address space (on machines with TLB support for multiple page sizes (R4000 and R10000), the VM system allows a program to request specific ranges of its address space to be represented using pages that are multiples of the base page size of the system; this allows programs with large data sets and poor spatial reference locality to minimize address translation (TLB) overhead)
- Topologically aware memory placement, including both user APIs and automatic tools to allow programs to be mapped onto Origin2000 hardware as efficiently as possible
- Automatic replication of shared text pages on Origin2000
- Automatic migration of shared data pages on Origin2000 (migration of pages can also be

performed explicitly under user program control)

For more details on the new memory systems features added to support Origin2000, refer to section 1.2.1.

Back to top

8.0 Scheduling

A Cellular IRIX process is an executable instance of a program. The context of an active process includes its virtual address space, process attributes (working directory, signal handling status, process, user, and group ID numbers), and the contents of machine registers.

To support the sizeable number of processes typically running on a Silicon Graphics workstation or server, Cellular IRIX provides a timesharing priority band with a process-scheduling algorithm that assures an equitable division of processor time among all processes. This time-sharing algorithm is non-preemptive; the running process cannot be preempted by another process (but can be preempted by the kernel).

In normal operation, the kernel pauses briefly every few milliseconds to make scheduling decisions for the processors under its control. Processes are each given a guaranteed time slice, a period when each is allowed to execute without being preempted. Often, a process yields to another process voluntarily, by making a system call, such as an I/O request, that causes it to sleep. In this case, another process is selected to run. Otherwise, at the end of a time slice, the kernel will choose which process to run next based on process priority. If two runnable processes have the same priority, the kernel runs them in turn.

There are three major types of jobs which can run on Cellular IRIX 6.4: timesharing/interactive, real-time, and background (batch). Priorities in Cellular IRIX 6.4 range from 0 (low) to 255 (high). Real-time processes can be assigned a priority anywhere in the entire range, background processes (with priority 0) run when free cycles are available, and time-sharing processes can have priorities anywhere in the range from 0 to 40. The kernel chooses which interactive/time-sharing process to run based on the currency-based scheduling mechanism described previously in Section 1.2.2. The chosen process then runs on the processor with a non-degrading default priority of 20 until it sleeps on a system call or reaches the end of its time slice.

Scheduling features in Cellular IRIX 6.4 include:

- Gang scheduling
- Affinity scheduling
- Frame scheduling

Beginning with Cellular IRIX 6.4, deadline scheduling and processor sets are no longer supported.

8.1 Gang Scheduling

The user can schedule related processes or threads to run as a group. Threads that communicate with each other using locks or semaphores can be scheduled to start in parallel. Cellular IRIX then attempts to schedule the related threads concurrently, provided the parent process has earned enough CPUs. Gang scheduling helps insure that a thread holding a lock is scheduled in the same time interval as another thread that is waiting on the lock, to avoid having the second thread spin while the first thread is not running.

8.2 Processor Affinity

As a process executes on a processor, it causes more and more of its data and instruction text to be loaded into processor cache and contiguous main memory. This creates an "affinity" between the running process and that CPU: no other process can use that CPU as effectively, and the process cannot execute as fast on any other CPU. The scheduler on a Cellular IRIX-based multiprocessor automatically notes the CPU on which a process last ran, and attempts to run a process on that same CPU, based on the assumption that some of its data remains in cache and possibly local memory on that CPU.

8.3 Frame Scheduling

Real-time applications require precise rates of forward progress, hence real-time users need complete control over a fixed set of system resources. The REACT/Pro frame scheduler is a process execution manager that completely takes over scheduling and dispatching processes on one or more CPUs. The frame scheduler makes it easier to organize a real-time program as a set of independent processes, cooperating to insure that activities happen in a predefined, scheduled sequence for data sampling, visual simulation, and related applications.

8.4 Obsolete Features: Deadline Scheduling, Processor Sets

Deadline scheduling, a feature in earlier versions of IRIX, allowed the real-time user to guarantee a certain proportion of processing time to a process in successive time intervals, without being able to guarantee where in each time interval the cycles would be available. Deadline scheduling has been superceeded by frame scheduling, and consequently deleted in Cellular IRIX 6.4.

Processor sets allowed the system administrator to reserve sets of processors to run specific tasks, in effect partitioning the system into groups of CPUs that could run batch, real-time, and timesharing jobs separately. Cellular IRIX 6.4 distributes scheduling and provides much-finer-grained control over process priorities, making processor sets obsolete.

Back to top

9.0 Symmetric Multiprocessing and Multitasking

Cellular IRIX 6.4 supports up to 32 MIPS RISC processors on Origin2000, with plans to support larger numbers in the future. Multiprocessor versions of IRIX have been available since 1988. Since that time, access to kernel resources has become more and more optimized. Thousands of locks insure the synchronization of accesses to individual data structures rather than accesses to entire subsystems, reducing the likelihood that any given processor must wait for a locked resource. As a result, Cellular IRIX, like earlier releases, offers exceptional performance scalability as processors are added.

In addition to a re-entrant multiprocessing kernel, Cellular IRIX provides interrupt thread support and two types of user threads with the appropriate synchronization mechanisms in each environment. User threads allow developers to create high-performance applications for single-job throughput across multiple CPUs and more modular and efficient applications on uniprocessors. Interrupt threads allow device driver writers and real-time developers to more easily develop higher-performance code.

9.1 *Sproc* Calls and Synchronization

The system call *sproc*() lets a developer create an additional user process that shares the virtual address space (and potentially other attributes) of the parent process. The parent and child both have their own program counter, register contents, and stack pointer, but all the text and data in the shared address space is visible to both processes. There are three basic synchronization mechanisms to serialize access to shared data using *sproc* calls: semaphores, locks, and barriers.

Sproc() calls have long provided the basic mechanism upon which many parallel programs have been built on Silicon Graphics platforms. With Cellular IRIX 6.4, POSIX threads provide an alternate, industry-standard threading mechanism for new applications.

9.2 POSIX Threads and Synchronization Primitives

A POSIX (1003.1c-1995) thread is an industry-standard lightweight user execution entity which is able to share text and data with other threads within a user process context. POSIX threads are lightweight (in general and compared to *sproc* calls, not much more than a program counter, stack, and register contents), enabling quick context switches and low-cost thread creation/destruction.

POSIX threads provide a mechanism for developers to enable and exploit parallelism in programs which run on a variety of compliant multiprocessing platforms. POSIX threads are also useful across a spectrum of uniprocessors for managing asynchronous behavior or structuring applications composed of many logically distinct paths (e.g., simulations, windowing systems).

There are three basic mechanisms to serialize access to shared data using POSIX threads:

- Mutexes
- Condition variables
- Semaphores (POSIX 1003.1b)

POSIX threads are available on Silicon Graphics systems running IRIX 6.2 or subsequent releases. POSIX threads do not replace sprocs, both threading models are supported (though not in the same application) to ensure compatibility with the full range of new and existing applications.

9.3 Kernel Threads and Synchronization Primitives

Prior to release 6.2, IRIX allowed multiple traditional heavyweight processes to run within the kernel, where critical regions were protected with mutexes and spinlocks.

Kernel threads provide a lightweight kernel execution entity that shares common text and data space with other threads independent of a user process context. Kernel threads are easy to create and destroy, can be scheduled, and use common, familiar kernel synchronization primitives:

- Non-sleeping locks:
    - Spinlocks

- Sleep-wakeup locks:
    - Mutexes
    - Semaphores
    - Multireader locks

Kernel threads offer new capabilities for device driver writers and real-time programmers:

- Device driver code is less complex (common primitives) and more efficient (sleep-wakeup locks).
- Driver performance is better (low-cost thread creation and destruction)
- Real-time performance is more predictable (better pre-emption)
- Most interrupts now run as threads, with the ability to block and be rescheduled

Back to top

10.0 Enhancements to Support 64-bit Computing

The IRIS Development Option (IDO) for Cellular IRIX 6.4 includes MIPSpro™ compilers technology to generate either:

- 32-bit code for MIPS 2, MIPS 3, or MIPS 4
- 64-bit code for MIPS 3 or MIPS 4

Over the last several releases, key components of the operating system have been progressively enhanced to support 64-bit binaries as well as existing 32-bit binaries. These include (but are not limited to):

- Kernel structures
- Include structures
- Extended linking format
- Dynamic shared objects

10.1 System Interface Structures

The header files listed in Table 1-4 contain structure definitions that have been extended to support 64 bits:

Table 1-4 Cellular IRIX System Structures Extended to Support 64 Bits

```
--------------------------------------------------------------
direct.h        fcntl.h      ftw.h           grp/h
locale.h        math.h       netconfig.h     netdir.h
nl_types.h      poll.h       pwd.h           rpc.h
search.h        setjmp.h     sigaction.h     signal.h
stddef.h        stdio.h      stdib.h         stropts.h
sys/ipc.h       sys/msg.h    sys/procset.h   sys/resource.h
sys/sem.h       sys/shm.h    sys/siginfo.h   sys/stat.h
sys/statvfs.h   sys/time.h   sys/times.h     sys/tiuser.h
termios.h       ucontext.h   sys/uio.h       sys/utime.h
sys/utsname.h
```

10.2 Directory Structures

Include structures have been modified to provide the correct 32- or 64-bit sizes. Since both 32- and 64-bit environments are supported under Cellular IRIX 6.4, multiple sets of libraries reside in different directories. 32-bit compilations, targeted at the R4x00 and MIPS 2 ISA, are the default. The library path must be changed for other targets.

Each header file under */usr/include* will support both 32-bit and 64-bit compilation. Table 1-5 details the directory structures used for Cellular IRIX 6.4 libraries. Any of these directories may have nonshared sublibraries.

Table 1-5 Cellular IRIX 6.4 Library Directory Structure

Directories	Contents
/usr/lib/*.so	32-bit DSOs
/usr/lib/*.a	32-bit static libraries
/usr/lib32	32-bit MIPS 3 and 4 libraries
/usr/lib64	64-bit libraries and tools
/usr/lib64/*.so	64-bit DSOs (also MIPS3 and MIPS4 subdirectories))

10.3 Extended Linking Format

The IRIX 5.x/SVR4 standard Extended Linking Format (ELF) for object modules has also been extended for 64-bit support. ELF-64 is supported in all the tools which work with object modules.

10.4 Dynamic Shared Objects

The Dynamic Shared Object (DSO) architecture has been extended to support 64-bit addressing. DSO technology provides upward compatibility between IRIX 5 and IRIX 6, and dynamically binds applications to the appropriate shared libraries at execution time. This scheme enables a single shrink-wrapped version that can operate without change on a wide variety of MIPS hardware platforms.

Figure 1-2 summarizes changes for 64-bit support and the resulting benefits--both 32-bit and 64-bit kernels can dynamically link 32-bit or 64-bit applications as needed.

<table>
<tr><td colspan="2">Development Environment
32 or 64 bit*</td></tr>
<tr><td colspan="2">Network File Service
32 or 64 bit**</td></tr>
<tr><td colspan="2">Local File Service
32 or 64 bit***</td></tr>
<tr><td colspan="2">Libraries and DSOs
32 bit and 64 bit****</td></tr>
<tr><td>32 bit virtual Addressing</td><td>64 bit virtual Addressing</td></tr>
</table>

* Develop 64-bit applications anywhere.
** Use >2GB NFSv3 files anywhere.
*** Use large local files (to at least 1TB).
**** 32-bit IRIX 5 and 64-bit IRIX 6 applications run on the same system.

Figure 1-2 Cellular IRIX 6.4 64-bit Enhancements

Back to top

11.0 Cellular IRIX 6.4 Application Binary Options

Developers can target one of three ABIs, depending on their compatibility and performance goals.

11.1 O32 for 32-bit Applications

Typical 32-bit applications from IRIX 5 run on Cellular IRIX 6.4.

11.2 64 bits: Full Addressability and MIPS 4 Performance

Applications should incorporate 64-bit support if they will benefit from 64-bit features such as:

- Faster logical/arithmetic operations
- Larger virtual address space
- Very large file sizes

64-bit logical/arithmetic operations can increase performance on inner loops by a factor of two. Large models used in applications such as computational fluid dynamics or finite element analysis frequently require greater than 2GB virtual address spaces. Similarly, seismic and reservoir simulation applications in geophysics commonly use data files far larger than 2GB.

11.3 n32: Speed Without Porting

Applications that already exist were often written with one or more target platforms in mind, including 32-bit systems. As a result, the developer may have consciously or unconsciously included 32-bit data structures, pointer lengths, or byte alignments in ways that make a port to 64-bits difficult. ISVs and others may also want to keep memory images small to run on as many platforms as possible.

The n32 Software Developers Kit brings two major benefits:

- Existing 32-bit applications can take advantage of the features of the MIPS 4 ISA without a 64-bit port.
- Developers can compile one binary (-n32 -mips3), which runs on all currently shipping Silicon Graphics platforms.

Developers who simply recompile on Cellular IRIX 6.4 with the -n32 -mips4 flags set will generate binaries that use all 32 floating-point registers, eight argument registers, and the MADD instruction of the MIPS 4 ISA. With -n32, fundamental data types do not change: all addressing is 32 bits. Please note, however, that R4x00 systems do not support the MIPS 4 ISA.

Developers who compile on Cellular IRIX 6.4 with the -n32 -mips3 flags set will have binaries that run across the entire current Silicon Graphics product line.

Execution and compilation options for Cellular IRIX 6.4 and the platforms they run on are shown in Table 1-6.

Table 1-6 Cellular IRIX 6.4 Execution/Compilation Table

	o32		n32		n64	
	MIPS I	MIPS II	MIPS III	MIPS IV	MIPS III	MIPS IV
Flags		-32	-n32 -mips3	-n32 -mips4	-64 -mips3	-64 -mips4
Indigo, Indigo2 (R4K), Indy/Challenge S/M (R4K)						
Execution	●	●	●			
Development		●	●	●	●	●
Indy/Challenge S R5000						
Execution	●	●	●	●		
Development		●	●	●	●	●
Challenge (R4K), Onyx (R4K)						
Execution	●	●	●		●	
Development		●	●	●	●	●
POWER Indigo2, POWER CHALLENGE, POWER Onyx, and all R10K-based systems						
Execution	●	●	●	●	●	●
Development		●	●	●	●	●

Back to top

12.0 Development Environment

Cellular IRIX 6.4 provides many components to streamline applications development in a Silicon Graphics environment. With all necessary execution-only environments needed to run applications on an Cellular IRIX 6.4 system, the development products and development libraries are available as layered components.

The base product offering, IRIS Development Option (IDO), provides the basic tools and libraries needed for developing IRIX applications. These include:

- C compiler that supports both 32-bit and 64-bit development
- Pertinent code generator
- OpenGL
- X11, Motif, and Digital Media libraries
- Development tools such as *dbx, pixie,* and *prof*
- Online reference manuals

For high-performance computational applications, Cellular IRIX delivers the highly-optimized CHALLENGEcomplib for access from either Fortran or C. Cellular IRIX 6.4 provides the additional capability of Power Fortran and Power C compiler preprocessors for automatic parallelization of applications for execution on multiprocessor systems.

Additional compiler options are available, including Fortran 77, Fortran 90, and C++, each capable of producing 32- or 64-bit code. Cellular IRIX 6.4 provides 32-bit support for Pascal™, Ada83™, and Ada95™, as well as providing support to the additional high-functionality libraries such as Open Inventor™, Performer™, ImageVision™, and ViewKit™.

Cellular IRIX 6.4 also provides access to the software engineering application tools. Developer Magic ProDev WorkShop, an awarding winning X11-based visual development environment, offers unmatched debugging and performance tuning capabilities including visual debug tools, graphical visualization of structures and arrays, source code analysis and browsing, test coverage, debugging of parallel and single stream codes, performance analysis, memory leak detection, and build analysis. Tightly integrated into a cohesive development environment, ProDev WorkShop provides support for C, C++, Fortran, and Ada95.

RapidApp is a rapid application development tool for creating media-rich, visual applications for C++. Providing a drag-and-drop mechanism for building applications with objects, RapidApp offers access to X11, Motif, and ViewKit visual components and provides access to Open Inventor and ImageVision Library.

ProDev/Ada95 provides the most complete development environment for developing Ada95 applications, including extending ProDev WorkShop to understand task-based parallel applications. It provides a validated Ada95 compiler and Ada95 bindings to all Silicon Graphics libraries.

Back to top

13.0 Performance Services

Cellular IRIX 6.4 includes interfaces that can be used in systems demanding the maximum possible I/O performance. These features require source changes, but when used appropriately, the performance gains can be dramatic. These I/O-enhancing features are:

- Asynchronous I/O
- Memory-mapped I/O
- Direct I/O

13.1 Asynchronous I/O

Traditionally, UNIX systems have allowed a given process to have only one I/O operation in progress at a time. This situation forced applications, such as database servers, that required several concurrent I/O operations to adopt complex, multiprocess architectures. Asynchronous I/O allows a user application to issue I/O requests and continue processing before the kernel has transferred the data requested. This results in the application executing in parallel with (potentially) multiple, concurrent I/O operations, improving overall performance for I/O intensive applications. Cellular IRIX asynchronous I/O interfaces comply with the POSIX 1003.1b-1993 specification, allowing a user to queue *read(2)* and *write(2)* requests to a device, and receive an optional queued signal when the request completes. A process can simultaneously queue a number of requests without having to wait for any of them to complete. Since the interface is standards-compliant, applications using it are portable.

13.2 Memory-Mapped I/O

Cellular IRIX 6.4 supports the SVR4 interfaces for memory-mapped I/O. Using these interfaces, disk files are visible in the address space of the program, and can be accessed without explicit I/O operations. The kernel can then bring in pages as it would for an executable, using the demand paging features of the VM subsystem.

13.3 Direct I/O

Direct I/O enables direct transfers between disk and the user address space. It bypasses the kernel buffer cache, using DMA to transfer data directly between the disks and the user address space. Memory-mapped files, by contrast, make the data in the kernel buffer cache directly accessible from the user address space.

Direct I/O requires substantially fewer changes to an existing application compared with memory-mapped I/O. Using both direct and asynchronous I/O, the program can have complete control of its own I/O operations, yielding performance close to that of raw disk, while retaining the benefits of the filesystem. The interfaces for direct I/O were defined by Silicon Graphics.

Back to top

14.0 Real Time

The REACT™ extensions included in Cellular IRIX let programmers configure a multiprocessor system to provide deterministic performance, including response to external interrupts. REACT also provides features that simplify the implementation of real-time applications.

Cellular IRIX with REACT achieves determinism by providing the user with full control over the assignment of software activity to processors. One processor (or more, if desired) is designated as the system processor, and all non-deterministic system activity takes place on that processor. For example, system activity typically includes the UNIX scheduler and general-purpose disk and network I/O. The remaining processors are designated as real-time processors, and no system activity takes place on those processors unless explicitly requested by a real-time process.

Silicon Graphics guarantees deterministic interrupt response on a properly configured system. Performance is specified in terms of total interrupt response, which is defined as the interval between the occurrence of an external interrupt and the start of execution of a user process that was enabled by that interrupt. The worst-case total interrupt response time for a properly configured system is guaranteed not to exceed 200 microseconds. This guarantee covers all multiprocessor Onyx, CHALLENGE, and Power CHALLENGE systems.

For further information on real-time support in Cellular IRIX 6.4, refer to the *REACT in Cellular IRIX 6.4 Technical Report.*

Back to top

15.0 Supercomputing Features

Cellular IRIX 6.4 offers supercomputer-class functionality including kernel-level checkpoint-restart and extended accounting. Cray UNICOS supercomputing features will evolve into later releases of Cellular IRIX as described in Section 1.3.

15.1 Fair Share Scheduler

The IRIX 6.2 Fair Share Scheduler (based on Share II™ from Software Pty) is not available on Cellular IRIX 6.4. Some similar functionality is provided by the native currency-based space sharing scheduler of IRIX 6.4 for Origin2000. See Section 1.2.2 for more details.

15.2 Checkpoint-Restart

A new, integrated implementation of checkpoint-restart is standard in Cellular IRIX 6.4 replacing Hibernator II™ from Softway Pty., which was available as an option with earlier versions of IRIX 6. For details, see Section 1.1.4.

15.3 Extended Accounting

IRIX accounting has traditionally been process oriented. Once activated, information logged for each user login included programs run, CPU and wall clock time, and character and block I/O resources used. Cellular IRIX extended accounting adds to these capabilities via projects and array session accounting. Projects accounting allows users to bill multiple projects separately. Array sessions accounting groups process together across the nodes of an Origin2000 array under a single identifier for unified accounting, job control, and more. Both forms of extended accounting also add monitoring of data commonly tracked in a supercomputing environment, such as swaps, bytes read/written, read/write requests, and time spent waiting on the run queue and for raw and block I/O.

15.4 NQE Batch Scheduling

NQE is a load sharing and distributed batch queueing product from Silicon Graphics' Cray Research Division. Additional details are available in Section 1.3.6.

Back to top

16.0 Security

Cellular IRIX 6.4 incorporates technology originally developed for Trusted IRIX/B for identification, authentication, and auditing. Shadow password support places encrypted user passwords in a file that is inaccessible to non-privileged programs, thwarting brute force attacks on the password data base. Auditing logs security-relevant events, enabling analysis of attempts at unauthorized entry and changes made to system files and configurations. Cellular IRIX 6.4 also supports automatic password aging, password quality and reuse controls, and optional login restrictions.

These features are designed to meet the U.S. Department of Defense "Orange Book" C2 level of trust.

An optional Trusted IRIX/B layered product is planned for customers who need a B1 Level of Trust. Trusted IRIX/B assures a B1 level security environment while providing the features from Cellular IRIX that are required for compatibility with the other Silicon Graphics products and third-party application software. Multiprocessing, graphics, real-time features, the X Window System, and networking will all be included.

Back to top

17.0 Internationalization (I18n) and Localization (L10n)

The Cellular IRIX operating environment supports multibyte character sets and local conventions, and serves as the foundation for the WorldView™ family of native language support products. WorldView merges international character sets and local conventions into the Indigo Magic user environment on Silicon Graphics workstations. Users can tailor their language environment using the new desktop application "ipanel". This unique tool allows users to select their environment from a variety of keyboards, language, national customs, and timezones through an intuitive and fun new user interface. Developers can design applications to support multiple localized versions.

Silicon Graphics recognizes that it is essential to provide native language support for its international customers. Documents can be created, edited, and printed in the native language, and many of the user interfaces are localized as well. These include key user interface and system messages. Not only is text localized, but formatting of lists, text, and data--such as date, time, number, and currency--follow native conventions for each country.

Silicon Graphics will provide the following native language support products:

WorldView European
WorldView Arabic
Worldview Japanese
WorldView Korean
WorldView Chinese, for traditional and simplified Chinese forms

For all WorldView products, the Indigo Magic Desktop Interface and printing support will be localized. Cellular IRIX system messages will be localized specifically for German, French, Japanese, and Chinese.

In addition to the native language support, Cellular IRIX will provide locale support. Different languages and places have different ways of formatting data such as time, dates, and numerical amounts. A database containing such local data formats for a specific language and location is called a locale. A locale specifies both a language and a country. If an application program is written in a way that is independent of any locale--that is, if it is written to use any user-specified locale to provide input and output in the appropriate language and format--the program is said to be internationalized. If a database for a locale has been created to allow an internationalized program to use that locale, the program is said to be localized to that locale.

The locale information for Japanese, Korean, and Chinese will be included as part of the language specific WorldView products.

Planned SIlicon Graphics Locale support (Cellular IRIX)

Locale Name	Territory	Language
cs_CS	Czech Republic	Czech
da	Denmark	Danish
de	Germany	German
de_AT	Austria	German
de_CH	Switzerland	German
el	Greece	Greek
en	UK	English
en_AU	Australia	English
en_CA	Canada	English
en_US	U.S.A.	English
es	Spain	Spanish
es_AR	Argentina	Spanish
es_MX	Mexico	Spanish
fi	Finland	Finnish
fr	France	French
fr_BE	Belgium	French
fr_CA	Canada	French
fr_CH	Switzerland	French
is	Iceland	Icelandic
it	Italy	Italian
it_CH	Switzerland	Italian
ja_JP.EUC	Japan	Japanese
ko_KR.EUC	Korea	Korean
nl	Holland	Dutch
nl_BE	Belgium	Dutch
no	Norway	Norwegian
pl	Poland	Polish
pt	Portugal	Portuguese
pt_BR	Brazil	Portuguese
ru.KOI8	Russia	Russian
sk	Slovakia	Slovak
sv	Sweden	Swedish
tk	Turkey	Turkish
th_TH	Thailand	Thai
zh_CN.ugn	PRC	Chinese (Simplified script)
zh_TW.ucns	Taiwan	Chinese (Trad.script)

There are are other cultural parameters in a locale, collectively referred to as cultural support. Examples include:

Currency format:
USA $123,200.45
    vs. France 123.200.45F
    vs. Canada 123,200.45$
Date format:
    ddmmyy vs. mmddyy vs. yymmdd
Time format:
    12 hours vs. 24 hours

An addition for Cellular IRIX will include the *iconv* International Codeset Conversion library, with extensive support for Unicoe based conversions.

Back to top

18.0 System Management

Cellular IRIX offers three different levels of administrative tools, aimed at users with various needs and professional experience. A simple set of visual tools for desktop users is included as part of Indigo Magic. IRIXpro is a suite of tools for system administrators. Performance CoPilot is a tool suite for monitoring local and remote network performance.

Silicon Graphics also has developed *swmgr*, a graphical interface to *inst*, to ease software and patch installation, handle installation problems, and provide software version tracking.

A variety of ISVs, including Tivoli, Bull, CosMos/Computer Associates, Legent, and Raxco, sell system and network management solutions for Silicon Graphics platforms.

18.1 IRIXpro™

IRIXpro is an optional suite of tools for the professional systems administrator in the technical computing market. When available, IRIXpro will provide two applications:

- Propel, *rdist*-based software and file distribution (integrated with *swmgr*)
- ProVision, a centralized SNMP-based system event monitoring tool

18.2 Performance Co-Pilot™

Performance Co-Pilot, another optional Silicon Graphics tool suite, facilitates distributed, integrated performance monitoring and performance management across a range of network platforms, databases, and applications. The product is targeted at performance analysts, developers, and system administrators. Performance Co-Pilot delivers integrated solutions for Silicon Graphics systems.

18.3 New Desktop Administration

The IRIX personal system administration tools have been rewritten for Cellular IRIX 6.4. *Active Guides* walk users through administration procedures step by step.

Back to top

19.0 Backup

Users and Administrators have the option of using traditional tape archiving tools *(cpio, tar, dump)* or using the graphical backup and restore utility included with Silicon Graphics systems. XFS includes an advanced dump/restore facility (*xfsdump/xfsrestore*) to support 64-bit files and offer high-performance via multiple, parallel tape drives. We also offer the IRIX NetWorker product: a full-featured data management tool for multi system data backup and recovery. NetWorker allows for unattended backups with superior performance and scalability.

Back to top

20.0 Data Migration

The Data Management API (developed with the Data Management Interchange Group) lets programmers implement hierarchical storage management software without kernel modifications, or write high-performance *dump* programs without requiring any knowledge of filesharing structures or requiring "raw" accesses to the disk.

Back to top

21.0 Compatibility

Every effort has been made to maintain or expand compatibility between IRIX versions.

21.1 COFF (Obsoleted)

IRIX 5 and IRIX 6 binaries are produced using the SRV4 Executable and Linking Format (ELF) which replaces the ECOFF object format in IRIX 4 and enables Dynamic Shared Objects. Development and execution of COFF (IRIX 4 and earlier) binaries is not supported on Cellular IRIX 6.4. A utility to find/identify COFF-dependent binaries is provided with the release and installed on the system. It may be executed before Cellular IRIX 6.4 is installed.

21.2 Binary Compatibility: IRIX 5.3 and Cellular IRIX 6.4

Nearly all binaries built on IRIX 5 or IRIX 6 can be run under Cellular IRIX 6.4. It may be necessary to recode or recompile to take advantage of IRIX 6 enhancements. There may be rare cases--such as applications that examine internal operating system data structures--in which application code must be recompiled.

21.3 Object Compatibility: IRIX 6.1 and Cellular IRIX 6.4

Except for n32 C++, user object code is compatible between IRIX 6.1 and Cellular IRIX 6.4. However, device drivers and any other applications that access the hardware directly must be recompiled. Compatability for n32 C++ was not possible between IRIX 6.1 and IRIX 6.2 because of changes in the C++ standard. n32 C++ object code is compatible between IRIX 6.2 and Cellular IRIX 6.4.

Back to top

We welcome feedback and comments. Send mail to jcoombs@asd.sgi.com.

We welcome feedback and comments at webmaster@www.sgi.com.