Virtual Frame Buffer for PV Xen

Markus Armbruster, 2006
armbru@redhat.com
Red Hat GmbH
Why and Who

Why? Want a graphical console!
☞ Just like a real machine, just like FV

Whodunnit?

• Anthony Liguori
  Design and initial implementation

• Markus Armbruster & Jeremy Katz
  Flesh out, fix, extend

• The Xen Community
Architecture

Similar to common split driver architecture:

- **Frontend in domU:**
  xenfb and xenkbd kernel modules
- **Backend in dom0:**
  user space VNC server
- **Communicate through**
  shared memory page and event channel
  shared frame buffer
- **Forward compatible** protocol
Shared Page & Event Channel

One page for xenfb and xenkbd each, with:

- Device information
- Input event ring
- Output event ring
- Event notification on ring change
Xenkbd Events

Input:
- Pointer motion
- Button down/up
- Key down/up

Output:
- None defined yet

Forward compatibility:
- Can safely ignore unknown input events
- Generate only requested output events
Xenfb Events

Input:
- Request to signal framebuffer update

Output:
- Signal framebuffer update

Forward compatibility:
- Can safely ignore unknown input events
- Generate only requested output events
Future Work

- Absolute pointer
- Shadow translate mode guests
- International keyboards
- Dynamic resolution
- Multiple frame buffers per guest
- Share backend with FV
- Save/Restore and Migrate
Resources

- **Design notes**
  
  http://wiki.xensource.com/xenwiki/VirtualFramebuffer
  
  (somewhat dated)

- **Discussion**
  
  http://lists.xensource.com/xen-devel