Megasas megasas

An efficient SCSI HBA emulation for KVM/Qemu

Hannes Reinecke
SUSE Labs
hare@suse.de

Novell®

Qemu device emulation goals

- Emulate existing hardware
- Emulate common drivers
- Device multiplication
  - Network drivers
  - Storage drivers
- Supports various platforms

© Novell Inc. All rights reserved

KVM device emulation goals

- Efficient emulation on Linux
- Paravirtualisation: virtio
- One device only, but efficient
- Main focus on x86 platform
- (Alex Graf it ported to PPC and S/390 :-)

© Novell Inc. All rights reserved

SCSI device emulation

- Two types: scsi-disk, scsi-generic
- Scsi-disk emulates disk access
  - Translates every command
  - Using bounce-buffers to transfer data
  - Using bdrv_XXX functions to access backing device
- Scsi-generic for everything else
  - No command translation
  - SG_IO to pass commands to backing device

© Novell Inc. All rights reserved

SCSI HBA emulation

- Emulates NCR/LSI 53c8xx
- SCSI parallel HBA
- All devices are emulated as SCSI parallel devices
- Emulates on HBA level:
  - Scripts engine
  - SCSI parallel protocol quirks
- Prone to breakage

© Novell Inc. All rights reserved

The search for an HBA

HBA emulation goals

- Should be a SCSI driver
  - I want to use all the nifty sg_XX tools
- Existing driver / hardware
  - Emulate only one side, use existing driver for the guest
- Simple driver / HBA interface
  - Efficient implementation, less coding
- Protocol independent interface
  - All type of devices should be passed/emulated
  - Full protocol handling cumbersome

© Novell Inc. All rights reserved

Protocol independent interface

- Excludes protocol-specific drivers
  - SAS/FC HBA
- Possible HBA types:
  - LSI MPT driver
  - I2O driver
  - RAID driver

© Novell Inc. All rights reserved

Simple HBA interface

- LSI MPT interface rather complex, so not doing it
- I2O … well
- RAID driver:
  - Aacraid: Can't stand Adaptec …
  - Hpsa: new development, not readily available
  - Megaraid: Old interface horrible
  - 3Ware: SATA mainly, not really multiprotocol
- Megaraid_SAS has a pretty simple interface
  - Protocol independent
  - Simple structure

© Novell Inc. All rights reserved

Updating KVM/Qemu

Qemu/KVM SCSI implementation

- Very old style:
  - Requests are opaque, access via 'tag'
  - Sequential processing
  - Call ->send_command() to evaluate CDB
  - Receive completion on tag
  - Call ->read_data() / ->write_data() to fill buffer
  - Receive completion
  - Access the buffer via ->get_buf()
  - Finish command
  - Sense data has to be requested separately

© Novell Inc. All rights reserved

Megasas cmd handling

- Receive command
  - Two type of commands: SCSI Passthrough and direct I/O (Read/Write) commands
  - Data is passed via SGLs
- Handle command
  - Process command
  - Process I/O
  - Handle exceptions
- Send back status & sense code

© Novell Inc. All rights reserved

First try: Rewrite SCSI stack

- Expose the SCSI request to SCSI driver
- Allow driver to pass scatter-gather list to the block driver
- Allow the driver to access sense data directly
- Update existing driver to the new interface
- Quite intrusive ...

© Novell Inc. All rights reserved

Scatter-gather list passing

Scatter-gather list handling

- Scatter-gather list is received by the HBA
- Convert and pass down to backing device
- File-backed:
  - Using aio to write data / sgl to disk
- Device-backed:
  - Using SG_IO to write data/sgl to device
- SG list might be misaligned
- No definite indication for this
  - A priori checking required

© Novell Inc. All rights reserved

Misaligned sg lists

- Request queues on host and guest might have different parameter
- Passing misaligned sg list results in I/O error
- Ideally: queue parameter are aligned
  - No guarantee
- Pass host queue parameter up to the guest
  - megaraid_sas reads parameter off the hba, so we can be passing the information

© Novell Inc. All rights reserved

HBA – disk mapping

- Request queue reflects the HBA settings
- Connecting devices from different HBAs to megasas emulation will not work
- Disallow, have a 1:1 (host)disk / (guest) HBA mapping initially

© Novell Inc. All rights reserved

Developing the driver

Linux driver as template

- Quite complete interface description in megaraid_sas.h
- Implement target side to match linux driver expectations
- Simple frame interface:
  - Driver allocates ring buffer and passes location of head and tail pointer to HBA
  - Driver writes frames, updates head pointer, and notifies HBA via mmio
  - HBA processes frames, updates tail pointer, and notifies driver via doorbell register & interrupt

© Novell Inc. All rights reserved

Windows booting

- HBA detection failed; frame analysis didn't reveal anything
- Analyse qemu traces: driver checks for the number of available frames and aborts??
- Disassemble windows driver:
  - Driver reads the number of frames from the driver
  - If this number is <u>not</u> 1000 → abort
- (How stupid can one get …)
- Fix frame number to 1000
- Windows boots :-)

© Novell Inc. All rights reserved

Management interface

- Most of the protocol given to management functions
- Stand-alone management CLI 'MegaCLI'
- Implement rudimentary functions to make CLI work
- ¼ of calls undocumented
- 'Educated guesswork' to implement missing functions
- CLI now runs without errors
- (without functionality, too)

© Novell Inc. All rights reserved

Initial results

I/O performance

- Bonnie run on LSI:

```
Writing with putc()...         done:     33 MB/s  51.3 %CPU
Rewriting...                   done:     27 MB/s   3.4 %CPU
Writing intelligently...       done:     23 MB/s   4.1 %CPU
Reading with getc()...         done:     30 MB/s  45.8 %CPU
Reading intelligently...       done:   1063 MB/s  62.3 %CPU
```

- Bonnie run on Megasas:

```
Writing with putc()...         done:     28 MB/s  45.1 %CPU
Rewriting...                   done:     27 MB/s   3.2 %CPU
Writing intelligently...       done:     34 MB/s   5.5 %CPU
Reading with getc()...         done:     51 MB/s  67.0 %CPU
Reading intelligently...       done:   1220 MB/s  81.0 %CPU
```

© Novell Inc. All rights reserved

I/O performance (contd.)

- Bonnie run on virtio:

```
• Writing with putc()...          done:     54 MB/s  80.6 %CPU
• Rewriting...                    done:     47 MB/s   5.3 %CPU
• Writing intelligently...        done:     52 MB/s   8.8 %CPU
• Reading with getc()...          done:     58 MB/s  74.8 %CPU
• Reading intelligently...        done:   1318 MB/s  66.9 %CPU
```

- Bonnie run on Megasas:

```
• Writing with putc()...          done:     28 MB/s  45.1 %CPU
• Rewriting...                    done:     27 MB/s   3.2 %CPU
• Writing intelligently...        done:     34 MB/s   5.5 %CPU
• Reading with getc()...          done:     51 MB/s  67.0 %CPU
• Reading intelligently...        done:   1220 MB/s  81.0 %CPU
•
```

© Novell Inc. All rights reserved

Smartd weirdness

- Log during boot:

```
Starting SSH daemon                                                  done
Starting cupsd                                                       done
Starting irqbalance                                                  unused
Starting Name Service Cache Daemon                                   done
Starting mail service (Postfix)                                      done
Starting CRON daemon                                                 done
Starting smartd megasas: xfer length mismatch, frame 512 cdb 256
                                                                     unused
Master Resource Control: runlevel 3 has been                         reached
Skipped services in runlevel 3:            nfs smbfs irq_balancer smartd
```

© Novell Inc. All rights reserved

LSI legalese

- Got in touch with LSI to get legal clearance
- Legal review at LSI
- Rejected in the sense that LSI will not endorse any HBA virtualisation
- “This would allow you to run LSI driver on a non-LSI hardware” … well, yes … that was the whole point...
- Discussions ongoing.

© Novell Inc. All rights reserved

Multipath testing

- 'Real' SCSI HBA, no problems with multipath setup
- Using LIO-target to emulate disks
- Testing and implementing advanced multipath scenarios:
  - Dynamic ALUA
  - Multipath with Referrals

© Novell Inc. All rights reserved

Upstreaming it

- Fighting with Qemu community
- Main points of contention:
  - SG list mismatch
  - Limited I/O size
- Original LSI HBA emulation transfers data in chunks of 128k, thus allowing for (in theory) unlimited data size
- SGL lists are limited by the hardware / HBA emulation, thus the total I/O size is limited, too.
- But the OS will only allow I/O which matches to hardware limits, hence quite a pointless argument

© Novell Inc. All rights reserved

Next try: Updating existing stack

- Rearrange patchset to send 'real' enhancements first
  - Sense code handling
  - INQUIRY fixes
  - Use SCSIRequest directly instead of referencing it via 'tag'
  - Add possibility to pass in SG Lists
  - Rewrite megasas emulation to match the updated interface
- Leaving existing drivers intact, thus objections should be resolved
- Sending it upstream shortly

© Novell Inc. All rights reserved

Novell®

Unpublished Work of Novell, Inc. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary, and trade secret information of Novell, Inc. Access to this work is restricted to Novell employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. Novell, Inc. makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for Novell products remains at the sole discretion of Novell. Further, Novell, Inc. reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

Color Palette

BLUE RGB **0 166 238**	RED RGB **224 0 0**	ORANGE RGB **230 120 20**
TEAL **RGB** **50 118 109**	GREEN RGB **98 158 31**	YELLOW **RGB** **255 221 0**
DK GRAY RGB **60 60 65**	MD GRAY RGB **90 90 100**	LT GRAY RGB **204 204 205**

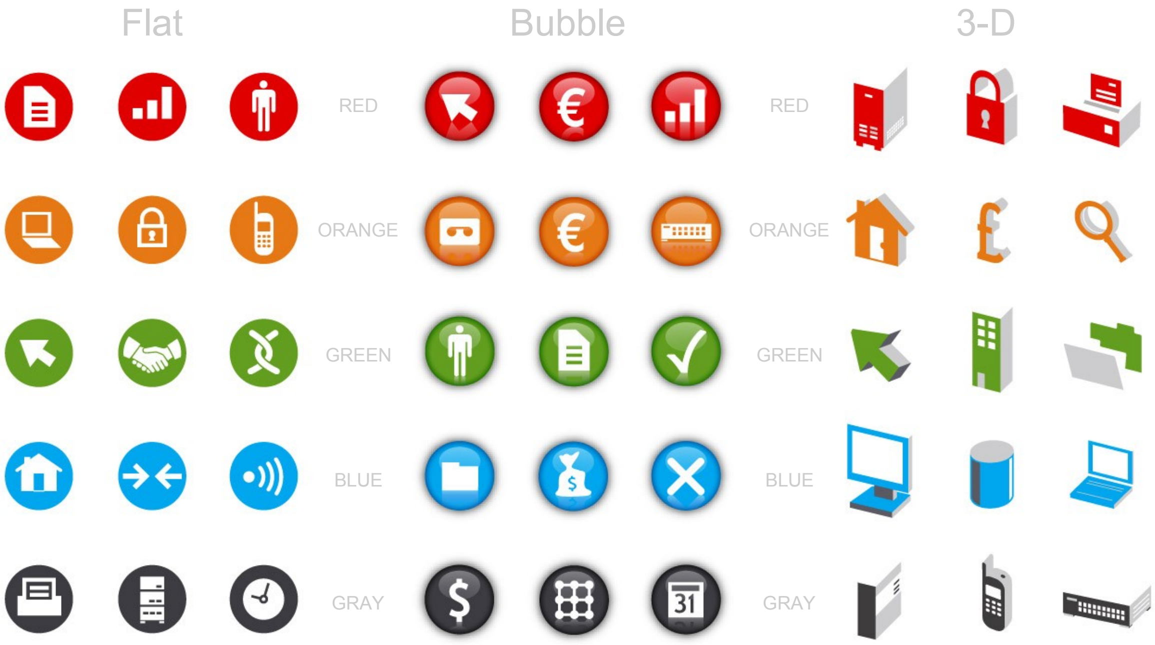
Note:
The gray dotted-line box represents the margins or "working area" into which all text and most graphics and diagrams should conform.

How to Add Novell Colors to Your OpenOffice Color Palette:

1. Go to the "**Tools**" menu
2. Select "**Options**"
3. Expand "**OpenOffice.org**"
4. Select "**Colors**"
5. Delete existing colors (one-by-one)
6. Add Novell Colors by giving them a name and entering RGB values
7. Click "**OK**"

© Novell Inc. All rights reserved

Graphics & Typeface



Download Icon Library at: http://innerweb.novell.com/brandguide

How to Add Novell Icons to OpenOffice Gallery:

1. Go to the “**Tools**” menu
2. Select “**Gallery**”
3. In the Gallery window select “**New Theme...**”
4. With the “**General**” tab active name your new theme (ie.Red flat)
5. Select the “**Files**” tab.
6. Select “**Find Files...**”
7. Find the downloaded folder containing the icons named and click “**Select**”
8. Select “**Add All**” and then “**OK**”
9. Repeat for all icon groups

Note:

Icons/Lines: This presentation refresh simplifies the current template and pushes focus on the content being presented. The icon library will continue to be utilized, but a refresh will be noticeable with the addition of the “Bubble” set of icons, and a subtle color shift. These icons are created to provide a professional, consistent look. When these icons are used sparingly, and in direct relation to the content on the slides, our presentations will communicate and work more effectively.

Typeface: Arial has been selected as the new typeface for all Novell communications. The following were considered.

1. Our typeface needs to be designed to carry information quickly to the reader.

2. It needs to be usable for Novell employees in company correspondence and presentations, as well as for outside vendors for marketing and promotion.

3. It needs to easily function on the Linux, Windows and Macintosh platforms.

4. And finally, Arial was created for these exact purposes.

© Novell Inc. All rights reserved