- **Name:** Akshay Jaggi
- **Email:** akshay1994.leo@gmail.com
- **Phone:** +91-888-678-7185
- **IRC:** akshay1994 (freenode.net)

- **Availability:**
    - 6 hours a day, 5 days a week (~120 hours a month)
    - Classes/Exams end on April 30, 2016 so they won't interfere with the GSoC Coding period
- **Biography:** On Page 5
- **Possible Mentor:** Roger Pau Monné <royger@freebsd.org>

Project Proposal

Project Title

Grant-Table User-Space Device

Project Description

Abstract

To ensure efficiency there is a need to share page frames between Xen domains, which can then be used for purposes like bulk data transfer between domains, inter-domain communication, direct DMA of data from hardware devices into the target domain by a driver domain, etc.
Dom0, being the privileged domain, has access to all the pages of all the domains, but the same is not true for domU unprivileged domains. Grant-Tables come to the rescue! Grant tables provide a mechanism to share and transfer pages between domains.
Each domain has its own grant table, a data structure that is shared with Xen, and which contains information about what kind of permissions other domains have on this domain's pages. Two kinds of operations are supported: granting a page to another domain, and mapping the grant from another domain.
Entries in the grant table are indexed by an integer - grant reference, which is generated by the granting domain and used by the receiving domain to map the shared page.
Although the kernel can utilise grant tables, there is no way for user-space applications to use them in FreeBSD. Hence, the goal of creating a grant table userspace device, which can be utilised by userspace applications to grant local memory and map grants from other domains.

Goals

Write a device exposing the grant table to the user-space.

Detailed Goals

Write a character device which will do all the required tasks. We will try to expose an API which is very similar to the linux user-space grant-table device, so that we can take advantage of existing handlers as much as possible.
The API exported by the device is comprised of two components:

- **Grant Allocator** - Manages creation/deletion of grants from local memory to be shared with other domains.
    - **Allocate Grant** - Allocates a new page of kernel memory and creates a new grant reference for this page using the Xen Control Library (libxc).

    - **Deallocate Grant** - Deallocate the grant reference. If the page is not mapped by anyone anymore, free it.
    - **Set Unmap Notification**
- **Grant Mapper** - Manages grants from other domains.
    - **Map Grants** - Prepare grants from other domains to allow them to be mapped to user-space.
    - **Unmap Grants** - Unmap grants previously mapped.
    - **Virtual Address to Offset** - return offset (used as an index to refer to the appropriate grants (explained later)) using virtual address where the start of this grants range was mapped. Used by programs which do not keep track of offset.
    - **Set Max Grants**
    - **Set Unmap Notification**

Both Allocate Grants and Map Grants return an **offset** which can be passed along with mmap() by the user-space application. The appropriate pages (allocated, or the ones received as a grant) are then mapped into the user-space application's memory.
We need to write a device that exposes this API; an interface to talk to the Xen control library to manage the grants; and a pager to manage allocation and deallocation of pages.

Grant tables are used extensively by QEMU in order to implement para-virtualized backend drivers, for instance, qdisk[7], in user-space. This project will allow us to utilise these backend drivers in a FreeBSD dom0 and thus also provide us a way to benchmark and test our implementation.

Deliverables

- Grant Table User Space Device
- Working QEMU qdisk backend for FreeBSD Dom0

Test Plan

The best way to test the implementation would be to use existing user-space applications which utilise grant tables.
XEN can utilise the para-virtualized qemu-xen backend, qdisk, as a disk backend for DomU guests. This is only possible if the Dom0 host supports user-space grant table access, as qdisk is implemented in userspace.
With this project a FreeBSD Dom0 would be able to utilise these backends for guests, and it also gives us a robust way to test out implementation.
Possible tests include:

- Filesystem consistency checks after multiple read/write operations on a guest with different media sizes[1], using a FreeBSD Dom0 and qdisk as disk backend.
- Disk benchmarks (read/write speed, etc.) on guests with different media sizes[1] and comparison amongst following configurations:
    - FreeBSD Dom0, qdisk

    - FreeBSD Dom0, tap
    - Linux Dom0, qdisk
    - Linux Dom0, tap

Project Schedule

Community Bonding Period (April 22 - May 22)

I would like to spend this period learning more about FreeBSD and it's people. Learning more about the different components (especially Xen) and talking to people who worked on those components. This will help me get more involved with FreeBSD as a whole.
Apart from this, this period would also be utilised to come up with a detailed plan with respect to the project execution, so that we do not end up wasting time during Coding Period. Reading similar implementations (privcmd), and resolving any design issues that need to be sorted out, etc. should also be done in this period. Also, I'll utilise this period to go through the style guidelines, etc.

Coding Period (May 23 - August 23)

Pre Midterm (May 23 - June 21)

My goal for midterms is to submit the grant table device for testing. This will pave the way for post-midterm work, and at the same time keep us on track.
Work involves writing the device, pager and the libxc interface, with basic testing so as to make it capable enough for it to be tested by the community.

Midterm Evaluation (June 21 - June 27)

Post Midterm (June 28 - August 15)

Work involves making QEMU backend qdisk works with the newly designed device. Begin benchmarking and testing the implementation. Lookout for bugs, and bug-fixes.
Post benchmarking, prepare and submit the device for release, along with the benchmarking results.

Pencils Down (August 16 - August 23)

Write project report and documentation. Finish up any remaining work, tidy up things, and submit code for final evaluation.

Biography

I am Akshay Jaggi, a fourth year undergraduate student, studying Computer Science and Engineering, at International Institute of Information Technology, Hyderabad, India.

Experience

I've worked on Operating Systems, Compilers, Filesystems. Below are some of the projects I have worked on. A more complete list can be found on my resume[2].

- I was a **Google Summer of Code** student for **Haiku, Inc.** in **2014**. My project was porting libusb (a user-space USB library) to Haiku. It involved writing an interface to link up the libusb backend with usb_raw, a character device which exposed Haiku's USB API to user-space. Bugs in the kernel USB driver (specifically short-packet transfer bugs in UHCI and EHCI) caused attempts at porting to fail. We managed to fix those bugs and also managed to do some other fixes to the XHCI host controller. Link to Pencils Down Blogpost [3]. Link to Proposal [4].
- I was a summer intern with the Search Infrastructure team at Google, Inc. (Mountain View). My work involved migrating late RPCs to a simpler parallelisation scheme.
- Compiler for Decaf (stripped down version of C), which used Flex and Bison as the frontend-parser and llvm as the backend for code generation.

FreeBSD Experience

I've spent the last few weeks learning a lot about a lot of new things. I was interested in the "Ext2fs: Implement new features" project, and was in touch with Pedro Giffuni (pfg@). I read up quite a bit about ext2/3/4 and also looked at the current state of ext2fs in FreeBSD. A lot of time here was spent reading code in [5], and making sense of how it works. In the meantime, I also fired up a VM and played around with FreeBSD, setting up a build environment, compiling and installing the kernel, and the world, etc.
Then I worked for the USB and SCSI Front End Drivers project, got myself acquainted with Xen, tried out certain things, learnt up about para-virtualized drivers, skimmed through the code for netfront/netback and blkfront/blkback[6], but sadly PVUSB is still not supported by xl, and thus had to give up on that project. Last few days, I read up on grant tables and how they work, the API linux exports for the grant table user space device, etc.

References

[1]http://lists.xen.org/archives/html/xen-devel/2012-02/msg00918.html
[2]http://web.iiit.ac.in/~akshay.jaggi/resume.pdf
[3]https://www.haiku-os.org/blog/akshay1994/2014-08-18_libusb_port_pencils_down
[4]https://docs.google.com/document/d/1C837X1Y5-VDASYCQaSoTBP47heFUdYtZ3TdMoOhuXB4/edit?usp=sharing
[5]http://bxr.su/FreeBSD/sys/fs/ext2fs/
[6]http://bxr.su/FreeBSD/sys/dev/xen/
[7]https://code.grnet.gr/projects/qemu/repository/revisions/62d23efac8905a46277f666c909e826f91c12aa1