The Linux/ia64 Project

HP World Conference

September 2000

Philadelphia, Pa
USA

Stéphane Eranian
Hewlett Packard Laboratories
eranian@hpl.hp.com

hp invent

Outline

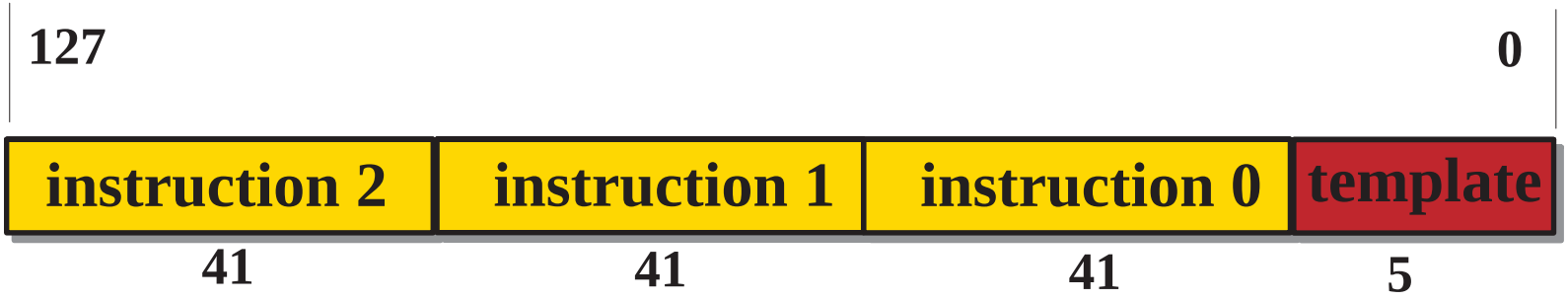
- Brief IA-64 overview
- Project overview
- IA64 SDK
- Conclusion
- Demo

What is IA-64?

- next-generation, high-performance 64-bit architecture co-designed by Intel and HP
- new EPIC paradigm: Explicitly Parallel Instruction-set Computing
- first implementation: Itanium™
  - engineering samples available since Fall'99
- complete specifications available
  - Itanium™ micro-architecture, ABI, calling conv, DIG64
- systems general availability: 2H2000

What is EPIC ?

- Explicit parallelism
  - bundles of 3 instructions
  - template field encodes:
    - type of execution units needed (M,I,B,F)
    - stop bit to express sequential dependency



- Massive resources
  - 128 integer (64bits) & 128 floating point (82bits) registers
  - lots of execution units
- RISC-like load/store model

Key features of IA-64

- Predication
  - Avoid costly branches
- Speculation (control & data)
  - safely move loads to hide memory latency
- Register Stack Engine (RSE)
  - avoid registers spill/fill on procedure calls
- Register Rotation
  - easy loop unrolling with minimal code expansion
- Performance Monitoring
  - very rich set of counters (thresholding, trace buffer, EARS)
- 64bit address space
  - 8 regions of 61 bits (8x2048PB = 16384PB)

Predication

- To reduce branching
  - 64 predicate registers (1 bit each)
  - when predicate is false instruction is considered as no-operation(nop)

```
C code:
r2= r1==0?r4+r5:r3+r6+1;

IA-64 assembler:
      cmp.eq p1,p2=r0,r1
      ;;
(p1) add r2=r4,r5
(p2) add r2=r3,r6,1
```

Control Speculation

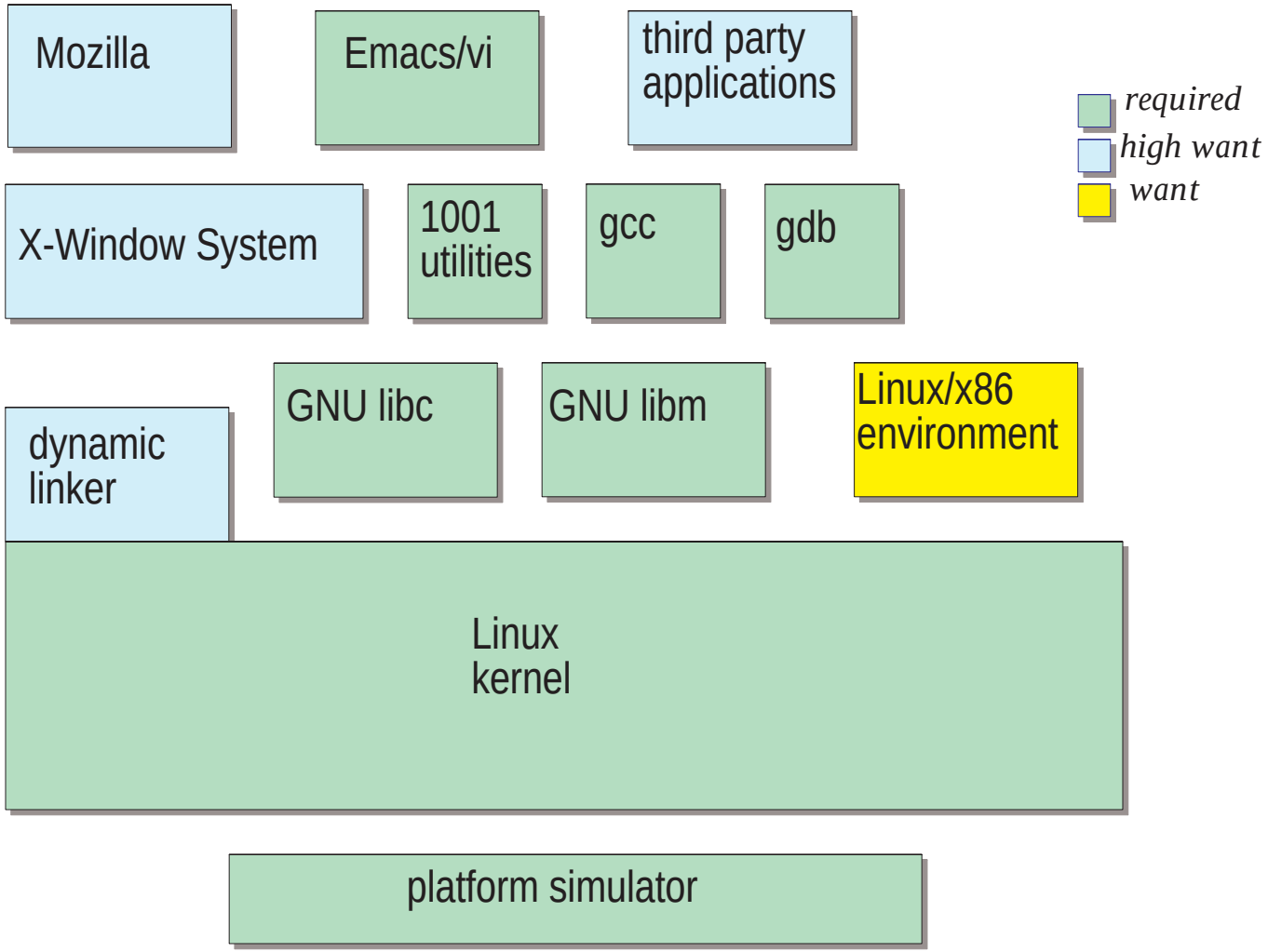
- Execution of a load before the branch that guards it
  - available for integer & floating point registers loads
- Safety ensured with NaT (Not a Thing) bit
  - "65th" bit of integer registers
  - Specific "NatVal" used for floating point registers

```
(p1) br.cond label      0  ||      ld8.s r1=[r5]          -10
     ld8 r1=[r5];;      0  || /* do something else */
     add r2=r1,r3      10  || (p1) br.cond label            0
                           ||      chk.s r1, recovery       0
                           ||      add r2=r1,r3             0
```

Why Start the Linux/ia64 Work?

- Want it to be there at launch of Itanium™
  - easy-to-install distribution, functionally complete, optimized
- Developing IA-64 optimizing compiler, kernel and applications takes time
- Why at HP ?
  - IA-64 technology expertise
  - existing Linux expertise/interest at Labs
- Code to be released to Open Source community
- Project started at HPLabs in February 1998

What's involved?



Who's Involved?

- Original Linux/ia64 project:
  - HP Labs
    - initial toolchain, kernel architecture and implementation, simulator, apps
    - demonstrated first kernel booting to user mode on simulator in March 1999 (1st LinuxWorldExpo, San Jose CA)
  - CERN (birthplace of the WWW)
    - User-level libraries (libc/libm)
- Spring '99: co-founded the "Trillian" project
    - now renamed IA-64 Linux Project

Goals of the Trillian project

- Single IA-64 Linux port
- Optimized for IA-64
- Open source the code before or at Itanium™ processor launch
- Cooperative effort to deliver best possible code
  - follow the Linux development model
  - many players contributing technology and resources:
    - Caldera, CERN, HP, IBM, Intel, Redhat/Cygnus, SGI, SuSe, TurboLinux and VA Linux Systems.

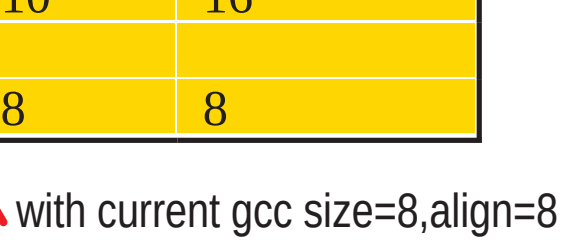
Development Toolchains

- Based on egcs-2.96 (Redhat/Cygnus)
  - some EPIC style optimizations coming
  - profiling support available
  - stack unwinding support available
  - recompile as better compilers become available
  - Binary format: standard ELF64/IA-64
- debugger: gdb-5.0 available
- SGI released Pro64 C/C++/F90 compilers
  - C/C++ compilers GNU compatible
  - check out http://oss.sgi.com/
  - more EPIC optimizations (floating points)

Gobal Properties

- LP64 Data Model

Type	Size	Alignment	Type	Size	Alignment
char	1	1	float	4	4
short	2	2	double	8	8
int	4	4	long double	10	16
long int	8	8			
long long int	8	8	void *	8	8



- Endianess
  - Little-endian is native byte order
  - Big-endian processes are possible

Kernel Design Goals

- Pure 64bit kernel for IA-64 (no legacy)
- Minimize modifications to machine independent code:
  - Added arch/ia64 and include/asm-ia64
  - Follow development kernels: from v2.1.126 to v2.4
- Some compatibility with Linux/x86 wherever possible (e.g.,sig-num, ioctl-num, errno,...)
- Optimize for 64 bit performance
- Follow standard wherever possible:
  - IA-64 SW calling convention, DiG64, EFI, UNIX ABI

Kernel VM Features

- Page Size:
  - Configurable page sizes: 4, 8 , 16, 64 KB
  - =4KB for perfect IA-32 emulation
  - >4KB:
    - allows good Linux/x86 emulation
    - better for native IA-64 binaries (8 or 16KB best)
- User address space:
  - 43bits with 8Kb pages ➠ 8TB
  - 47bits with 16Kb pages ➠128TB

Kernel Status

- Source code released since 2/2/2000 (LWE)

- All subsystems enabled:
  - vm, interrupts, signal, ptrace, network, smp, scsi, ide, usb, sound, video, platform specific support (EFI,PAL)
  - kernel modules
  - IA-32 execution support present
    - runs unmodified Linux/x86 binaries (Netscape, Acroread, WP)
    - Linux/x86 shared libraries support
- Integrated into official 2.4 kernel
  - integrated since 2.3.42
  - check out http://www.kernel.org/pub/linux/kernel/ports/ia64 for latest patches

User Level Libraries

- GNU libc v2.1.x (libc,libm)
  - shared library support available
  - pthreads available
  - some optimizations in performance critical routines
  - unoptimized libm
  - GNU libc v2.2 in progress
- C++ library (stdc++) available
- Optimized Math library (libm) contributed by Intel
  - 90 hand optimized assembly code math routines
- Source code released since 2/2/2000 (LWE)

User Level Applications

- Alpha distributions available since March'2000
  - TurboLinux (www.turbolinux.com)
  - RedHat (www.redhat.com)
  - Suse and Caldera in progress
- Graphical environment:
  - XFree86-3.3.6, GNOME, Enlightenment, Gimp available
  - XFree86-4.0 in progress
- Internet: Apache
- IA-32 emulation:
  - netscape browser, acroread, WordPerfect, Applix5.0
- Development:
  - egcs, gdb, perl, tcl/tk, python...
  - Java via gcj (java ➠ native code compiler) work from HPLabs

IA64 Software Development Kit

- ⇨ Port & develop Linux/ia64 apps on Linux/x86
- ⇨ Based on HPLabs Native User Environment (NUE)
  - Create illusion of target dev. environment:
    - cross dev. tools, headers files, libs : real name and location
    - chrooted environment: no conflict with host system
    - no tweaking of Makefiles required
    - RPM rebuild easy: `% rpm --rebuild mingetty-0.9.4.src.rpm`
  - Execution of produced binaries possible:
    - includes the HP IA-64 instruction set simulator (ski)
    - transparent execution of IA-64 binaries at shell prompt
    - mix & match IA64/IA32 binaries
    - debugging with ski

IA64 Software Development Kit

- Kernel development & execution possible
  - very good for low level code debugging
  - applications testing
  - OS research activities
- Other Open Source OS development possible
  - like FreeBSD or NetBSD
- Kit available for free since 06/13/2000
  - web site: http://www.software.hp.com/ia64linux

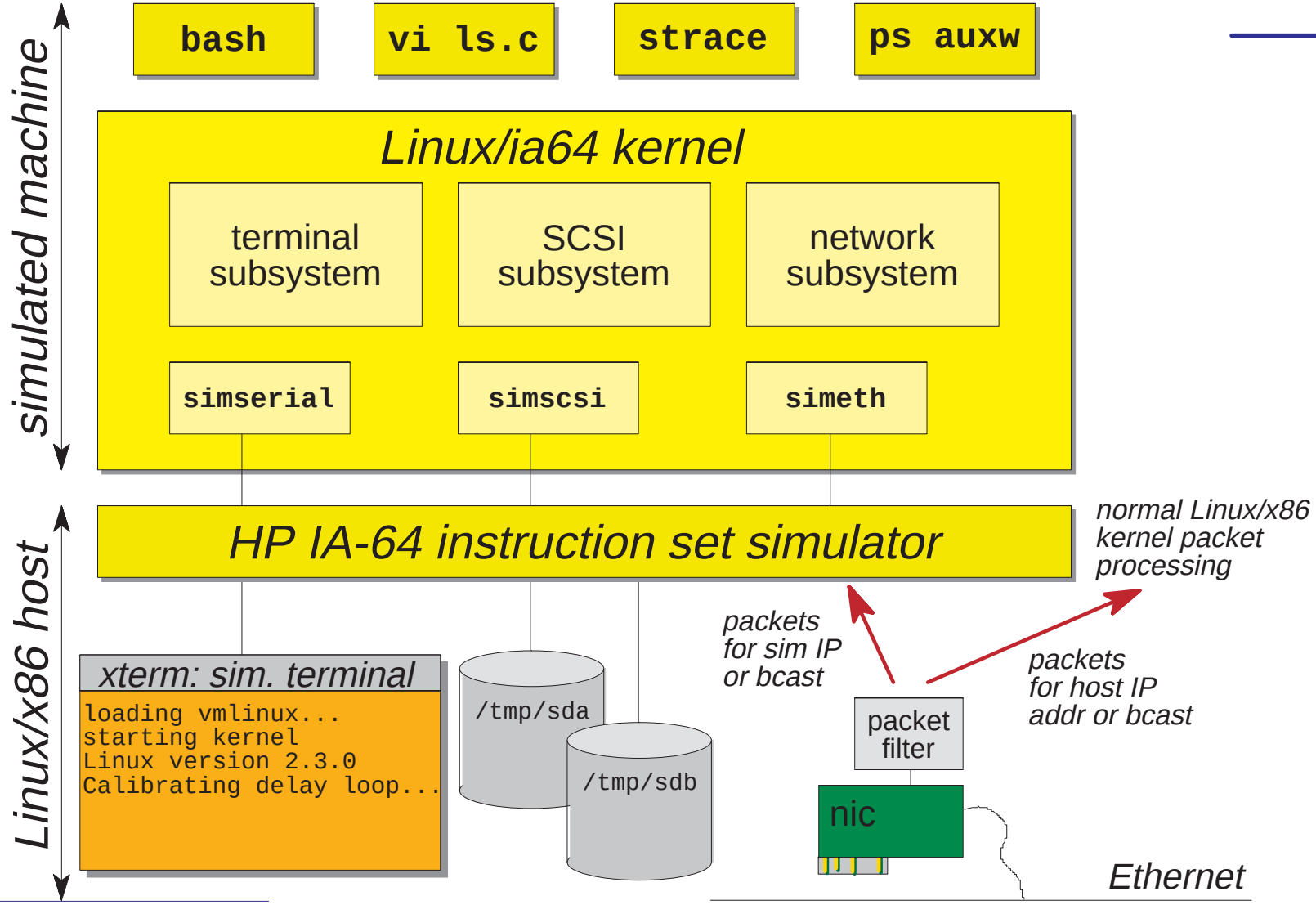
Conclusion

- Linux/ia64 is real
  - It is publicly available TODAY !
  - bootable CDROMs from some Linux distributors
- IA64 Linux SDK
  - software development without hardware: early access
- HP is committed to Linux
  - Major technology contributions (kernel, toolchain, simulator)
- Next steps:
  - debugging
  - tuning, performance optimizations
  - porting more applications
- Learn about the architecture
  - look at the linux/ia64 source code

Resources on the WEB

- IA-64 Architecture
  - http://www.hp.com/go/ia64
  - http://developer.intel.com/design/ia64
- HP & Linux
  - http://www.hp.com/go/linux/
- IA-64 Linux project
  - http://www.linuxia64.org/
- Papers & presentations
  - http://www.hpl.hp.com/techreports/
- IA-64 SDK discussion forum
    - http://devresource.hp.com/devresource/Topics/ia64linux.html

Kernel Simulation Environment



The Demo !