



LKST for SH



Lineo Solutions, Inc



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_EN TRY
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           : 2.2.1↓  
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)

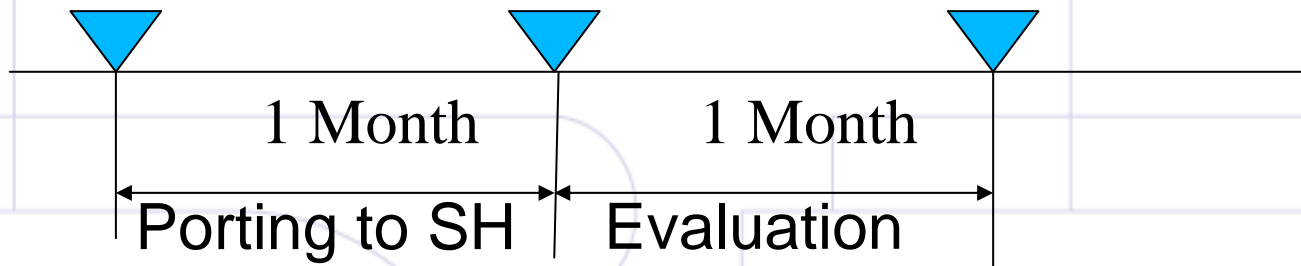
- Output Example of LKST

```
event_type=interrupt_hardware_entry↓
  cpu=00, pid=00000410↓
  time=Sat Jan 01 00:00:01.060551991 2000↓
  arg1=0x00000010 00000000 : IRQ number irq ↓
  arg2=0x00000001 00000000 : interrupt status status ↓
  arg3=0x8f8f3e68 00000000 : pointer to register stack↓
↓
event_type=process_add_waitq↓
  cpu=00, pid=00000410↓
  time=Sat Jan 01 00:00:01.060551275 2000↓
  arg1=0x8f8f3c40 00000000 : pointer to wait_queue_head↓
  arg2=0x8fe6a460 00000000 : pointer to added process↓
↓
event_type=context_switch↓
  cpu=00, pid=00000000↓
  time=Sat Jan 01 00:00:01.060551258 2000↓
  arg1=0x8c21ba9c 00000000 : pointer to task_struct prev
  arg2=0x8fe6a460 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





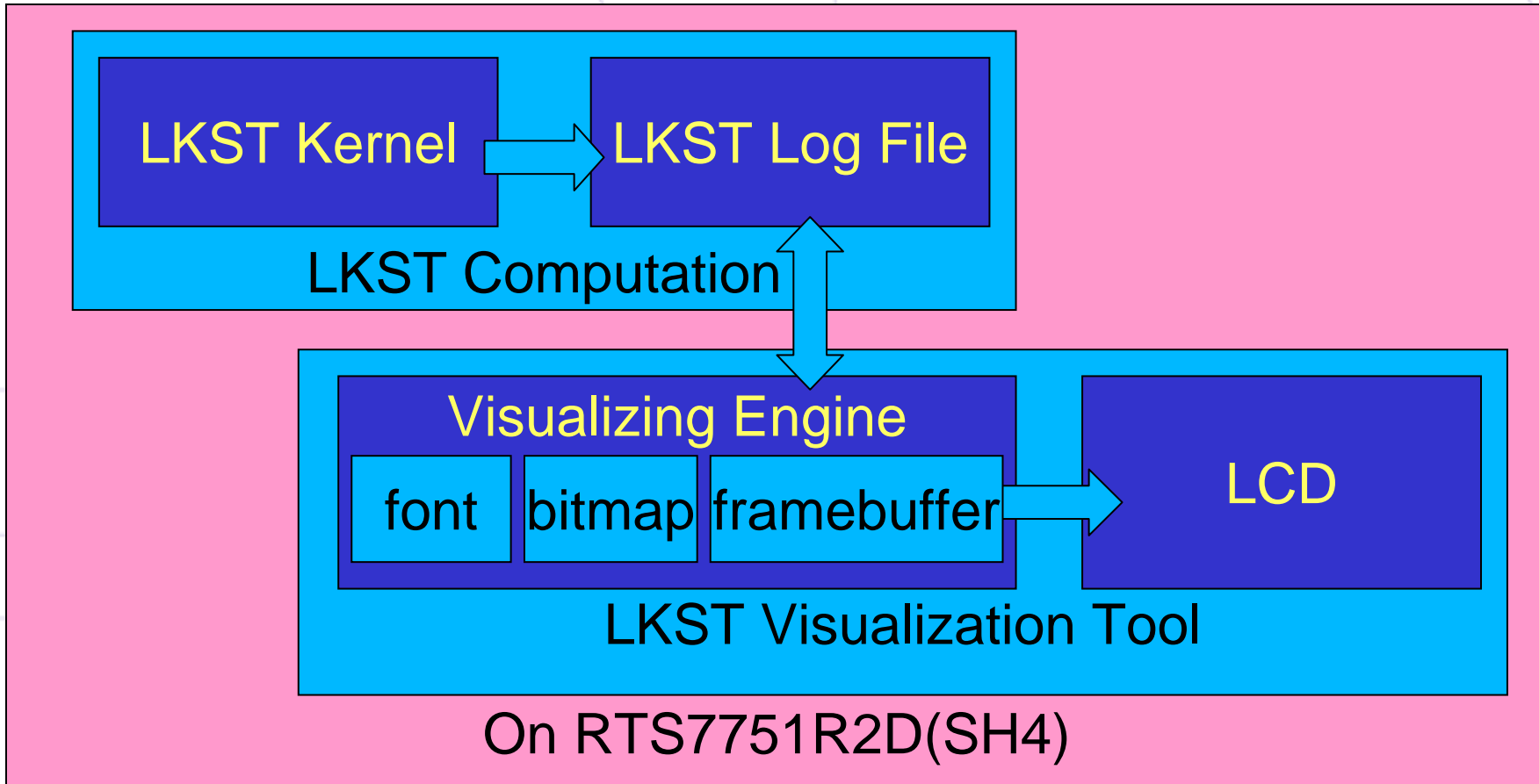
Demonstration

- 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





Visual Demo

- Video demo is available





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



Thank You!

ATTENTION

Corresponding Demonstration

Jan. 25 17:00-18:30