

OBF

Meeting Minutes December 2, 2010

Attendees: Heather Kelly, Jana and Gregg Thayer, Tracy Usher, Tony Waite

Tony provided the current state of things. In August/September much work was done. Started by looking over the list of unresolved references in offline software provided by Tracy, and figured out what FSW packages are involved in satisfying them. Joanne also helped out by providing a script showing which packages we grab to create our OBF external library. Tony pointed out that we have been including a whole bunch of libraries that we do not need, such as task communications. Tony and company worked to find a "cleave" point to cut down on the packages offline needs. This was fairly successful and where we previously used 19 FSW packages, we are now down to 10. A sandbox has been set up. The code building has been updated to support rhel++ platforms and then they plowed through the 10 packages. Most of the changes were trivial. CMT requirements files were updated to handle new targets, and gcc 4.1 requires the use of < > instead of "" in the include statements. A test build still needs to be done.

Tony reports that they have a way to back propagate to prior builds, back as far as 1-1-3 should be possible. This suits offline just fine, as we are still currently using B1-1-3.

Further progress has been held up by other pressing FSW upgrades and EXO work.

What is left to do? Haven't yet gone through the data packets, get things into the LAT testbed and make sure things haven't broken, JJ would like to retrofit some of the test programs for rhel4 & 5 to make sure they still function. Tony suggests that we need a month of his, JJ and Owen's time. The remaining work is "grunt work", the real hurdles have been overcome.

Jana asked when Tony expects we could get a month of his, JJ and Owen's time, and we suspect around February/March. We'd also like to get some of Tracy's attention at that time to help test things out on the offline side. Tracy pointed out that there is a Pisa workshop in mid-February, but other than that, that timeline sounds feasible.

The meeting ended with an internal goal of delivering an updated FSW to offline by the end of February.

The current progress is very promising and should allow offline to handle a potential migration of GlastRelease to RHEL5 by the end of 2011.

Meeting Minutes July 16, 2010

Attendees: Heather Kelly, Kim Lo, JJ Russell, Jana and Gregg Thayer, Tracy Usher, Tony Waite

Jana asks what our drop dead date is for OBF support of gcc4/64 bit builds. In consultation with Richard, we absolutely need an updated OBF by the end of life for RHEL4 which is Feb, 2012. That being the case, we would need OBF by Nov, 2011. Heather would like to back that up a little bit and say summer, 2011 so offline has time to test things out and make sure all of our other code is working on RHEL5.

From Tony:

The only good news is that I believe FSW's CMX system is now capable of generating all elements of the two-by-two matrix (RHEL4/RHEL5 64/32). I have not attempted a "micro-trace" of Obf through the flight software code base.

From Gregg:

I've been working on and off at trying to refactor the OBF code so that that list gets shorter. Some small progress has been made, but not a heck of a lot.

Tony provides one example of the type of required changes: apparently gcc4 makes a distinction between <> and "" for include files. This will require modifying the include directives throughout the FSW code. Jana notes that FSW uploads are limited to 1.3 MB and if every file is touched, such an upload may exceed this limit. That would require multiple uploads for one change, something we haven't had to do previously.

Once Tony has an initial list of required constituents, he can pass it on to JJ, Gregg, Kim for further investigation.

We'll check in with Tony in the Oct/Dec time frame to see if some time has freed up to allow him to move forward with creating a list of constituents required for OBF. If at that time, progress does not seem possible, we'll come up with another plan.

In speaking to Richard afterwards, he wonders if it is truly necessary to upload any FSW changes that come about for OBF support in GR?

Meeting Minutes March 24, 2010

Attendees: Heather Kelly, Kim Lo, JJ Russell, Jana and Gregg Thayer, Tracy Usher, Tony Waite

Tony reviewed where he stood last October, see extreme bottom of this Confluence page. JJ noted that he intended OBF to be independent of some of the lower level packages like PBS. If that dependence is truly there, he may be able to clean it up. Tony reminded us that CMX desires to build everything from the bottom up. We have not previously attempted to excise bits of FSW and build subsets.

FSW options:

1. Actually move all of FSW to build on RHEL5-64.
2. Try to excise only what OBF requires.
3. Something else?

and does this version propagate onto the satellite?

The Offline alternative is to keep OBS/FSW on RHEL4-32 and virtualize that step and pull it out of GR.

Tony on-the-fly-brainstorming thinks it may be possible to get CMT to build sub-trees, by using the special tags associated with RHEL5-only. JJ and Gregg will review what portions of FSW are absolutely necessary for OBF. Then Gregg and Kim can attempt a "path-finding effort" (where they basically compile and see what errors pop up and fix them) with Tony's assistance as necessary.

Jana requested that Tracy provide a list of FSW libraries that offline currently depends on (where that list was obtained by building and adding libraries in until all missing references were resolved):

```
"CDM/V0-2-4"
"EDS_DB/V0-0-2"
"EFC_DB/V2-0-0"
"GEO_DB/V2-0-0"
"GGF_DB/V2-0-0"
"GFC_DB/V3-0-0"
"XFC_DB/V3-1-1"
"CPU_DB/V0-4-2"
"LEM_DB/V0-1-5"
"CGB_DB/V0-1-0"
"COP_DB/V0-0-1"
"CPP_DB/V0-1-1"
"COG_DB/V0-0-1"
"CPG_DB/V0-1-0"
"EFC/V4-3-0"
"XFC/V0-1-2"
"EDS/V2-9-1"
"PBI/V0-1-0"
"LSE/V1-3-6"
"FBS/V0-2-3"
"CMX/V2-12-2"
"CAB/V1-0-0"
"MDB/V0-0-1"
"ZLIB/V2-4-0"
"PBS/V2-10-15"
"EMP/V1-3-5"
"IMM/V0-3-2"
"MSG/V3-1-1"
"ITC/V3-9-0"
"CCSDS/V3-5-2"
"LCBD/V1-4-3"
"LCBT/V1-6-1"
"THS/V1-6-1"
"LEM/V4-7-1"
```

Options	Consequences
Convince FSW to exert effort to build on 64 bit machines	Work cannot start until February and it is unclear how much time will be necessary or how many FSW packages will require updates.
Build FSW (or portions of it) ourselves using CMT (actually do we want to use FSWs CMX build system?)	Can we afford the resources to do that? Building is one thing - working through the pile of issues for 64 bit or gcc 4 is another and would still likely require a lot of interaction with the FSW team.
Ask FSW to at least support gcc 4 but stay with 32 bit (RHEL5-32)	May be as much effort as moving to 64 bit and gcc 4 Work cannot start until February Many concerns about distributing such 32 bit builds to the collaboration where those with an interest in gcc4 would likely have 32 bit machines. We cannot mix and match 32 bit and 64 bit libs.
Stay with what we have: FSW built on RHEL4-32	We would have to upgrade to this newer FSW build to take advantage of it, hopefully not too labor intensive as the OBF portion is supposedly unchanged. Either we freeze all of GR on RHEL4-32 - we can run a RHEL4-32 build on RHEL5 - but requires that compatibility libs be available on the machines this may cause problems for non-SLAC machines if they are not set up. OR we extract the OBF portions of GR to allow the rest of GR to move ahead. OBF becomes its own separate step in the processing and to support more modern OSES we would virtualize the OBF step. Virtualizing the OBF step will also involve some work - who will do that?
Stop using OBF and write our own filter code as we did in the old days	Risk of not fully duplicating the existing OBF code. Future changes to OBF in FSW would then have to be reflected in our version (it is unclear how much more modification to OBF there really will be though).

Minutes from Meeting Friday October 23, 2009

Tony Waite, Jana and Gregg Thayer, Heather Kelly

Currently offline is using FSW B1-1-3. Tony has no way to estimate how much effort it is to go to 64 bit without actually going through the exercise. Tony and JJ are unavailable. Jana and Gregg are unavailable until at least February. Tony was wondering about having offline stick with 32 bit on RHEL5 & RHEL4 as it is unclear what benefit we get from 64 bit anyhow. That way we could pursue a gcc4.1 build - though to scope out the amount of work required would be similar to just forging ahead on RHEL5-64. I'm not sure how content offline would be with 32 bit builds, but we could consider it. Tony wanted to discuss with Tracy how we build and use OBF. Heather mentioned that we use the FSW shareables on Linux, while on Windows Tracy is building from source.

Note from Tracy Oct. 23, 2009

basically, in order to use the various filters we have no choice but to use their higher level interface to it, since the filters expect to see the data in a particular format and we need their higher level stuff to handle the setup of that format. Without a HUGE amount of work on our part, we aren't going to change that. Anyway, when we went to the model of using the FSW libraries I had discussions with JJ and he agreed that was a good way to go, even though it did mean bringing in a lot of extra FSW dependent libraries. The goal is to use THE filter code that is running on the LAT. We could back off and have our own filters but then you run the risk that you are not really duplicating their operation. In addition, any future changes to the filters will require parallel changes to our stuff so we'll need to maintain an "expert" to do that. Since my boss tells me I am supposed to be doing other things now, I don't know who that person is. So, in short, if the collaboration requires that the filters are part of Gleam then the FSW group needs to get their hands dirty one way or the other.

An option is to go back to building the stripped down windows version of OBF on linux. Since I do this with a cmt requirements file it's probably not that difficult to do. I guess I don't know anything about 64 bit machines and what has to be done with cmt to get a requirements file to do a build this way, but we could look at it.

Email from Tony Waite after first look at FSW and RHEL5-64

I've looked at 32/64 RHEL4/RHEL5 support being properly built into CMX. It's a nightmare. Several things conspire to just make it a royal pain:

- 1) The AFS system does not distinguish between RHEL4 and RHEL5 in its 'fs sys' response ... bit of a shame because CMT uses that to generate the CMT tag!
- 2) Other methods of determining the host/OS/bit-width are completely flaky. The only way I've found to develop this is to inspect the linux kernel version ... and that would only work on redhat-linux machines!
- 3) Even if I could distinguish RHEL4/5, what does that really mean? Unless Rehat pull another backward incompatibility trick (libstdc++ changed significantly from RHEL3 to RHEL4 ... hence our recent problems), then anything built with RHEL4 should be fine under RHEL5. The only real distinction I have found is a significant bump in the gcc compiler version (3.4.6 to 4.1.2). That might sound bad, but FSW is all written in C, which is far less vulnerable to compiler upgrades.

Nevertheless, I'm gradually retrofitting CMX where I can. Which brings me to the next problem. Along with the code itself, CMX versions the tools it uses. CMX is itself a CMX package. Bootstrap problem. When (for instance) FSW build B1-1-3 was declared, it froze out the version of CMX associated with it. If I have to modify CMX itself to produce 32/64 RHEL4/RHEL5 builds, then technically, I'm breaking the build (this did not apply when rebuilding on rhel4-32 ... no modifications were needed ... just needed to log into a machine with the right architecture/OS).

So I'm exploring another method for producing all the variations. It's far too hand-held for my taste and will have poor traceability, but it's probably all ground software needs. The basic principle is to bugged a minimum number of CMX files (so far that's just the requirements file and the cmx_link.pl script). The modifications amount to redefining what our "linux-gcc" tag means. If and when the build completes, rename all directories names "linux-gcc" to an architecture specific name (e.g. "rhel5-32"), then repeat for the next architecture. I'm practicing on the B1-1-3 build. If I can get this going, I'll hand off the procedure to Kim, so that he can do the same for other FSW builds needed by ground.

Next message:

FSW B1-1-3 on rhel5-32: (major characteristic ... compiler is gcc 4.1.2)

- 1) New compiler generates some new global symbols. That's a no-no in a CDM style shareable (a shareable that exposes global symbols cannot be guaranteed to be unloadable). To make progress, I assumed that the symbols involved were not linked against and added them to the "forgive" list.
- 2) New compiler is very fussy about placement of "laddr" objects. Results in a fatal error in package PBS (fixed packet allocator). Didn't try to fix.

FSW B1-1-3 on rhel4-64: (major characteristic ... 64 bit width ... d'oh)

- 1) Has new/different compiler macros to express architecture. Package PBI does not understand/recognise them, meaning that the "Endianness.h" header file turns round and complains that it can't determine machine endian-ness. Didn't try to fix.