Practical Experiences With Software Crash Analysis in TV

Wim Decroix/Yves Martens, TPVision

Wim Decroix/Yves Martens
12 November 2012
Overview

• Context
• One-slide overview of tooling approach
• Practical usecases
Context: SPACE SW Architecture

**SPlit Application arChitecturE**

- Applications are isolated in dedicated processes
- The resources in the system are explicitly and centrally managed
- The client applications are system context unaware
- The lifecycle, focus and visual layout of the client applications is centrally managed
Problem: Increasing # SW crashes
Problem: Increasing # SW crashes

- Increasing number of SW crashes found during QA testing
- Lot of effort spent to analyse and solve these
- Leads to longer time to mature SW
- Leads to longer time to market
Crash Analysis and Resolution Flow

Crash seen -> Problem Report

Developer 1 analyses which app/…

Developer 2 analyses root cause

Developer 2 solves crash

Normal Flow

Desired Flow

Wim Decroix/Yves Martens
12 November 2012
How to reach desired flow

• Detect crashes in SW
• Dump as much of information on crash as possible
• Good analysis and visualization tools of dumps
  – To dispatch to correct developer
  – To indicate root cause
• Continuously improve tooling
  – Analyse problems in the long flow
  – Investigate what can be changed to reach the desired flow
Roles

• QA testing
  – Black box functional tests, duration tests (overnight, weekend)
  – File problem reports with attached post mortem dumps.

• Developers
  – Write functional code, follow SPACE architecture
    • following execution architecture rules
  – Debug using post mortem info / debug tooling

• “Execution Architecture” team
  – Create tooling for post mortem info & debug
  – Blessed with solving the ‘hard’ crashes
  – Set execution architecture rules
  – Special thanks for their contribution to this presentation!
    • Nico De Ceulaer, Pieter Van Loocke, …
One slide overview

• Fault Detection
  – Signal detection
  – Watchdogs
  – Kernel crashes

• Fault Information Reporting
  – UART, various circular buffers capturing events, stack backtracing, standard Linux info, memory usage monitoring

• Fault Storage
  – NAND Flash post mortem dumps, UART logs, EPROM NVM

• Fault Recovery
  – Various degrees of reboot

• Fault Retrieval
  – Via USB, service, ethernet…

• Fault Analysis
  – Website for dump translation, PR (cross-)analysis
  – TimeDoctor Visualisation
Examples of software crashes

1. Signals
2. Watchdog (worker thread not responding)
3. Watchdog (blocked through other thread)
4. Watchdog (CPU overload)
5. Watchdog (deadlock)
6. Linux Out of Memory
Example 1: SIGSEGV/… backtrace

• Crash:
  – Null pointer dereference/…

• Detect:
  – Install signal handler in all applications

• Dump:
  – Stacktrace of thread causing signal
  – Dump on internal flash
  – Testers copy dump from flash and attach to problem report

• Analysis
  – Automatic translation of backtrace in problem report
Example 1: SIGSEGV/… backtrace

- Standard solution: using glibc backtrace in signal handler
- Proprietary kernel based backtrace used
- Mainly due to historical reasons
- Other advantages:
  - Combines userspace and kernel space stack
  - Backtrace also available when in uninterruptable sleep or when signals are blocked
**Example 1: SIGSEGV/... backtrace**

<table>
<thead>
<tr>
<th>Function</th>
<th>File/Line</th>
<th>File/Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>context_switch</td>
<td>vmlinux_app_flash</td>
<td>linux-2.6.35.13/kernel/sched.c:2920</td>
</tr>
<tr>
<td>schedule</td>
<td>vmlinux_app_flash</td>
<td>linux-2.6.35.13/kernel/sched.c:3370</td>
</tr>
<tr>
<td>futex_wait_queue_me</td>
<td>vmlinux_app_flash</td>
<td>linux-2.6.35.13/kernel/futex.c:1702</td>
</tr>
<tr>
<td>futex_wait</td>
<td>vmlinux_app_flash</td>
<td>linux-2.6.35.13/kernel/futex.c:1820</td>
</tr>
<tr>
<td>do_futex</td>
<td>vmlinux_app_flash</td>
<td>linux-2.6.35.13/kernel/futex.c:2561</td>
</tr>
<tr>
<td>sys_futex</td>
<td>vmlinux_app_flash</td>
<td>linux-2.6.35.13/kernel/futex.c:2636</td>
</tr>
<tr>
<td>handle_sys</td>
<td>vmlinux_app_flash</td>
<td>linux-2.6.35.13/arch/mips/kernel/scall2-o32-S:59</td>
</tr>
<tr>
<td>__pthread_cond_wait</td>
<td>libpthread-2.13</td>
<td>glibc/pthread_cond_wait.c:156</td>
</tr>
<tr>
<td>SendAndReceive</td>
<td>liberrlib</td>
<td>space/plf/vpl/ceos/comp/ceoserrlibpriv/errlib_incom.c:306</td>
</tr>
<tr>
<td>errlib__drmpcom_DumpException</td>
<td>liberrlib</td>
<td>space/plf/vpl/ceos/comp/ceoserrlibpriv/errlib_incom.c:526</td>
</tr>
<tr>
<td>ExceptionHandler</td>
<td>liberrlib</td>
<td>space/plf/vpl/ceos/comp/ceoserrlibpriv/errlib_rndetector.c:1013</td>
</tr>
<tr>
<td>bxst_AddToWanted</td>
<td>btApp</td>
<td>space/app/bx/Hsvbc/comp/bxmlf/comp/bxsto/bxsto_requestlist.c:198</td>
</tr>
<tr>
<td>bxst_MovePageToWanted</td>
<td>btApp</td>
<td>space/app/bx/Hsvbc/comp/bxmlf/comp/bxsto/bxsto_requestlist.c:626</td>
</tr>
<tr>
<td>bxsto__req_SuggestPage</td>
<td>btApp</td>
<td>space/app/bx/Hsvbc/comp/bxmlf/comp/bxsto/bxsto_pagestore.c:2330</td>
</tr>
</tbody>
</table>

Wim Decroix/Yves Martens
12 November 2012
Example 1: SIGSEGV/… backtrace

Backtrace:

```c
bpst_AddToWanted  txtApp space/app/bxt/Hsvbxt/comp/bxplf/comp/bxsto/bxsto_requestlist.c:198
bpst_MovePageToWanted txtApp space/app/bxt/Hsvbxt/comp/bxplf/comp/bxsto/bxsto_requestlist.c:626
txsto__req_SuggestPage txtApp space/app/bxt/Hsvbxt/comp/bxplf/comp/bxsto/bxsto_pagestore.c:2330
```

Patch:

```c
190a191,192
>     if(pckt_ptr != NULL)
>     {
197a200,201
>         if(next_ptr != NULL)
>         {
198a203
>             
208a214
>         }
```
Example 2: Watchdog (worker thread not responding)

- TV software works with worker threads (called pump engines) executing tasks (called pumps).
- All tasks must be finished within certain amount of time.
- Otherwise this is assumed to be an error and the TV is restarted.
- Reason: Avoid that the user manually has to unplug and replug the TV to recover from endless running task.
Example 2: Watchdog (worker thread not responding)

- **Crash:**
  - Watchdog due to task in endless loop/…
- **Detect:**
  - Every worker thread feeds watchdog
- **Dump:**
  - Dump backtrace of worker thread that caused watchdog
    (standard solution: raise signal to thread causing watchdog)
- **Analysis:**
  - Translate backtrace
Example 2: Watchdog (worker thread not responding)

Wim Decroix/Yves Martens
12 November 2012
Example 3: Watchdog (blocked on other thread)

• Crash:
  – Watchdog due to task waiting on another task
• Detect:
  – Every worker thread feeds watchdog
• Dump:
  – Dump backtrace of all threads (standard solution: raise signal to all threads)
  – Dump trace of last events in system
• Analysis:
  – Translate backtrace
  – Visualize last x seconds
Example 3: Watchdog (blocked on other thread)

- Proprietary tracing format which logs
  - In kernel (using tracepoint):
    - Task switches in kernel
    - Interrupt handling
    - Syscalls
    - Signals
  - In userspace:
    - Pump execution
    - RPC calls between applications
- Modified timedoctor is used for visualization
  - http://sourceforge.net/projects/timedoctor/
Example 3: Watchdog (blocked on other thread)

TODO: add timedoctor
Example 3: Watchdog (blocked on other thread)

TODO: add timedoctor

12 November 2012

Wim Decroix/Yves Martens
Example 3: Watchdog (blocked on other thread)
Example 3: Watchdog (blocked on other thread)
## Example 3: Watchdog (blocked on other thread)

<table>
<thead>
<tr>
<th>Function/Module</th>
<th>Time Last Run</th>
<th>Current Pump</th>
<th>pump_send</th>
<th>Pump Runs</th>
<th>IPC Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>isilib low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>netvApp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERROR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>context_switch</td>
<td>14s 675ms</td>
<td></td>
<td>4</td>
<td>4 (30)</td>
<td></td>
</tr>
<tr>
<td>schedule</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isilib</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OneQueue_Receive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IComComponent_One_Call</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>icspifaconn___ufsroidp_CcloselUr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>elsmmedobj_tmlUmsic2_CloseUrl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>elsmmedobj_vpl_CloseUrl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>olajapi_vplN_OnEndOIRendering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NfHandler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EngineTask</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>--------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>context_switch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>schedule</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>futex_wait_queue_me</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>futex_wait</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>do_futex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sys_futex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>handle_sys</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>___lock_wait</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>__pthread_mutex_lock</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>csgethoststruct2__xattr_MutexLock</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>elsmmedobj_mediate_GetCurrentState</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Wim Decroix/Yves Martens
12 November 2012
Example 3: Watchdog (blocked on other thread)

12 November 2012

Wim Decroix/Yves Martens
Example 4: Watchdog (CPU overload)

- Crash:
  - Watchdog due to task taking longer due to CPU consumed by other threads
- Detect:
  - Every worker thread feeds watchdog
- Dump:
  - Dump backtrace of all threads (standard solution: raise signal to all threads)
  - Dump trace of last events in system
- Analysis:
  - Translate backtrace
  - Visualize last x seconds
Example 4: Watchdog (CPU overload)

Wim Decroix/Yves Martens
12 November 2012
Example 5: Watchdog (deadlock)

- Crash:
  - Watchdog due to task blocked on lock
- Detect:
  - Every worker thread feeds watchdog
- Dump:
  - Dump lock status as much as possible
- Analysis:
  - Show status of locks in backtraces
Example 5: Watchdog (deadlock)

- context_switch
- schedule
- ioctl
- fusion_skirmish_prevail
- fd_messenger_lock
- IFusionDaleMessage_AllocateData
- hsvprefixhsvchanlib__sendevt_SendEvent
- hsvprefixhsvpgdat__pgdat_Update
- mSetScalarAttributeNat64
- hsvprefixhsvpgdat__pgdat3_StoreChannelDataNat64
- CreateChannelNode
- AddNodesInParts
- PumpHandler
- Execute
- EngineTask

vmlinux_app_flash
libfusision-1.6
libfusisiondale-1.6
libhsvchanlib
libhsvchanlib
libhsvchanlib
libhsvchanlib
libcesgpi
libcesgpi

vmlinux_app_flash
libc-2.13
0x14

linux-2.6.35.13/kernel/sched.c:2920
linux-2.6.35.13/kernel/sched.c:3870
dfb/core/DirectFB.git/lib/fusion/lock.c:142
dfb/libs/FusionDale.git/src/core/messenger.h:141
dfb/libs/FusionDale.git/src/messenger/fusiondalemessage.c:304
space/app/eu/hsvchanlib/prod/hsvchanlib/hsvchanlib_m.c:623
space/app/eu/hsvchanlib/comp/hsvpgdat/hsvpgdat_mpgdb.c:425
space/app/eu/hsvchanlib/comp/hsvpgdat/hsvpgdat_mpgdb.c:1255
space/app/eu/hsvchanlib/comp/hsvpgdat/hsvpgdat_mpgdb.c:1197
space/app/eu/euint/comp/eucontentlib/eucontentlib_m.c:1757
space/app/eu/euint/comp/eucontentlib/eucontentlib_m.c:670
space/app/eu/euint/comp/eucontentlib/eucontentlib_m.c:827
space/pltf/tvplf/clos/comp/cesgipmpi/cesgipmpi_m.c:841
space/pltf/tvplf/clos/comp/cesgipmpi/cesgipmpi_m.c:1045
## Example 5: Watchdog (deadlock)

<table>
<thead>
<tr>
<th>Function</th>
<th>Source Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>context_switch</code></td>
<td>vmlinux_app_flash</td>
</tr>
<tr>
<td><code>schedule</code></td>
<td>vmlinux_app_flash</td>
</tr>
<tr>
<td><code>futex_wait_queue_me</code></td>
<td>vmlinux_app_flash</td>
</tr>
<tr>
<td><code>futex_wait</code></td>
<td>vmlinux_app_flash</td>
</tr>
<tr>
<td><code>do_futex</code></td>
<td>vmlinux_app_flash</td>
</tr>
<tr>
<td><code>sys_futex</code></td>
<td>vmlinux_app_flash</td>
</tr>
<tr>
<td><code>handle_sys</code></td>
<td>vmlinux_app_flash</td>
</tr>
<tr>
<td><code>__ill_lock_wait</code></td>
<td>libpthread-2.13</td>
</tr>
<tr>
<td><code>__pthread_mutex_lock</code></td>
<td>libpthread-2.13</td>
</tr>
<tr>
<td><code>coepgitphosttk2__pmtk_ProcessSharedSemAcquire</code></td>
<td>glibc/pthread_mutex_lock - 82</td>
</tr>
<tr>
<td><code>GlobalLock</code></td>
<td>libcoepgi</td>
</tr>
<tr>
<td><code>ca__ca_ReadNode</code></td>
<td>libca</td>
</tr>
<tr>
<td><code>eucontentlib__pgdatN_OnChannelRemoved</code></td>
<td>euApo - space/app/eu/euinit/comp/eucontentlib.eucontentlib_m.c:1188</td>
</tr>
<tr>
<td><code>mChannLibLibEventListen</code></td>
<td>libhsvchanlib</td>
</tr>
<tr>
<td><code>fd_messenger_port_reaction</code></td>
<td>libfusion-daemon-1.6</td>
</tr>
<tr>
<td><code>fusion_reactor_process_message</code></td>
<td>libfusion-1.6</td>
</tr>
<tr>
<td><code>fusion_dispatch_loop</code></td>
<td>libfusion-1.6</td>
</tr>
</tbody>
</table>

Wim Decroix/Yves Martens
12 November 2012
Example 6: Out Of Memory

• Problem:
  – Standard Linux OOM killer would remove forensic evidence from debug dumps. We must collect all info before it kicks in

• Detect:
  – Poll free memory (minus buffers and cached)
    • Heuristic value of required free memory defined

• Dump:
  – Dump memory status of all applications and libraries
    • Proprietary tool parsing kpagemap
### Example 6: Out Of Memory

<table>
<thead>
<tr>
<th>owner</th>
<th>unknown</th>
<th>code</th>
<th>data</th>
<th>heap</th>
<th>bss</th>
<th>stack</th>
<th>shared</th>
<th>file</th>
<th>other</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordApp</td>
<td>0.0</td>
<td>20.0</td>
<td>8.0</td>
<td>1068.0</td>
<td>0.0</td>
<td>200.0</td>
<td>0.0</td>
<td>0.0</td>
<td>60.0</td>
<td>1356.0</td>
</tr>
<tr>
<td>ytlbApp</td>
<td>0.0</td>
<td>0.0</td>
<td>12.0</td>
<td>280.0</td>
<td>0.0</td>
<td>160.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>452.0</td>
</tr>
<tr>
<td>epgApp</td>
<td>0.0</td>
<td>72.0</td>
<td>24.0</td>
<td>2684.0</td>
<td>0.0</td>
<td>257.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3988.0</td>
<td>7025.0</td>
</tr>
<tr>
<td>txtApp</td>
<td>0.0</td>
<td>16.0</td>
<td>12.0</td>
<td>1420.0</td>
<td>0.0</td>
<td>272.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1552.0</td>
<td>3272.0</td>
</tr>
<tr>
<td>oskbApp</td>
<td>0.0</td>
<td>8.0</td>
<td>8.0</td>
<td>1424.0</td>
<td>0.0</td>
<td>180.0</td>
<td>0.0</td>
<td>0.0</td>
<td>16.0</td>
<td>1637.0</td>
</tr>
<tr>
<td>upgApp</td>
<td>0.0</td>
<td>12.0</td>
<td>12.0</td>
<td>1412.0</td>
<td>0.0</td>
<td>192.0</td>
<td>0.0</td>
<td>0.0</td>
<td>60.0</td>
<td>1688.0</td>
</tr>
<tr>
<td>dfuApp</td>
<td>0.0</td>
<td>8.0</td>
<td>4.0</td>
<td>2832.0</td>
<td>0.0</td>
<td>172.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>3020.0</td>
</tr>
<tr>
<td>nettvApp</td>
<td>0.0</td>
<td>196.0</td>
<td>28.0</td>
<td>241264.0</td>
<td>0.0</td>
<td>1329.0</td>
<td>16.0</td>
<td>0.0</td>
<td>820.0</td>
<td>243653.0</td>
</tr>
<tr>
<td>ecdApp</td>
<td>0.0</td>
<td>12.0</td>
<td>8.0</td>
<td>208.0</td>
<td>0.0</td>
<td>180.0</td>
<td>0.0</td>
<td>0.0</td>
<td>80.0</td>
<td>488.0</td>
</tr>
<tr>
<td>playApp</td>
<td>0.0</td>
<td>12.0</td>
<td>28.0</td>
<td>1936.0</td>
<td>0.0</td>
<td>252.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1196.0</td>
<td>3425.0</td>
</tr>
</tbody>
</table>

### Libraries

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>-libdl-2.13.so</td>
<td>0.0</td>
<td>0.0</td>
<td>88.0</td>
<td>0.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>92.0</td>
</tr>
<tr>
<td>-librt-2.13.so</td>
<td>0.0</td>
<td>4.0</td>
<td>84.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>88.0</td>
</tr>
<tr>
<td>-libgcc_s.so.1</td>
<td>0.0</td>
<td>19.0</td>
<td>84.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>103.0</td>
</tr>
<tr>
<td>-libpthread-2.13.so</td>
<td>0.0</td>
<td>69.0</td>
<td>88.0</td>
<td>0.0</td>
<td>88.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>245.0</td>
</tr>
<tr>
<td>-libutil-2.13.so</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
</tr>
<tr>
<td>-libcrypt-2.13.so</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
</tr>
<tr>
<td>-libnsl-2.13.so</td>
<td>0.0</td>
<td>4.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>12.0</td>
</tr>
<tr>
<td>-libnss_files-2.13.so</td>
<td>0.0</td>
<td>4.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>12.0</td>
</tr>
<tr>
<td>-libnss_nis-2.13.so</td>
<td>0.0</td>
<td>4.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>12.0</td>
</tr>
<tr>
<td>-libc-2.13.so</td>
<td>0.0</td>
<td>617.0</td>
<td>251.0</td>
<td>0.0</td>
<td>340.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1208.0</td>
</tr>
<tr>
<td>-ld-2.13.so</td>
<td>0.0</td>
<td>99.0</td>
<td>227.0</td>
<td>0.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>334.0</td>
</tr>
</tbody>
</table>
Summary

• Reduce Time-To-Market by efficient software crash analysis
• Add information to crash dumps
• Improve analysis
• Continuous improvement based on experience
Questions?

It's QUESTION TIME!!
Appendix: Resources

• jointSPACE
  – http://jointspace.sourceforge.net/

• TimeDoctor
  – http://sourceforge.net/projects/timedoctor/