



LKST for SH updates - Let's Get Start with LKST/SH -



Lineo Solutions, Inc



Overview & Review Plenary Meeting, Jan. 2005



Presentation Overview

- LKST (Linux Kernel State Tracer) Porting to **SH4**
 - Renesas RTS7751R2D (CELF reference platform)
 - LKST kernel configuration
- Key Point of the Porting
 - Output **comparison** of x86 (Reference Model) vs. SH
- Examples for the LKST Output
- Demonstration
 - **Visualization technique of LKST output**



Our Motivation & Objective

- LKST as Debugging Tool
 - Powerful and efficient
 - Event tracing function is useful for trouble analysis
- Porting of Major Tracers (**LKST, LTT, ...**)
 - LTT: Version 0.9.5a supports x86, PPC and SH architecture
 - LKST: supports x86
- Contribution to Linux Improvements in Numerical Quantification Aspect
 - **Performance Evaluation** (Plans, exams and analyses with Visualization)
 - Supporting Performance Evaluation (Porting and/to integrated environments)



System Environments

- Hardware
 - RTS7751R2D(SH4)
- Software
 - Linux 2.6.8.1
 - LKST 2.2.1
 - GCC 3.2.3
- Cooperative Development for This System



System Concept and Hardware Support



Technical Advisory for LKST Technologies



System Construction



LKST Kernel (1/2)

- Setup LKST Kernel Configuration Environments
- linux-2.6.8.1 <http://www.kernel.org>
- Patches for LKST<<http://sourceforge.jp/projects/lkst/>>
 - lkst-2.2.1.tar.gz
 - lkstpatchset-2.2.1-for-2.6.8.1-2.tar.gz

LKST Kernel (2/2)

- LKST Kernel Configuration and Patch Application: STEPS
 - Expand linux-2.6.8.1.tar.gz
 - Expand lkst-2.2.1.tar.gz
 - Expand lkstpatchset-2.2.1-for-2.6.8.1-2.tar.gz
 - mv lkst-2.2.1/patches lkst-2.2.1/patches-2.6.9
 - mv patches-2.6.8.1 lkst-2.2.1/patches
 - make patch KPRESRC= <Kernel Expand Directory>

Port to SH CPU (1)

- Base: LKST kernel on i386
- Specific Points

File Name	Comment	Event Type
arch/sh/kernel/irq.c	Adds hook-points to do_IRQ()	INT_HARDWARE_ENTRY
arch/sh/kernel/process.c	Adds hook-points to kernel_thread()	PROCESS_LTHREAD_GEN
arch/sh/kernel/time.c	Adds cpu_khz variables initialization using time_init()	
arch/sh/mm/fault.c	Adds hook-points to do_page_fault()	LOOPS_PGFAULT
arch/sh/boot/compressed/misc.c	Adds #define __DISABLE_LKST_HOOK_	

Port to SH CPU (2)

File Name	Comment	Event Type
include/asm-sh/hook.h	SH Porting corresponding _IF_HOOK_ENABLED in i386	
include/asm-sh/hook_private.h	SH Porting Corresponding to is_asm_hook() in i386	
include/asm-sh/lkst.h	Changes Defined Value for LKST_BUFFER_SIZE_MAX (1MByte for Default)	
include/asm-sh/lkst_etype.h	Comment Out for SYSCALL_SYSENTER and SYSCALL_SYSEXIT SH Porting Corresponding to Atomic_read_and_add() in i386 SH Porting Corresponding to local_atomic_read_and_add() in i386	

Port to SH CPU (3)

File Name	Comment	Event Type
include/asm-sh/lkst_private.h	SH Porting Corresponding to lkst_evhandlerprim_mc() in i386	
include/asm-sh/timex.h	Adds extern Decralation for cpu_khz Adds hook-points to syscall_call	SYSCALL_ENTRY_HEADER
arch/sh/kernel/entry.S	Adds hook-points to syscall_exit Adds DEBUG_KERNEL Adds source "drivers/lkst/Kconfig" Adds config depends on config HOOK DEBUG_KERNEL	SYSCALL_EXIT_HEADER
arch/sh/Kconfig	Adds config ASM_HOOK	
include/asm-sh/hook*.h	config ASM_HOOK	



Port to SH CPU (4)

- Output Example for LKST

```
>>/root/lkstutils/lkst stat↓
press return key:↓
<Current status>↓
version of LKST           : 2.2.14
number of cpus           : 1↓
number of masksets       : 3↓
number of event-handlers: 3↓
current maskset_id       : 2↓
current writing buffer_id (cpu: 000): 0 ↓
>>/root/lkstutils/lkst stop↓
press return key:↓
Stop LKST event tracing.↓
>>/root/lkstutils/lkst start↓
press return key:↓
```



Port to SH CPU (5)

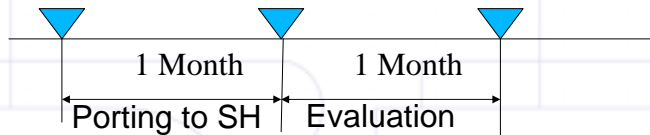
- Output Example of LKST

```
event_type=interrupt_hardware_entry↓
cpu=00, pid=00000410↓
time=Sat Jan 01 00:00:01.06051991 2000↓
arg1=0x00000010 00000000 : IRQ number irq ↓
arg2=0x00000001 00000000 : interrupt status status ↓
arg3=0x8f8f3e68 00000000 : pointer to register stack↓
↓
event_type=process_add_wait↓
cpu=00, pid=00000410↓
time=Sat Jan 01 00:00:01.06051275 2000↓
arg1=0x8f8f2c40 00000000 : pointer to wait_queue_head↓
arg2=0x8fe6a480 00000000 : pointer to added process↓
↓
event_type=context_switch↓
cpu=00, pid=00000000↓
time=Sat Jan 01 00:00:01.06051258 2000↓
arg1=0x8c21ba9c 00000000 : pointer to task_struct prev
arg2=0x8fe6a480 00000000 : pointer to task_struct next
arg3=0x00000000 00000000 : process state↓
arg4=0x00000000 00000000 : process count↓
```



Port to SH CPU (6)

- Porting was Smooth & Quick
 - Become Available ... about 1 month
 - Evaluation, Comparison with x86, Visualizing Tool ... another 1 month

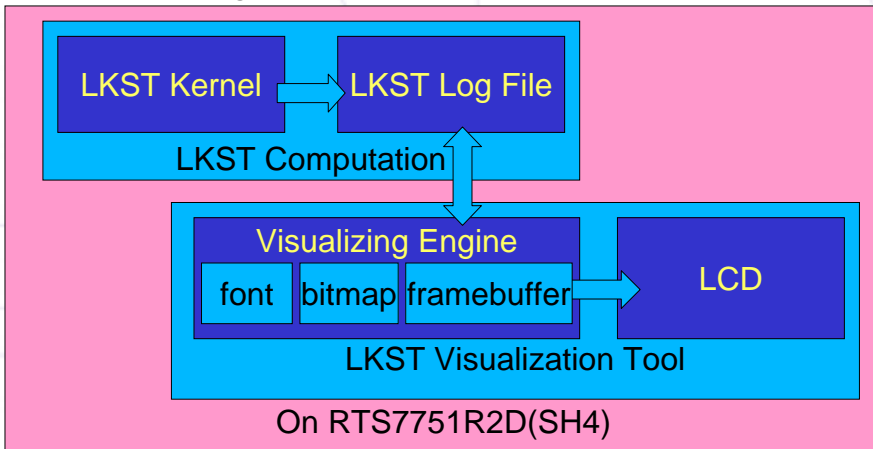


Demonstration

- Development of Visualization Tool for Output Results
 - From huge & complex output log text
To **BIOS-like Display**
 - Implemented on **Target Board**
 - Output Log can be checked right there on the target board.

Visualization Tool

- Block Diagram



Plenary Meeting on January 25, 2005
 Technical Jamboree on June 14, 2005

15

Visual Demo

- Video demo is available



Plenary Meeting on January 25, 2005
 Technical Jamboree on June 14, 2005

16



Summary

- Through LKST porting, we found;
 - High & smooth **portability**
 - Possible future approach: Usage as new **visualizing tool**
- Scopes in the future
 - **Port** other major/useful tracers
 - Stacking analysis
 - We Contribute to improve quality of Linux in its performance from the numerical quantification viewpoint.
 - Our challenge
 - Hook points to the system calls and exception processing



Updates Feb.-May, 2005

Presentation Overview

- Updates Feb.-May, 2005
 - Porting to SH
 - Tutorials and Practical Use
- **LKST vs LTT**
 - Hookpoints per Source File
- Tutorials
 - LKST Core
 - lkst and **lkstlogtools**
 - lkstlogtools Analyzers
- Hints for **Practical Use**
- Discussion
 - Binary Log File Interface

Updates Feb.-May, 2005

- **Porting to SH**
 - hookpoints to system calls --- done
 - Added hookpoints to exceptions --- done
 - LKST-SH patch for linux 2.6.8.1 --- done
 - LKST-SH patches for linux-2.6.9 --- done
 - Our Challenges
 - Clock Source --- where to choose
 - Submit to Community (LKST project of sourceforge)
 - Kernel Trace and its Objectives
 - Obstacle Analysis Viewpoints
 - Debugging Viewpoints
 - Performance Analysis Viewpoints

Updates Feb.-May, 2005

- Tutorials and Practical Use
 - Studying Introducing Procedure --- In Progress
 - Introduction
 - How to install
 - Analysis Guideline
 - Practical Use --- In Progress
 - Trial on **FC2 as Reference** (linux2.6.9-lkst2.2.1)
 - see lkst-2.2.1/howto.txt in detail
 - Training for **lkstlogtools**
 - Stacking Tracing Results --- Our Challenge
- Continual Discussion
 - via E-mail & (tele)conferences Cooperatively
 - Topics; Study **LKST vs LTT**
 - **Hookpoints comparison**

LKST vs LTT - Hookpoints

Source File	LKST	LTT
Process Management		
./kernel/sched.c	PROCESS_CONTEXTSWITCH	LTT_EV_SCHEDCHANGE
	PROCESS_WAKEUP	
./arch/i386/kernel/process.c	PROCESS_LTHREADGEN	LTT_EV_PROCESS
./kernel/fork.c		LTT_EV_PROCESS
./kernel/exit.c,		LTT_EV_PROCESS
./kernel/signal.c,	PROCESS_SIGSEND	LTT_EV_PROCESS
../include/linux/wait.h	PROCESS_INIT_WQH PROCESS_ADD_WQ PROCESS_REM_WQ	LTT_EV_PROCESS
./arch/i386/kernel/traps.c,		LTT_EV_TRAP_ENTRY
./arch/i386/mm/fault.c		LTT_EV_TRAP_EXIT

LKST vs LTT - Hookpoints

Source File	LKST	LTT
Hardware Interrupts		
./arch/i386/kernel/irq.c	INT_HARDWARE_ENTRY	LTT_EV_IRQ_ENTRY LTT_EV_IRQ_EXIT
Software Interrupts		
./kernel/softirq.c	INT_TASKLETHI_ENTRY INT_TASKLET_ENTRY INT_BH_ENTRY	LTT_EV_SOFT_IRQ
Exceptions		
./arch/i386/kernel/entry.S	EXCEPTION_ENTRY EXCEPTION_EXIT	/
System Calls		
./arch/i386/kernel/entry.S	SYSCALL_*	LTT_EV_SYSCALL_ENTRY LTT_EV_SYSCALL_EXIT

Technical Jamboree on June 14, 2005

LKST vs LTT - Hookpoints

Source File	LKST	LTT
Memory Management		
./mm/vmscan.c	MEM_SWAPOUT	/
./mm/memory.c	MEM_SWAPIN MEM_DO_NOPAGE MEM_DO_WPPAGE	LTT_EV_MEMORY
./mm/page_io.c	/	LTT_EV_MEMORY
./mm/filemap.c	MEM_WAIT_PAGE	LTT_EV_MEMORY
./mm/page_alloc.c	MEM_GET_FREEPAGE MEM_GET_ZEROPAGE MEM_FREEPAGE	/
./mm/vmalloc.h	MEM_VMALLOC MEM_VFREE	/
./mm/slab.c	MEM_CACHE_CREATE MEM_CACHE_ALLOC / MEM_MALLOC MEM_CACHE_FREE / MEM_FREE	/

Technical Jamboree on June 14, 2005

LKST vs LTT - Hookpoints

Source File	LKST	LTT
Filesystem		
./drivers/block/l1_rw_blk.c	FS_DEVRW	
./fs/buffer.c	FS_DEVEND FS_BUFBUSY	LTT_EV_FILE_SYSTEM_BUF_WAIT_START LTT_EV_FILE_SYSTEM_BUF_WAIT_END
./fs/exec.c		LTT_EV_FILE_SYSTEM_EXEC
./fs/open.c		LTT_EV_FILE_SYSTEM_OPEN LTT_EV_FILE_SYSTEM_CLOSE
./fs/read_write.c		LTT_EV_FILE_SYSTEM_READ LTT_EV_FILE_SYSTEM_WRITE LTT_EV_FILE_SYSTEM_SEEK
./fs/ioctl.c		LTT_EV_FILE_SYSTEM_IOCTL
./fs/select.c		LTT_EV_FILE_SYSTEM_SELECT LTT_EV_FILE_SYSTEM_POLL

Technical Jamboree on June 14, 2005

25

LKST vs LTT - Hookpoints

Source File	LKST	LTT
Network		
./net/core/dev.c	NET_PKTSEND NET_PKTSENDI NET_PKTRECV NET_PKTRECVI	LTT_EV_NETWORK
./net/socket.c	NET_SOCKETIF	LTT_EV_SOCKET
./i386/kernel/sys_i386.c		LTT_EV_IPC
SysV Inter-process Communication		
./ipc/sem.c	SYSV_IPC_SEM*	
./ipc/msg.c	SYSV_IPC_MSG*	
./ipc/shm.c	SYSV_IPC_SHM*	
Spinlocks		
./include/asm-i386/spinlock.h	LK_*	

Technical Jamboree on June 14, 2005

26

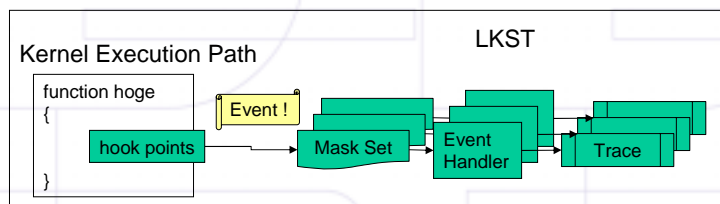
LKST vs LTT - Hookpoints

Source File	LKST	LTT
Timer		
./kernel/timer.c	TIMER_*	LTT_EV_KERNEL_TIMER LTT_EV_TIMER
Oops		
./arch/i386/mm/fault.c	OOPS_PGFAULT	
./arch/i386/kernel/nmi.c	OOPS_NMIWDOG	
Others (I/O commands, Panic, printk)		
./include/asm-i386/io.h	O_PORTIN O_PORTOUT	
./kernel/panic.c	O_PANIC	
./kernel/printk.c	O_PRINTK	

27

Tutorials; LKST Core

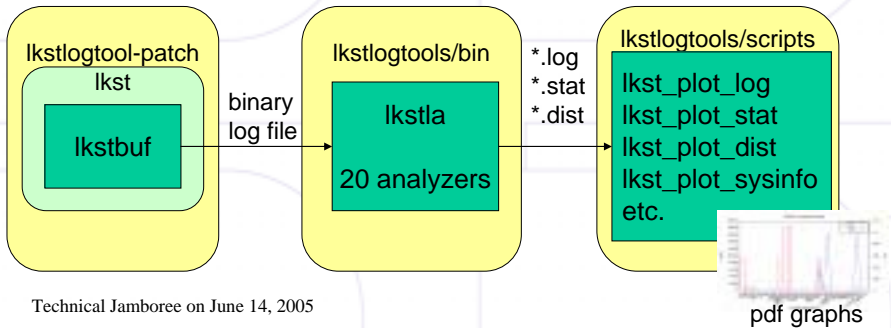
- Block Diagram



28

Tutorials; Ikst and Ikstlogtools

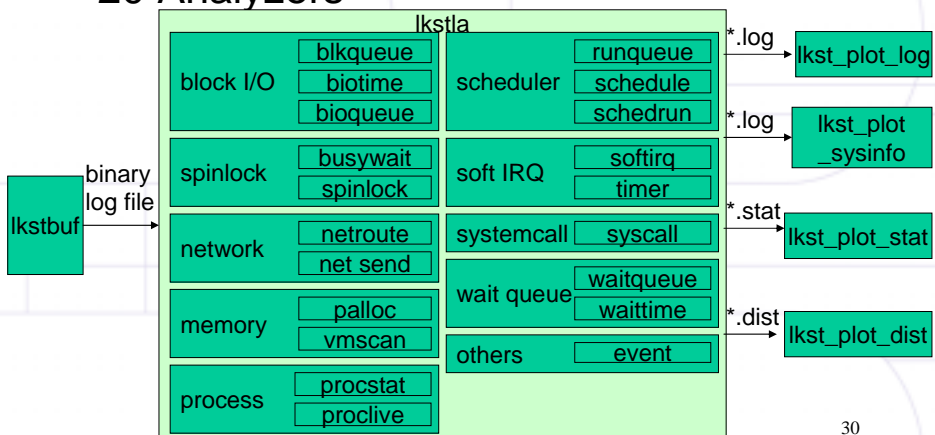
- Ikst ... Tracing Processing
- Ikstlogtools ... Postprocessing
 - developed by LKST Core Team



Technical Jamboree on June 14, 2005

Tutorials; Ikstlogtools Analyzers

- 20 Analyzers



Technical Jamboree on June 14, 2005



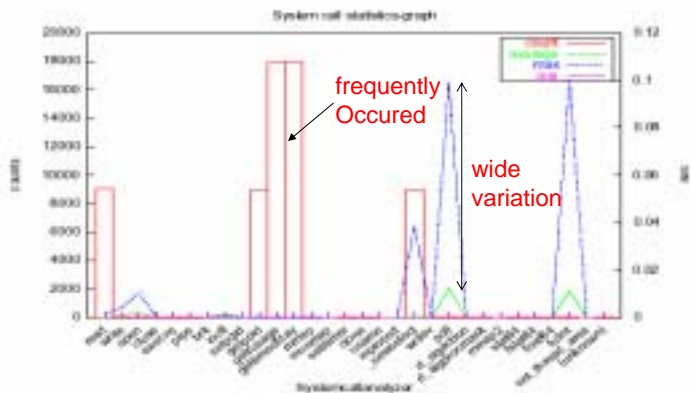
Hints for Practical Use(1)

- biotime.log
 - start [sec] vs blockio-time
- proclive.log
 - pid vs task_name vs start [sec] vs living-time
- procstat.log
 - pid vs task_name vs start [sec] vs process-stat
- runqueue.log
 - start [sec] vs runqueue-length
- systemcall.log
 - syscall_no vs syscall_name vs start[sec] vs processing-time
- sysinfo.log
 - time vs free vs buffer vs shared vs freefiles vs unused_inodes vs inodes



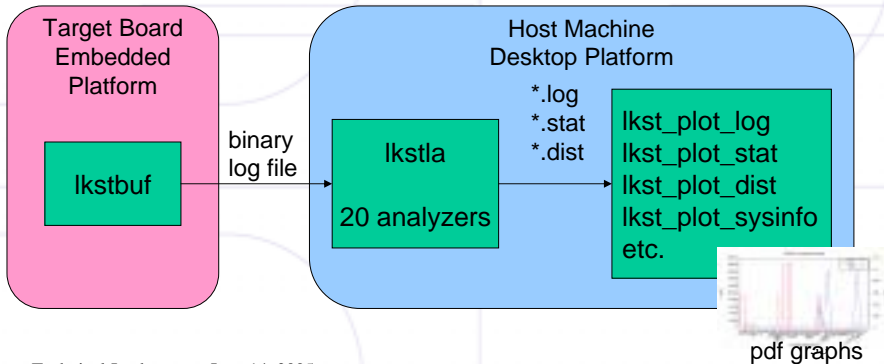
Hints for Practical Use (2)

- System Call Statistics Graph



discussion; binary log file I/F

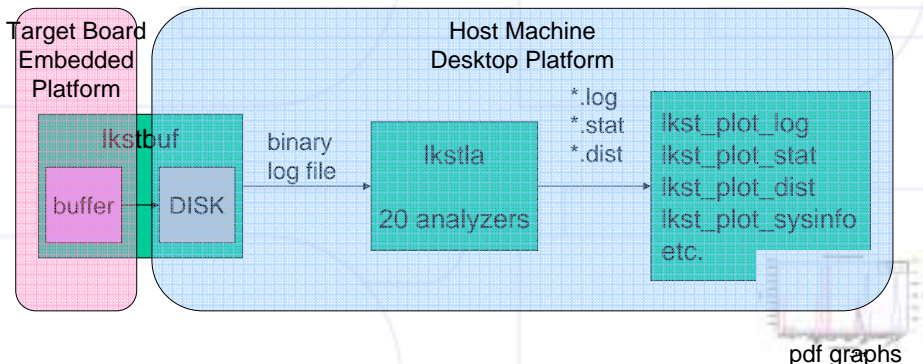
- Suitable for Embedded LKST Environments?



Technical Jamboree on June 14, 2005

discussion; binary log file I/F

- Better Solution?



Technical Jamboree on June 14, 2005



Summary

- Porting to SH is almost ready
- **LKST** vs **LTT**
 - Hookpoints per Source File Compared
- Tutorials for LKST Core + **lkstlogtools** Analyzers
- Hints for **Practical Use**
- Discussion
 - Binary Log File Interface for Embedded Environments



Thank You!