Status of Linux 3.x Real Time and Changes From 2.6

The current OSADL "Latest Stable" RT version is 2.6.33.7.2-rt30, but the current RT development release has moved forward to Linux 3.x.

The RT patches underwent a significant rewrite between Linux 2.6 and Linux 3.0. This presentation will discuss how the RT implementation has changed from 2.6 Linux, the current state of RT on 3.x Linux, and whether RT on 3.x is usable.

Frank Rowand, Sony Network Entertainment

September 20, 2012

120918_1913

The Plot

This talk will start at a high overview level then become more detailed and low level.

Some History -- 2004

... a number of projects working to implement realtime response have posted their work.

Whether any of the work described here will make it into the mainline kernel is another question.

... it may well be the sort of development that finally forces the creation of a 2.7 branch.

Jonathan Corbet
https://lwn.net/Articles/106010/

Some History -- 2008

The merging of the realtime Linux tree will be substantially complete by the end of the year. Your editor is out on a limb here; ...

So it seems likely that, by the end of 2008, the mainline Linux kernel will be fully capable of running in a realtime mode.

Jonathan Corbet
https://lwn.net/Articles/263129/

Some History -- 2009

The realtime patch set will be mostly merged by the end of the year. It really has to happen this time. What could possibly go wrong?

Jonathan Corbet
https://lwn.net/Articles/313615/

Some History -- 2010

The big kernel lock will be gone...

This work will be part of the larger job of getting the realtime preemption patch set into the mainline, but your editor dares not attempt another prediction on when that task will be complete.

Jonathan Corbet
https://lwn.net/Articles/368120/

How Does Realtime Go Mainline?

"Controlling a laser with Linux is crazy, but everyone in this room is crazy in his own way. So if you want to use Linux to control an industrial welding laser, I have no problem with your using PREEMPT_RT."

-- Linus Torvalds

https://rt.wiki.kernel.org/index.php/Main_Page

How Does Realtime Go Mainline?

So I can work with crazy people, that's not the problem. They just need to _sell_ their crazy stuff to me using non-crazy arguments, and in small and well-defined pieces. When I ask for killer features, I want them to lull me into a safe and cozy world where the stuff they are pushing is actually useful to mainline people _first_.

In other words, every new crazy feature should be hidden in a nice solid "Trojan Horse" gift: something that looks _obviously_ good at first sight.

The fact that it may contain the germs for future features should be hidden so well that not only is it not used as an argument ("Hey, look at all those soldiers in that horse, imagine what you could do with them"), it should also not be obvious from the source code ("Look at all those hooks I sprinkled around, which aren't actually used by anything, but just imagine what you could do with them").

-- Linus Torvalds
http://lkml.indiana.edu/hypermail/linux/kernel/1001.3/00384.html

Current PREEMPT_RT Versions

2.6.33.7.2-rt30 OSADL latest stable

https://www.osadl.org/Latest-Stable-Realtime.latest-stable-realtime-linux.0.html

Paul Gortmaker created a broken out version

http://marc.info/?l=linux-rt-users&m=129588844818236&w=2

2.6.34.8 from Paul (_not_ stable, best effort)

https://lkml.org/lkml/2011/3/4/281

3.0.43-rt65 stable, from Steve Rostedt
3.2.29-rt44
3.4.11-rt19

3.6 (soon) top of tree from Thomas Gleixner

Some Definitions

broken out patches

Each feature is in a separate patch file. The patch file may modify more than one source file. Also known as the 'quilt' model.

unified patch

One patch file contains all of the features.

PREEMPT_RT Patch Size

Features are added ---> patch gets bigger

Features are mainlined ---> patch gets smaller

Mainline Changes Related To RT

(from the beginning of RT)

- BKL gone
- lock notation and debugging features
- latency tracers
- priority inheritance
- mutexes
- robust futexes
- threaded interrupt handlers
- generic IRQ layer
- core timekeeping rewrite
- dynamic tick support
- high resolution timers
- dyntick patches

PREEMPT_RT Patch Size

Comparing 2.6 to 3.x

	files	lines	lines	
version	changed	plus	minus	
2.6.33.7.2-rt30	701	15723	4870	latest stable
3.0-rc7-rt0	500	9853	2573	first 3.x
3.4-rt8	394	9621	2306	May 2012 3.x

PREEMPT_RT Patch Size

Following graphs show July 2007 to May 2012 (2.6.22.1-rt2 to 3.4-rt7)

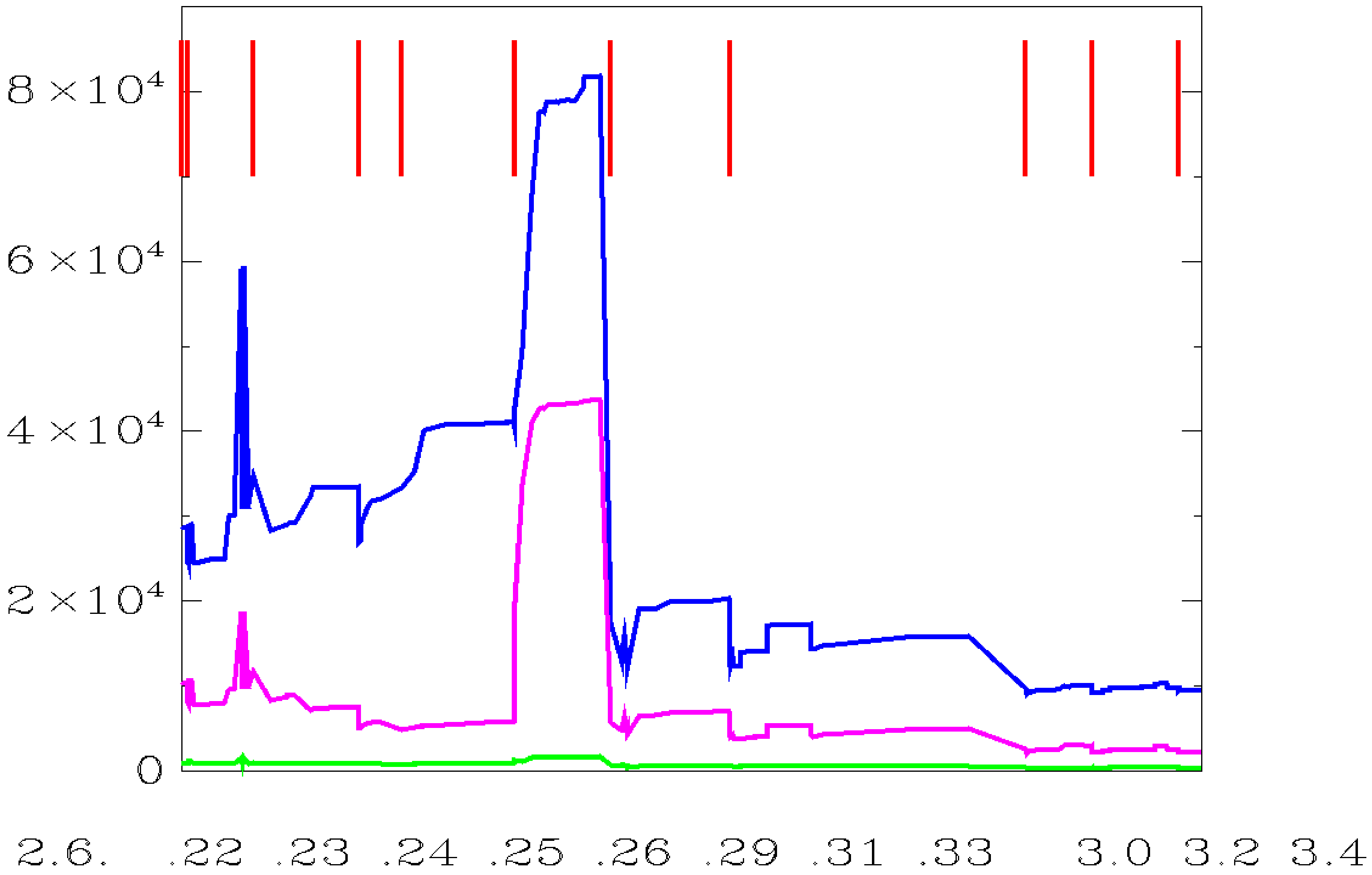
```
  - Number of files modified
  - Number of lines added
  - Number of lines deleted
```

PREEMPT_RT Patch Size

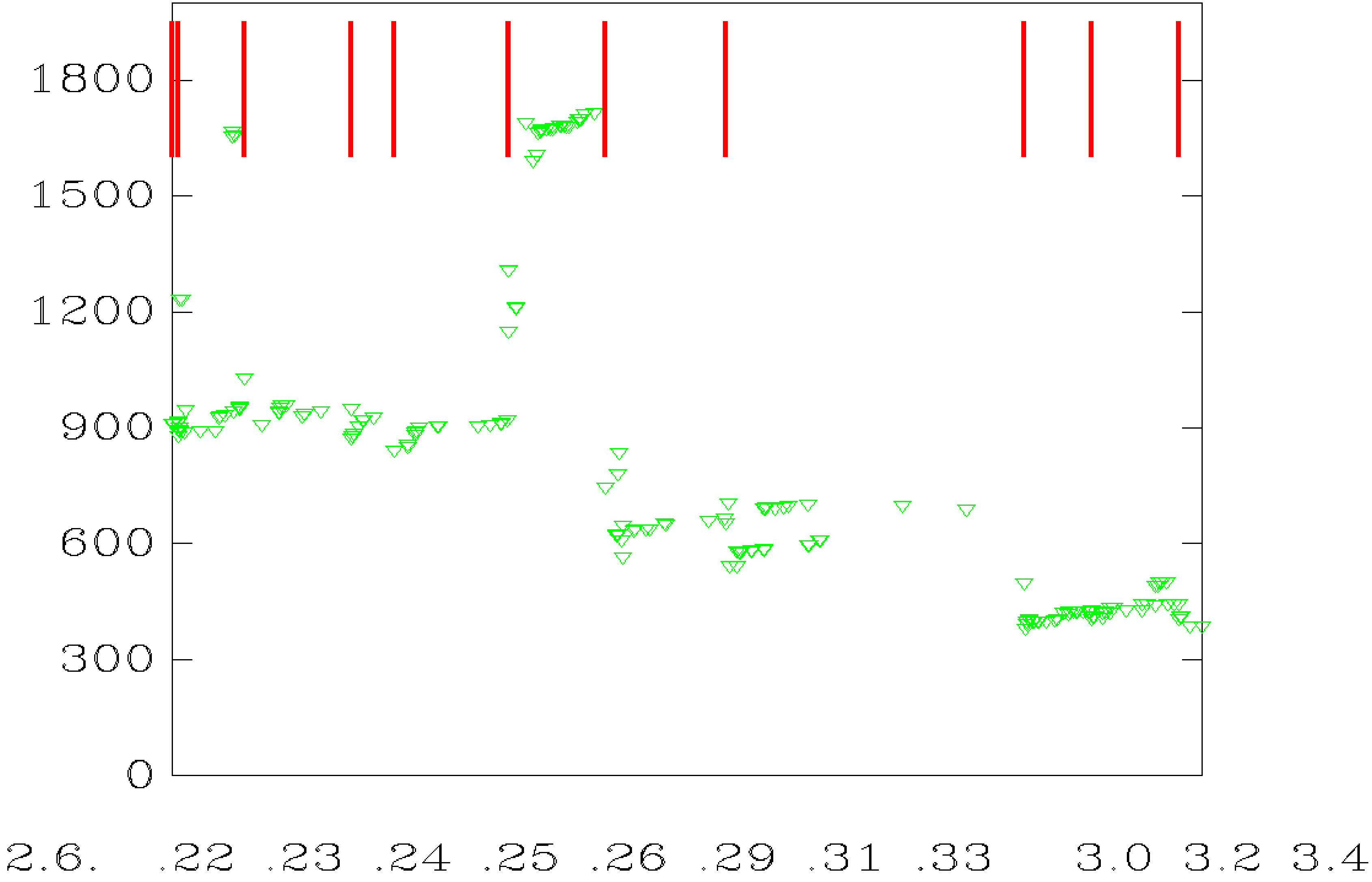
Some data points are excluded from the graphs

- extremely small values:
  - 2.6.29-rc8-rt1
  - 2.6.29-rc8-rt2
- extremely high values:
  - 2.6.29-rc8-rt3
  - 2.6.29-rc8-rt4

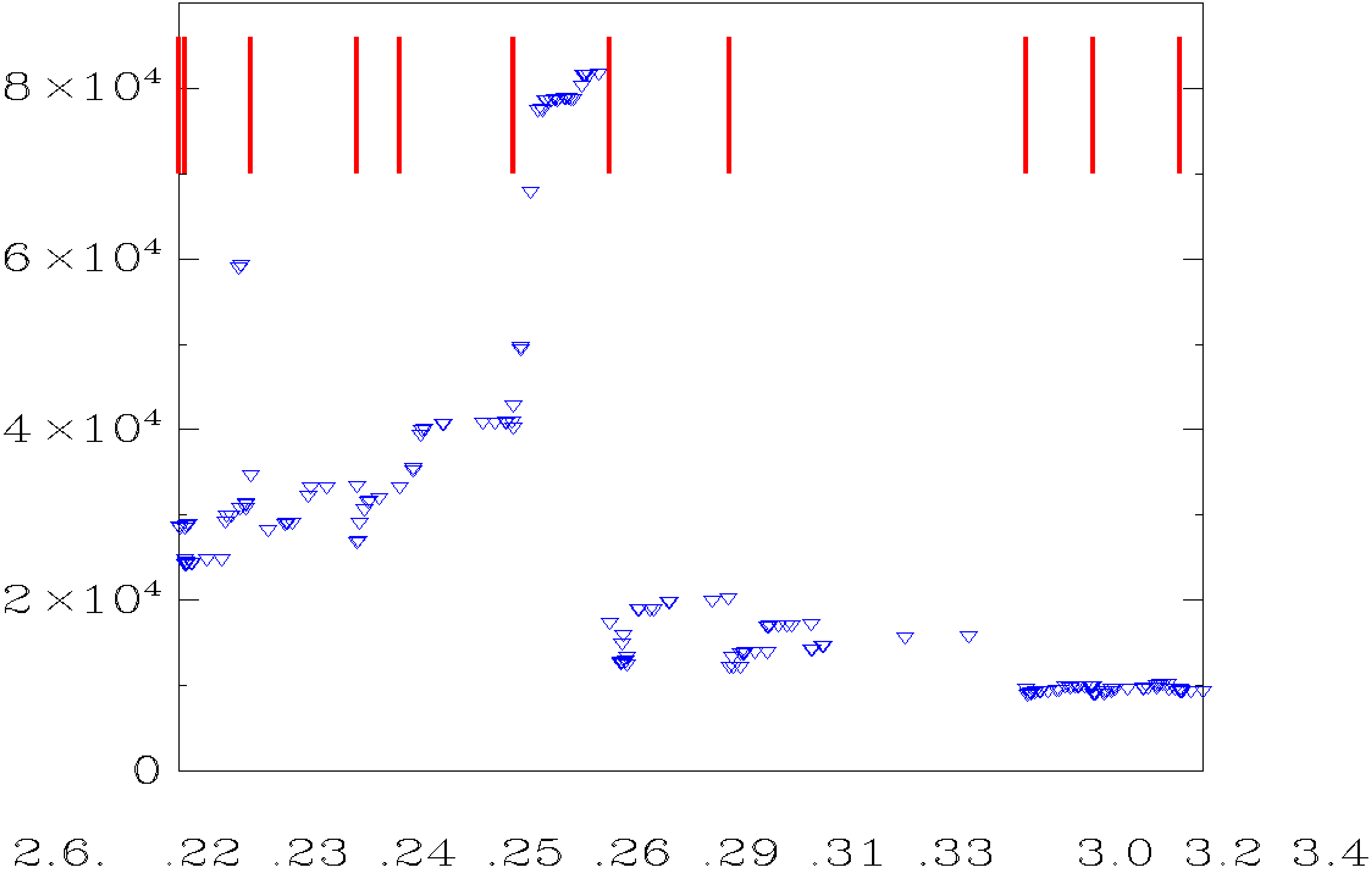
green: files blue: insertions pink: deletions



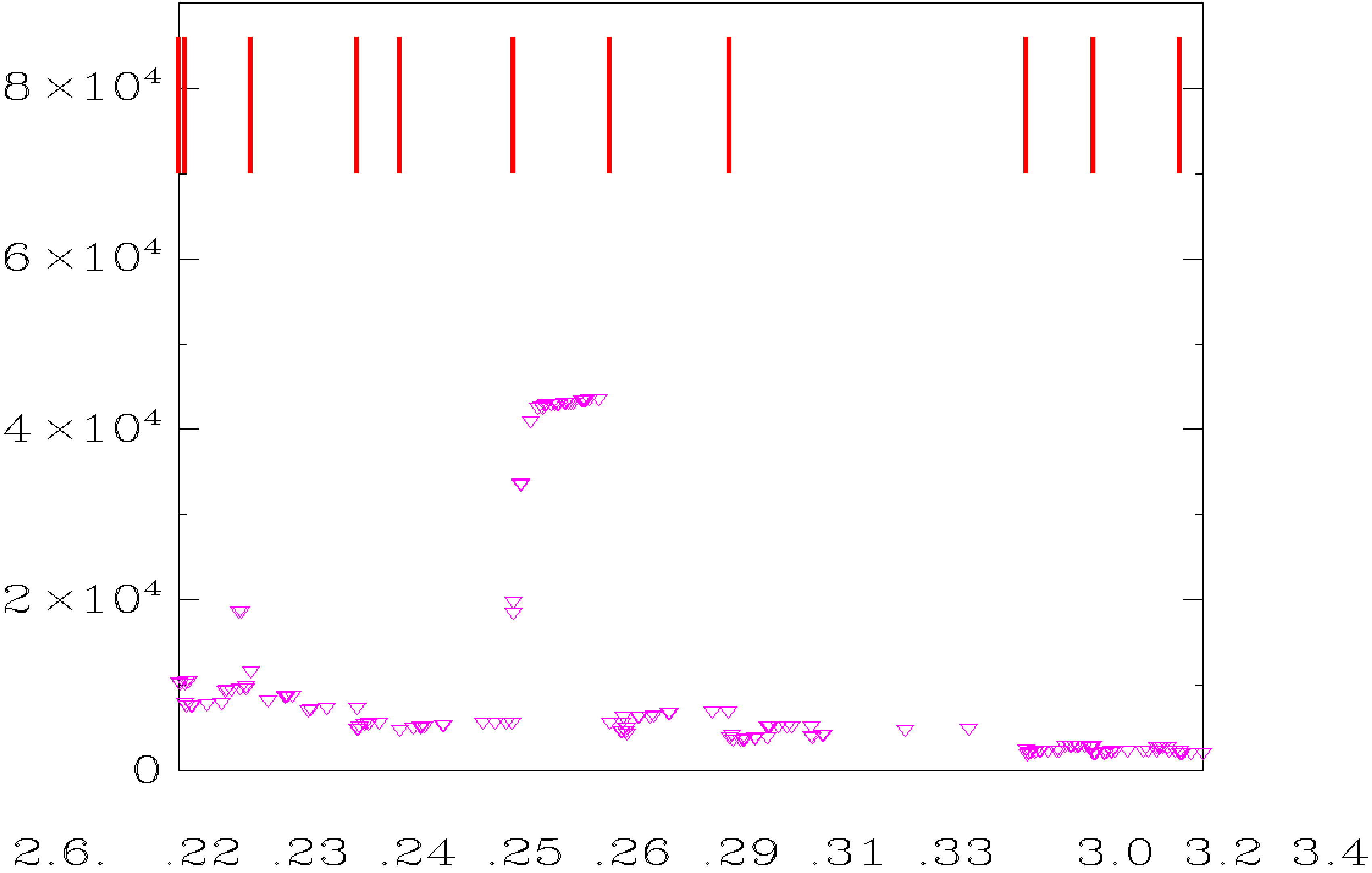
green: files



blue: insertions



pink: deletions



Contents of 3.4-rt8 RT Patches

# UPSTREAM changes queued for 3.3 or 3.2
1 patch

# Stuff broken upstream, patches submitted
1 patch

# Stuff which needs addressing upstream, but requires
# more information
2 patches

# Submitted on LKML, mips ML, ARM ML, ppc-devel
4 patches

Contents of 3.4-rt8 RT Patches

# Pending in peterz's scheduler queue
0 patches

# Stuff which should go upstream ASAP
44 patches

# Stuff which should go mainline, but wants some care
2 patches

# REAL RT STUFF starts here
198 patches

Stable Tree

Maintained by Steve Rostedt

Steve takes ownership of a branch when the top of tree development moves to a new release

Intent is to maintain stable RT tree for each mainline stable Linux kernel tree

Current stable tree versions:

- 3.0.43-rt65
- 3.2.29-rt44
- 3.4.11-rt19

Stable Tree

Policy based on the Greg KH stable trees policy

- Fixes only (no new features)
- Patch must first be in one of:
  - more recent tglx RT_PREEMPT patch
  - Linux mainline
  - Linux stable

To nominate a patch for stable:

Cc: stable-rt@vger.kernel.org

Stable Tree

Typical release cycle is a multi-stage process

1. Move existing RT patches to each new Greg KH stable version [release]
2. Add additional RT fix patches [-rc]
3. Add additional RT fix patches [release]

Steps 2/3 only occur if there are new RT patches.

All announced on lkml and linux-rt-users

Stable Tree

Patches can be dowloaded from

ftp://www.kernel.org/pub/linux/kernel/projects/rt/3.*/

Broken out patches (quilt model) through 3.0.14-rt31

Plan to create broken out patches for all releases

Unified patch for all versions.

Maintained in a public git repository

Most recent description:

ftp://www.kernel.org/pub/linux/kernel/projects/rt/3.*/README

https://lkml.org/lkml/2012/1/24/222

Stable Tree Public git Repository

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

v3.0-rt branch

- will never rebase
- should be what users develop git repositories with

Stable Tree Public git Repository

v3.0-rt-rebase branch

- will rebase at every -rt version
- should ***not*** be used by other developmental git repositories.
- will allow pulling out the commits that will apply to the stable tree

At each version, the rebase branch will be tagged with a "-rebase" name after it.

Stable Tree Public git Repository

“IMPORTANT NOTE: The rebase branch and tags are very low priority.

If it becomes too time consuming to maintain, I ***will*** stop adding them.

They will be created when I have time to do so.

This rebase branch is a convenience for people that may want it.

It may be discontinued at any time without notice!”

Stable Tree Public git Repository

If the rebase branch and tags exist, then the broken out patches and series file for quilt can be created.

Script to automate this is available from Frank Rowand. or at:

http://marc.info/?l=linux-rt-users&m=133886292607622&w=2

Stable Tree 3.0.x

Change for each release has been modest

The following slides detail the change in size between releases.

The change is reported as “-” if the release is moving to a new underlying Linux stable release. These releases contain no added RT patches.

The data is approximate due to tool issues.

Stable Tree 3.0.x

DELTA: patches, files, insertions, deletions

3.0.9-rt26	6	6	19	16
3.0.10-rt27	-			
3.0.11-rt28	-			
3.0.12-rt29	-			
3.0.12-rt30	11	15	303	181
3.0.14-rt31	-			
3.0.14-rt32	3	6	226	7
3.0.17-rt33	-			
3.0.18-rt34	-			
3.0.20-rt35	-			

Stable Tree 3.0.x

DELTA:	patches,	files,	insertions,	deletions
3.0.20-rt36	7	13	190	28
3.0.22-rt37	-			
3.0.23-rt38	-2	-3	-37	-13
3.0.23-rt39	24	80	1154	1045
3.0.23-rt40	6	7	43	18
3.0.24-rt41	-			
3.0.24-rt42	1	9	53	38
3.0.25-rt43	-			
3.0.25-rt44	2	2	4	4
3.0.26-rt45	-12	-27	-444	-444

Stable Tree 3.0.x

DELTA: patches, files, insertions, deletions

3.0.27-rt46	-			
3.0.28-rt47	-			
3.0.28-rt48	?	3	4	4
3.0.29-rt49	1	1	3	3
3.0.30-rt50	-			
3.0.31-rt51	-			
3.0.32-rt52	-			
3.0.33-rt53	-			
3.0.33-rt54	3	1	6	12
3.0.34-rt55	-			

Stable Tree 3.0.x

DELTA: patches, files, insertions, deletions

```
3.0.35-rt56     -
3.0.36-rt57     -
3.0.36-rt58    12    5    798   267
3.0.39-rt59     - *leap sec changes
3.0.40-rt60     -
3.0.41-rt61     -
3.0.41-rt62     1   -1      1     1
3.0.42-rt63     -
3.0.42-rt64     0    3      2    53
3.0.43-rt65     -
```

OSADL Latest Stable on 3.x

When will OSADL Latest Stable move to 3.x?

See:

linux-rt-users
From: Carsten Emde <C.Emde@osadl.org>
Date: Wed, 28 Mar 2012 15:08:31 -0700
Subject: Re: Determining latest stable release.

http://article.gmane.org/gmane.linux.rt.user/8115

OSADL Latest Stable on 3.x

Criteria:

- no known bugs or regressions
- all OSADL systems in the QA farm run one month under all appropriate load scenarios without any problem

'now are very close to label one of the 3.x kernels ... “Latest stable” '

OSADL Latest Stable on 3.x

'... the 3.x RT kernel is pretty stable and has impressive real-time capabilities. Such systems, if thoroughly tested, certainly may be used in a productive environment. However, the extra guarantee and confidence levels of an OSADL "Latest Stable" kernel unfortunately are not yet available.'

New or Changed Features

... a partial list

Performance Measurement

Performance measurement tools and kernel instrumentation were not fully functional in early 3.x RT versions.

Current 3.x RT versions are more complete.

migrate_disable()

New mechanism that in some situations can replace preempt_disable().

get_cpu_var()	uses preempt_disable()
get_local_var()	uses migrate_disable()
local_lock()	uses get_local_var()

CPU Hotplug

A little bit fragile...

Attempts to resolve the RT issues have led to the conclusion that the best approach is to refactor the architecture specific code into common shared code.

Summary of the redesign/rework discussion is at:

https://lkml.org/lkml/2012/3/19/350

Patches started appearing on lkml in April

Deadline Scheduler

A new scheduling class

Task registers:

- amount of cpu required to complete work
- when work must be completed
- how often task executes

Scheduler only allows task to register if there are sufficient cpu resources to guarantee task will complete work by deadline.

Deadline Scheduler

Juri Lelli has taken over from Dario:

> Is there any news on the possible integration of the EDF scheduling
> class (SCHED_DEADLINE) in the RT branch?

Cannot speak about "the integration" :-P, but I can tell you that I'm currently taking over Dario's work (he has a new job now, and I started working at his previous lab), even if he remains behind the scenes.

We focused on the RT branch (3.2-rc1-rtx) for the next release and I'm quite close to post it (hope before the end of November).

Re: EDF integration?
From: Juri Lelli
Date: 2011-11-15 9:59:20
http://marc.info/?l=linux-rt-users&m=132135121824549&w=2

Deadline Scheduler

[RFC][PATCH 00/15] sched: SCHED_DEADLINE v5
From: Juri Lelli
Date: Wed May 23 2012 - 17:44:18 EST

http://lkml.indiana.edu/hypermail/linux/kernel/1205.2/04642.html

Previous versions have reviews/comments from

- Peter Zijlstra
- Thomas Gleixner
- Steven Rostedt
- others...

Deadline Scheduler

Peter Zijlstra would like to accept it into mainline Linux kernel, but needs some justification to support the merge.

If you have a use case, email Peter and lkml.

CPU Isolation

Dedicate a processor to a specific task (kernel space or user space).

Eliminate all normal kernel interrupts and overhead on the processor.

Goal:

- minimize latency
- maximize throughput

CPU Isolation

Early reactions were

- skeptical
- somewhat hostile
- suggesting different solutions

CPU Isolation

[git pull] CPU isolation extensions
From: Max Krasnyansky
https://lkml.org/lkml/2008/2/7/1

Linus, please pull CPU isolation extensions from ...

The patchset consist of 4 patches.

- Make cpu isolation configurable and export isolated map
- Do not route IRQs to the CPUs isolated at boot
- Do not schedule workqueues on the isolated CPUs
- Do not halt isolated CPUs with Stop Machine

CPU Isolation - In Progress

[PATCH v7 0/8] Reduce cross CPU IPI interference
From: Gilad Ben-Yossef

- https://lkml.org/lkml/2012/1/8/109
- https://lkml.org/lkml/2012/1/26/70

CPU Isolation - In Progress

From: Frederic Weisbecker

[RFC PATCH 00/32] Nohz cpusets (was: Nohz Tasks)

https://lkml.org/lkml/2011/8/15/245

Status of Nohz cpusets (adaptive tickless kernel) for January 2012

https://lkml.org/lkml/2012/1/17/486

May include tick avoidance when there is a single userland process running.

https://tglx.de/~fweisbec/TODO-nohz-cpusets

CPU Isolation

One further suggestion for moving solutions forward:

- create a partition using cpusets
- run a while(1); userspace in it, and trace the thing
- Work up patches that remove all useless perbutations

Real-Time virtualisation

Not very visible in the Linux RT email lists

Real-Time virtualisation

The following 4 slides are from:

Real-Time Linux Failure
Frank Rowand
ELC 2010

Virtualization

Example Issue 1

Additional overhead of hypervisor

But real-time is not fast, it is determinism.

So if the deadlines are met, the extra overhead is not a problem.

Virtualization
and
Real Time

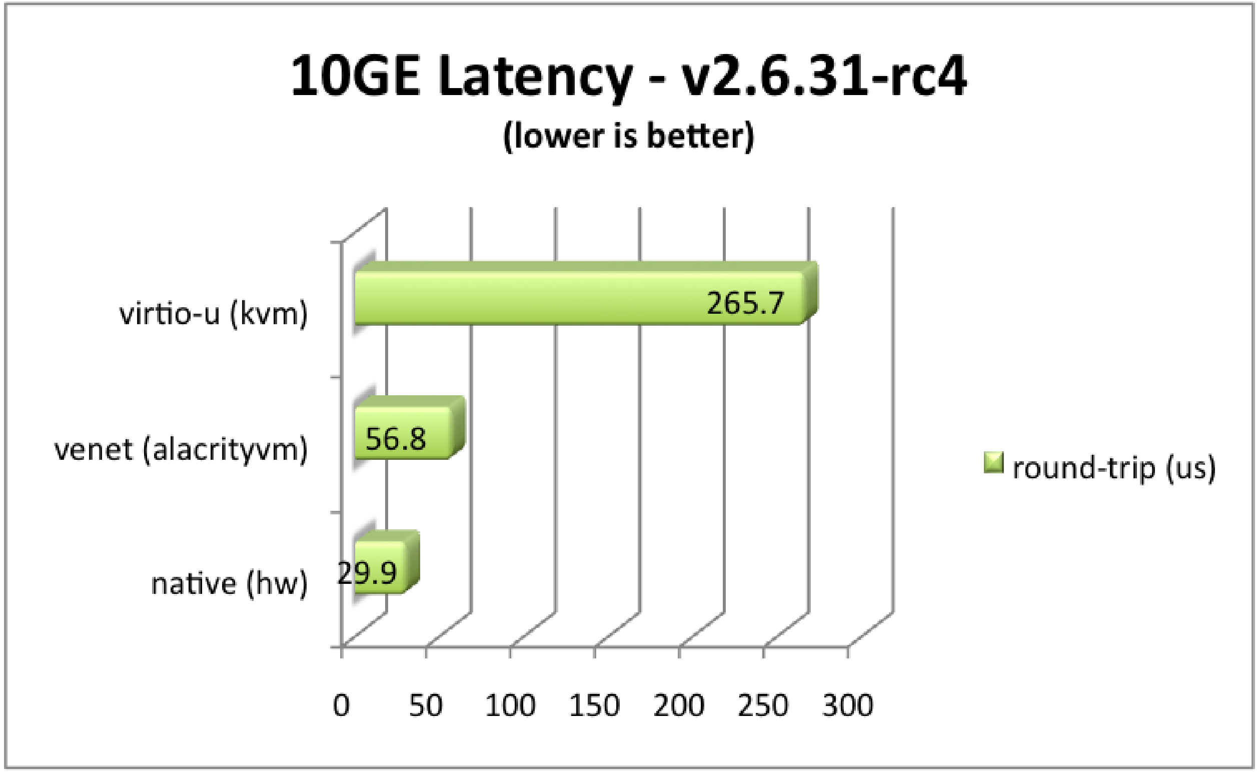
Frank's viewpoint:

On this path lies insanity.

But there are people who are braver than Frank.

Braver Than Frank, 1

Example of results



source (14 August 2009):
http://developer.novell.com/wiki/index.php/AlacrityVM

Braver Than Frank, 2

http://www.osadl.org/Abstract-20-Towards-Linux-as-a-Real-Tim.rtlws11-abstract20.0.html

“... research work on improving the real-time qualities the Linux hypervisor KVM can provide to its guests.”

“... a new paravirtualized scheduling interface ... allows guests to influence the scheduling parameters of their virtual CPUs (VCPU) on the host.”

“This ... enables the ... host to account for real-time load inside guest systems ...”

Real-Time virtualisation

Development is continuing. One example is:

Using KVM as a Real-Time Hypervisor
Jan Kiszka
KVM Forum 2011

slides at:

http://www.linux-kvm.org/page/KVM_Forum_2011

video at:

http://www.montanalinux.org/video-kvm-forums-2011.html

Tools:

Change is not revolutionary, but tools are slowly evolving and improving.

Fixes made so that tools run on 3.0 RT

rt-rests git repository is now at:

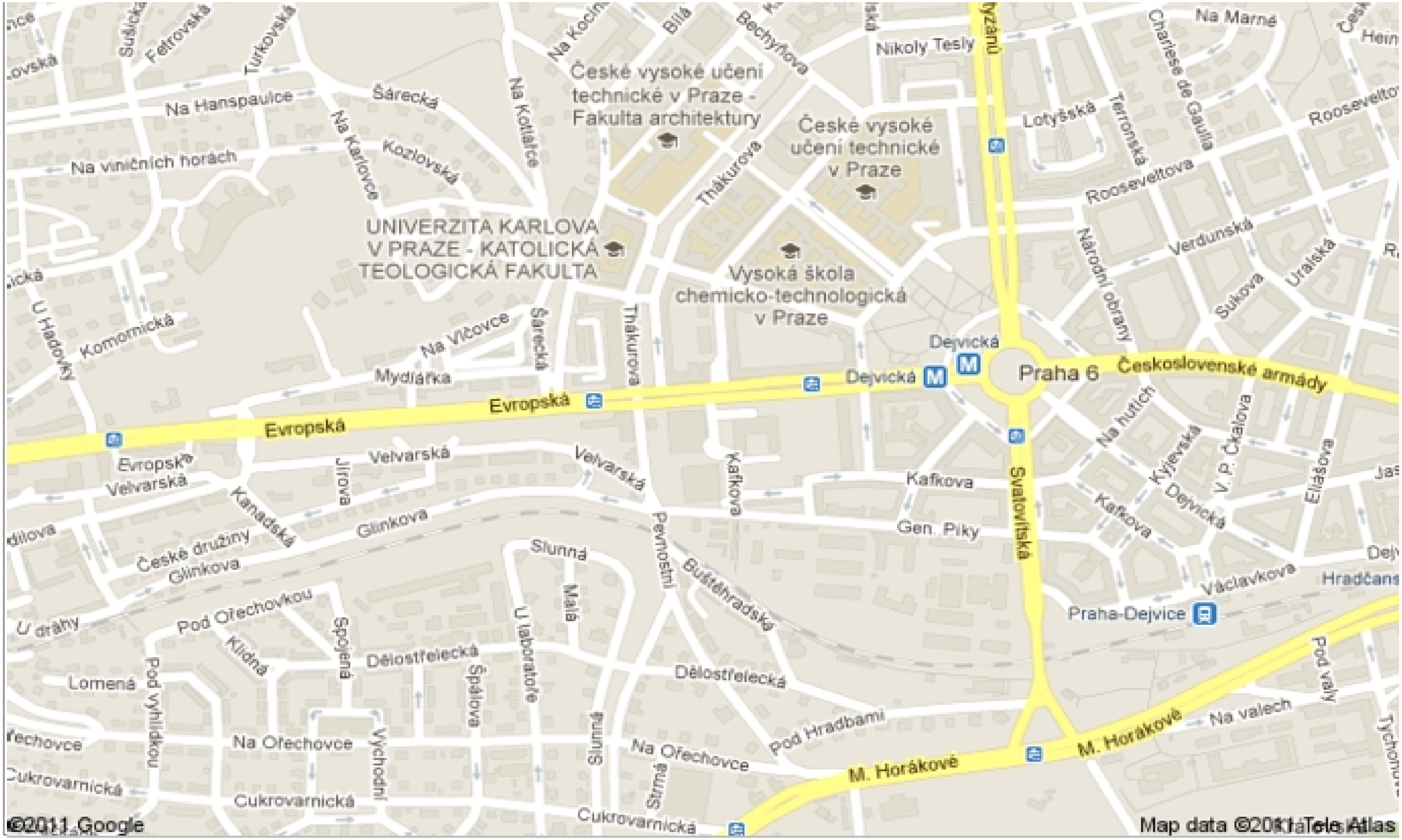
git.kernel.org/pub/scm/linux/kernel/git/clrkwllms/rt-tests.git

What is in the future?

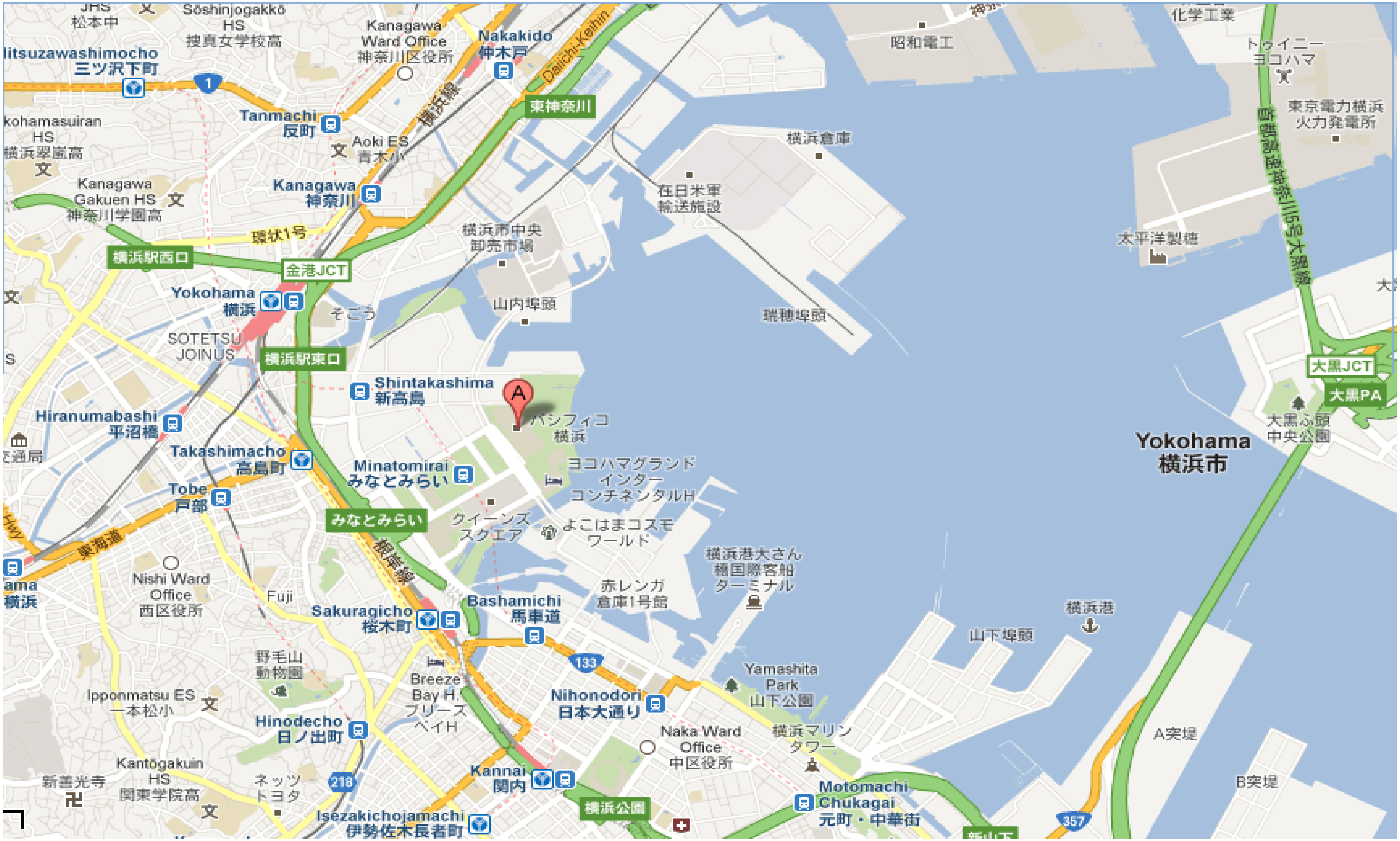
Linux Real-Time development

Here is a roadmap

Updated for ELCE 2011



Updated for LinuxCon Japan 2012



What events are coming up?

14th real time linux workshop (rtlws)
RT Summit

- October 18 - 20, 2012
  University of North Carolina at Chapel Hill

LinuxCon Europe /
Embedded Linux Conference Europe
3 or 4 real time talks scheduled at ELCE

- November 5 - 7

How to get a copy of the slides

1) leave a business card with me

2) frank.rowand@am.sony.com