On Tue, 2014-11-25 at 00:46 +0000, Juvva, Kanaka D wrote:

> Hi Matt,

>

> I have looked at how to submit patches to git.kernel.org and

> kernel.org. Please let me know this is pretty much what we do.

>

> Kernel.org:

>

> <https://www.kernel.org/doc/Documentation/SubmittingPatches>

>

Right, this is a good overview.

> Using git:

>

> <http://linux.koolsolutions.com/2011/02/26/howto-create-and-submit-your>

> -first-linux-kernel-patch/

Yeah this looks like a good start.

In more concrete terms, what I'd suggest you do is the following,

1. Add the stable kernel git repo to your current local git repo 2. Fetch the v4 patches from the mailing list 3. Use git am to apply each of the 11 patches and fix conflicts 4. Generate new patches

Step 1: Get the stable git repository

Say you've cloned Linus' git repository from kernel.org with the following command,

git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git

but you now want the stable tree too (which is a separate git repository). The stable repository url is,

git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git

and the specific version that Pandi and his team are wanting the patches applied against is v3.17.4 (we know this is a stable tag because they take the form <Linus kernel release version>.<some stable minor

version>, i.e. v3.17.4 is the v3.17 kernel that Linus released plus 4

stable releases of changes.

Anyway, to add an additional repository to the list in your existing Linus repository checkout do,

$ cd linux

$ git remote add linux-stable git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git

$ git fetch linux-stable

Step 2: Fetch the patches from the mailing list

Using whatever mail client you have save each of the 11 patches from the mailing list to a local directory on your machine. Perhaps /tmp. (Note you do not need the zero'th patch which is just a cover letter for the patch series).

Step 3: Apply the v4 patches

Now you need to create a new local git branch on which to apply the patches. You can do this with,

$ git checkout -b cqm-patches-for-pandi v3.17.4

Note that I've specified a git tag at the end (v3.17.4) which is where I want to point the HEAD of the 'cqm-patches-for-pandi' branch at.

Say you've saved each of the 11 patches from the mailing list into the /tmp directory. Applying the patches goes like this,

$ git am /tmp/\*.patch

If I do this, I see the following,

Parse event per-package info files

Applying: perf tools: Parse event per-package info files

error: patch failed: tools/perf/util/evsel.h:91

error: tools/perf/util/evsel.h: patch does not apply

error: patch failed: tools/perf/util/parse-events.c:681

error: tools/perf/util/parse-events.c: patch does not apply

error: patch failed: tools/perf/util/pmu.c:209

error: tools/perf/util/pmu.c: patch does not apply

error: patch failed: tools/perf/util/pmu.h:29

error: tools/perf/util/pmu.h: patch does not apply Patch failed at 0001 perf tools: Parse event per-package info files The copy of the patch that failed is found in:

/home/matt/src/kernels/efi/.git/rebase-apply/patch

When you have resolved this problem, run "git am --continue".

If you prefer to skip this patch, run "git am --skip" instead.

To restore the original branch and stop patching, run "git am --abort".

git failed to cleanly apply the very first patch; it generated some conflicts. So what you need to do is fixup the conflicts and commit those fixups.

Now, because things didn't apply cleanly git am made no changes whatsoever to your working tree,

$ git status

On branch cqm-patches-for-pandi

You are in the middle of an am session.

(fix conflicts and then run "git am --continue")

(use "git am --skip" to skip this patch)

(use "git am --abort" to restore the original branch)

nothing to commit, working directory clean

Handily, git stashed the patch in .git/rebase-apply/patch, so you use the standard patch command to apply it and see where the conflicts are,

$ patch -p1 < .git/rebase-apply/patch

patching file tools/perf/builtin-stat.c

Hunk #1 succeeded at 388 (offset -1 lines).

Hunk #2 succeeded at 525 (offset -1 lines).

Hunk #3 succeeded at 1209 (offset -2 lines).

patching file tools/perf/util/evsel.c

Hunk #1 succeeded at 866 (offset -49 lines).

Hunk #2 succeeded at 884 (offset -52 lines).

patching file tools/perf/util/evsel.h

Hunk #1 FAILED at 91.

Hunk #2 succeeded at 251 (offset -6 lines).

Hunk #3 succeeded at 264 (offset -6 lines).

Hunk #4 succeeded at 277 with fuzz 1 (offset -6 lines).

1 out of 4 hunks FAILED -- saving rejects to file tools/perf/util/evsel.h.rej

patching file tools/perf/util/parse-events.c

Hunk #1 FAILED at 681.

1 out of 1 hunk FAILED -- saving rejects to file tools/perf/util/parse-events.c.rej

patching file tools/perf/util/pmu.c

Hunk #1 succeeded at 171 (offset 8 lines).

Hunk #2 succeeded at 207 (offset 8 lines).

Hunk #3 succeeded at 221 (offset 8 lines).

Hunk #4 FAILED at 229.

Hunk #5 succeeded at 660 (offset -9 lines).

Hunk #6 succeeded at 681 (offset -9 lines).

1 out of 6 hunks FAILED -- saving rejects to file tools/perf/util/pmu.c.rej

patching file tools/perf/util/pmu.h

Hunk #1 FAILED at 29.

Hunk #2 FAILED at 39.

2 out of 2 hunks FAILED -- saving rejects to file tools/perf/util/pmu.h.rej

Hunks that failed to apply cleanly are in corresponding .rej (reject) files. Taking the first reject as an example, the reject looks like this,

-- tools/perf/util/evsel.h

+++ tools/perf/util/evsel.h

@@ -91,6 +91,7 @@

bool immediate;

bool system\_wide;

bool tracking;

+ bool per\_pkg;

/\* parse modifier helper \*/

int exclude\_GH;

int nr\_members;

Looking at the working copy of tools/perf/util/evsel.h we can see why it didn't apply cleanly,

bool immediate;

/\* parse modifier helper \*/

int exclude\_GH;

int nr\_members;

i.e. that the 'system\_wide' and 'tracking' members don't exist in this struct in v3.17.4. The way to resolve this is to add the 'per\_pkg'

member by hand to tools/perf/util/evsel.h. Once you've done that do,

$ git add tools/perf/util/evsel.h

$ rm tools/perf/util/evsel.h.rej

Once you've fixed up all conflicts this way for the first patch, simply do,

$ git am --continue

And then you get to do things all over again for the next 10 patches ;-) Some may apply cleanly.

Step 4: Generate new patches

Once you've fixed up all 11 patches and tested them to ensure things work as you expect, you can generate new patches against v3.17.4 by doing,

$ git format-patch v3.17.4

TO Test Patches:

On Mon, 2014-11-24 at 23:40 +0000, Juvva, Kanaka D wrote:

> 5) For testing the kernel:

>

> Do you use make install in gruntley?

>

> What if the kernel doesn’t come up…

>

> Any ideas. If we have console access we can boot to another kernel.

> How about gruntley2..

>

> Did you test first on dev machine so that we can fix kernel crashes

> before trying on gruntley?

Tony, can we get Kanaka access to linux-tty.sc.intel.com?

Kanaka, once you've got access to linux-tty you can view the serial console of grantley2 by logging in and issuing the following command,

console grantley2

If some other user is already viewing the console (this happens

regularly) you'll only be provided with read-only access to the console.

Usually this is all that's require anyway.

For information on using the console program, launch it like above, and hit Ctrl+E then C then ?

As far as building and installing new kernels goes, I usually do the following from the kernel source directory (~mfleming/linux-3.17),

$ make -j48 bzImage modules

$ su -c 'make modules\_install install'

$ su -c reboot

Then view the output of `console grantley2` in another window to ensure your kernel comes up correctly :)

Username: [kankaka.d.juvva@intel.com](mailto:kankaka.d.juvva@intel.com)

[11/26/2014 10:07 AM] Fleming, Matt:

outlookir.intel.com

[11/26/2014 10:07 AM] Fleming, Matt:

port 993

[11/26/2014 10:08 AM] Fleming, Matt:

outgoing: port 25

--stats for individual g

Stat –g

arch/x86/include/asm/cpufeature.h | 9 +-

arch/x86/include/asm/processor.h | 3 +

arch/x86/kernel/cpu/Makefile | 2 +-

arch/x86/kernel/cpu/common.c | 39 +

arch/x86/kernel/cpu/perf\_event\_intel\_cqm.c | 1349 ++++++++++++++++++++++++++++

include/linux/perf\_event.h | 48 +

include/uapi/linux/perf\_event.h | 1 +

kernel/events/core.c | 63 +-

tools/perf/builtin-stat.c | 88 +-

tools/perf/util/evsel.c | 12 +-

tools/perf/util/evsel.h | 9 +-

tools/perf/util/parse-events.c | 2 +

tools/perf/util/pmu.c | 74 +-

tools/perf/util/pmu.h | 4 +

14 files changed, 1643 insertions(+), 60 deletions(-) create mode 100644 arch/x86/kernel/cpu/perf\_event\_intel\_cqm.c

arch/x86/include/asm/cat.h | 31 +++++++++++++++---------------

arch/x86/include/asm/rdt.h | 14 ++++++++++++++

arch/x86/kernel/cpu/perf\_event\_intel\_cqm.c | 20 +++++++------------