WrtVMM: A Virtual Machine Monitor for Embedded Devices

Aaron Gember
Xixuan Feng
Yueh-Hsuan Chiang
Motivation

- Increasingly powerful embedded devices
- Few VMMs direct to embedded devices
- Virtual machine benefits
  - Isolation between routing and applications
  - Move and clone services between routers
  - Elimination of additional hardware
WRT54GL & OpenWrt (Host)

- Linksys WRT54GL Wireless Router
  - 200 MHz MIPS Processor
  - 16 MB RAM / 4 MB Flash
  - Added serial ports

- OpenWrt (Host)
  - 2.6.30 Linux kernel
  - Designed for small networking hardware
Embedded Xinu (Guest)

- OS designed for education and research
- Provides
  - Process management
  - Memory management
  - Serial I/O
- Disabled
  - Network stack
  - Memory protection (new feature)
Architecture

- Host OS Kernel
  - vmm-module
  - vmm-launch
- Guest OS
- Hardware
- user space
- kernel space
Memory Virtualization

- VMM Module Supports Memory Allocation
  - executable memory for running guest OS
  - special memory for guest OS registering interrupt handler (VCPU)
Interrupt Handling

- Conventional Interrupt
  - Hardware
  - Kernel
  - User Space

- with Virtualization
  - Hardware
  - Module
  - Kernel
  - VCPU
  - Guest OS
  - Host OS
  - wrtVMM

1. Hardware
2. Module
3. Kernel
4. Kernel
5. VCPU
6. Signal handler
7. handler
Privileged Instructions

- Use Built-in SIGILL

Examples

- `mtc0 t0, $12`
  - move `t0` to coprocessor 0 status register
  - emulate it by move to VCPU

- `mfc0 t0, $12`
  - Correctly move from VCPU
## Proc/Mem Benchmarking

- **CPU Intensive – N Queens Problem**

<table>
<thead>
<tr>
<th>Config</th>
<th>OpenWRT only</th>
<th>Xinu only</th>
<th>Xinu as Guest OS</th>
<th>Host &amp; Guest at the Same time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (ms)</td>
<td>5.39</td>
<td>13.74</td>
<td>13.93</td>
<td>27.91</td>
</tr>
</tbody>
</table>

- **Memory Intensive – Array Summing**

<table>
<thead>
<tr>
<th>Config</th>
<th>OpenWRT only</th>
<th>Xinu only</th>
<th>Xinu as Guest OS</th>
<th>Host &amp; Guest at the Same time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (ms)</td>
<td>0.014</td>
<td>0.033</td>
<td>0.034</td>
<td>0.066</td>
</tr>
</tbody>
</table>
Timer Interrupt Benchmarking

- Sleep Time: Expected vs. Observed

![Graph showing the relationship between expected and observed sleep times. The x-axis represents expected sleep time in seconds, and the y-axis represents average observed sleep time in seconds.]
Summary

• Built VMM for an embedded device
• Mechanisms for providing memory and handling interrupts and privileged instructions
• Minimal processor and memory overhead
• Future work: network device virtualization