Linux on top of OSF MK

François Barbou des Places (barbou@gr.osf.org)
Nick Stephen (stephen@gr.osf.org)
OSF Research Institute - Grenoble

• Objectives:
  • widen the audience for the free OSF micro-kernel
  • provide a complete and free platform to develop new
erswers, applications or kernel extensions
  • provide Apple with a freely redistributable
microkernel-based operating system for the
Apple/PowerMac platform

The Mach3.0 microkernel: Design

• microkernel abstractions
  • task, thread, port, message, memory object
  • resources and services represented as ports
  • interfaces implemented via RPC to an object port

The Mach3.0 microkernel: Benefits

• portability (modular design)
• symmetric multi-processing support
• scalability (from desktop PCs to high-end multi-computers)
• extensibility
• OS-neutrality
• support for OS personalities running as user tasks
  • pageable, preemptable
  • more portable: isolated from hardware
  • simpler: no device drivers, scheduling, VM...
  • external memory managers

OSF MK Improvements

• performance
  • kernel-loaded servers (collocation)
  • thread migration
  • short-circuited RPC

• real-time
  • preemptable kernel
  • real-time RPC, priority inheritance

• distributed system (clusters and multi-computers)
  • DIPC: transparent Distributed IPC
  • XMM: transparent distributed shared memory

• more: MK++, CORDS (xkernel), Fault Tolerance, ...
**OSF MK on the Apple/PowerMac**

- portability: no changes to machine-independent code
- validated with the MPTS (MK performance test suite)
- initial target: PPC601 (PowerMac 7100 and 8100)
- support from Apple Computer, Inc.
- status
  - supports the Linux server
  - console, SCSI disk, serial port, ethernet drivers
  - X-Window running
- next steps
  - support more hardware (drivers, PPC603 and 604, PCI and CHRP PowerMacs)

**Linux Server**

**Why Linux?**

- pleasant and efficient development platform
- evolving very quickly: leverage the effort
- heavily supported by Internet community
- not derived from BSD
- performant dynamic buffer cache

**Drawbacks**

- portability: still quite biased towards PC
- fast moving target

**Linux Server: Design Constraints**

- leverage Linux evolution
  - minimize changes to Linux code
  - binary compatibility
- integration into the Linux tree:
  - new "architectures": osfmach3_i386 and osfmach3_ppc
  - machine-independent code in osfmach3 subdirectory
- build on Linux or the Linux server
- serialized multi-threaded single server
- no emulation library
- objective: performance parity with Linux

**Linux Server: Code Reuse**

<table>
<thead>
<tr>
<th>linux-1.2.13 + server</th>
<th>1117 files</th>
</tr>
</thead>
<tbody>
<tr>
<td>new server files</td>
<td>188</td>
</tr>
<tr>
<td>machine dependent</td>
<td>100</td>
</tr>
<tr>
<td>machine independent</td>
<td>88</td>
</tr>
<tr>
<td>new PPC</td>
<td>20</td>
</tr>
<tr>
<td>modified for server</td>
<td>43</td>
</tr>
<tr>
<td>unmodified</td>
<td>866</td>
</tr>
</tbody>
</table>

~ 50% of those files are in machine-dependent locations, but don’t really need to be ported
### Linux Server: Syscall Redirection

**emulation library**
- emulation library
- unprivileged code
- complexity
- multi-threaded tasks
+ optimized local syscalls

**syscall exception**
- shared memory
- user task
- OS server
+ safety
+ unique and simple interface
- no optimized local syscalls

### Linux Server: Device Access

**Linux**
- `sys_read`  
- `block_read`
- `ll_rw_blockmake_request`

**Linux server**
- `device_reply_thread`
- `block_read_reply`
- `unlock_buffer`

This code is replicated in each block device driver.

This code is common to all block devices and machine independent.

### Linux Server: Memory Management

**Linux**
- `vm areas`
- `page tables`

**Linux Server**
- `vm areas`
- `dummy page tables`

### Linux Server: Signals and Time

**Fake Interrupts**
- signals normally processed on returned from interrupt or system call
- force signal delivery on user-mode-bound processes
  - take control of process by suspending and aborting the Mach user thread *if not in a system call*
  - fall back to the exception handling path

**Jiffies**
- jiffies thread
  - use Mach clock services to sleep 10 milliseconds
  - call `do_timer`
  - simple but inefficient, will be redesigned
**Linux Server: Dynamic Buffer Cache**

- cooperation between VM (microkernel) and Buffer Cache (Linux server)
- emulate the free_area pool of pages in a memory object
- external memory manager: a Linux server thread
- **advisory pageout** to avoid races and select page to free
  - memory_object_discard_request EMM interface
  - try_to_free_page()
  - memory_object_lock_request: let the MK release the page
  - MK falls back to default pager in case of problem
- also avoids double paging (paging buffer cache pages)

**Linux on OSF MK: Status on i*86**

- boots from LILO
- 100% binary compatible
- supports virtual consoles, X-Window, gpm and sdoom!
- current overall performance: 93% of Linux's performance
- disk IO performance better than Linux
- biggest performance penalty on syscall path and copyin/out operations

**Linux on OSF MK: Apple PowerMac**

- runs the available Linux/PPC commands and X-Window

**Further Information**

http://www.mklinux.apple.com

**Availability**

- summer 1996

**BOF Session at the OSF**

- this evening (Sunday Feb 5) from 8:30 to 10:00 PM
- at OSF, 11 Cambridge Center - 100 yards from the Marriott
- demo of Linux on OSF MK on Intel and PowerMac

**Linux on OSF MK: Conclusion**

**Achievements**

- OSF MK can support Linux and with good performance
- a free development platform for OSF MK now exists

**Next Steps**

- complete functionality and improve performance
- integrate Linux device drivers
- take advantage of OSF MK added value
  - multi-threaded tasks, SMP, real-time, clusters

**Snapshot Available for i*86**

http://www.gr.osf.org/mklinux
Linux on OSF MK: Apple PowerMac

- runs the available Linux/PPC commands and X-Window

**Further Information**

http://www.mklinux.apple.com

**Availability**

- summer 1996

**BOF Session at the OSF**

- this evening (Sunday Feb 5) from 8:30 to 10:00 PM
- demo of Linux on OSF MK on Intel and PowerMac