Leveraging Linux in a Lousy Economy

Theodore Ts'o

THE LINUX FOUNDATION

The 56 slide Deck of Doom...

I prefer Matt Maroon's “Whiner Jerkins” deck

How’d We Get In This Mess?

East Coast Finance Douchebags

- Harvard MBAs.
- Never trust anyone with degree from anywhere other than Stanford or MIT.
- If they were so smart, they would have been VCs to begin with.

Got Greedy

- Needed more money for yachts and personal trainers for their trophy wives.
- Gambled by borrowing, then betting that people who had poor credit ratings were really just misunderstood.
- Didn’t understand anything about the web, but bought Google until P/E ratio hit 872.

Systemic Collapse

- Turns out poor people can’t pay the mortgage on their $500k homes by selling their furniture and TVs on Craigslist for more than about 26 months.
- Analysts downgraded GOOG to hold when market cap exceeded total supply of money on earth.

Saving Money Using Linux

- **Value of Linux**
  - Cost to develop the Linux Kernel from scratch: **$1.4 billion dollars**
  - Cost to build the Fedora 9 distribution: **$10.8 billion dollars**
- **No cost of acquisition fees**
- **No cost of per unit licensing costs**
- **Wide array of support options**
  - Do it yourself (thanks to Open Source)
  - Free e-mail support from the community
  - Paid enterprise-style support 24x7 with guaranteed response times
  - Competition between support options leads to lower costs
- **Huge numbers of college students are graduating with deep Linux skills**

Leveraging Linux Effectively

- How to protect your “secret sauce”
- Pick your hardware carefully
- Don't reinvent the wheel
- Reduce your maintenance costs
- Co-create with your consumers

Protecting your secret sauce

- **Secret sauce is important**
  - Key question which every VC asks: “What prevents a competitor from entering your market?”
  - It's the key to monetizing the investment in your company
- **Where to put your secret sauce**
  - User space
  - Hardware
  - On your servers, on the other side of the network connection
- **Don't try to hide your secret sauce in the kernel**
  - Legal reasons
  - Practical reasons
  - Marketing reasons

The Open Source Licensing Rules

- **Make sure that you (and your engineers) understand the rules**
  - Make sure they are reviewed regularly
  - Both kinds of rules: legal and community standards
- **Requirements of the GPL**
  - Anything derived from the GPL must be made available in source form
  - "What is a derived work?"
    - Talk to a lawyer... answer may very depending on jourisdiction
    - Play it safe --- important from a community relations POV anyway
- **Are Binary Drivers OK?**
  - Commonly done; may be legal in some circumstances/jourisdictions
    - Legally, no court decisions on point
  - Frowned on by the community
  - Many technical disadvantages of binary drivers

Myth: Kernel code is harder to reverse engineer

- **Reality: any code can be reverse engineered by someone who is motivated**
- **Linux has many kernel debugging facilities**
  - As of 2.6.26, Linux has an in-tree kernel debugger
    - Before that, there were plenty of out-of-tree debuggers
  - Kprobes and systemtap make it easy to set trace points inside kernel code
- **Plenty of choke points where it is easy to add tracing code – for example, where a USB packet is sent to the hardware**
  - This is how most USB devices are reverse engineers to write Linux device drivers; it is very easy to intercept USB I/O in Windows.
- **Cold reality: Kernel code is harder to debug**
  - Minimizing the amount of code in the kernel is just smart engineering

Pick your hardware carefully

- **If you are not creating your own custom peripherals/devices, pick hardware that has open source device drivers which are in the mainline kernel source; this will reduce your costs**
- **In almost all cases, there are alternatives that have in-tree device drivers**
  - Intel and ATI have open source drivers
  - Intel and Atheros have open source in-tree drivers
- **Demand it from your suppliers**
  - That's what has happened in the enterprise server space
- **Out-of-tree binaries with binary blobs not as good**
  - Such drivers are very kernel version specific
  - If you need to customize your kernel, they may not work
  - Hard to debug if they cause problems
  - No guarantee that the manufacturer will continue to update the driver for new kernels

Don't reinvent the wheel

- **Before you add functionality to the kernel, check very carefully; someone may have already done something similar or identical**
  - Could meet your needs with at most very minor modifications
- **Do you need to modify the kernel at all? Linux is used by a large number of users and companies. The facilities are deliberately made to be general.**
- **Many channels for doing this research**
  - Ask on a mailing list (“I need a way of controlling what happens the system critically runs out of memory”)
  - At a conference in “the hallway track”
  - Hiring a consultant or an expert employee

Reduce your maintenance costs

- **You will probably at some point want to release a follow-on product; that product will probably want use the latest technology**
  - Example: Sony eReader PRS-505 used Linux 2.4.17; the PRS-700 uses Linux 2.6.23
  - Many reasons: better power efficiency, new kernel facilities etc.
- **If any kernel modifications are submitted upstream, your long-term maintenance costs will be lower; no need to port your code because others will do it for you**
- **This may require that your kernel modifications be made general; the community will likely help you do this.**

Co-create with your consumers

- **Lead adopters like to be able to “tweak” their products**
  - Tivo
  - Palm Pilot
- **The product should be able to stand alone without the tweaks**
  - **That's how the reviewers will review your product**
  - **But lead adopters will be delighted if they add new secondary features**
- **Not necessary to make it easy; but counter-productive to try fight them**
  - “Your customer is not the enemy” --- Lesson which the RIAA has ***still*** not learned...
- **Adding some easy “customizations” which the UI designers didn't want to expose can be a good thing**

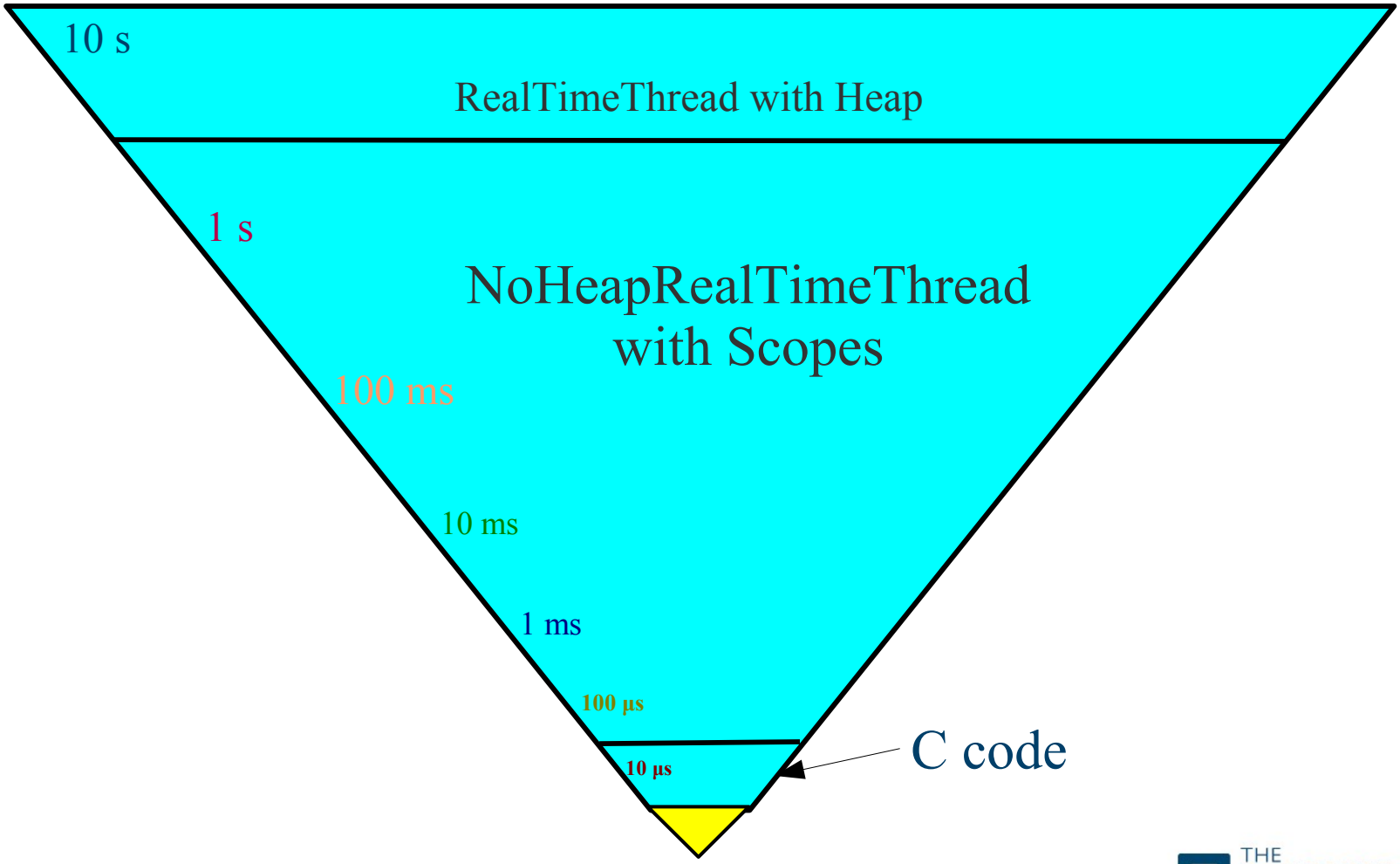
Case Study: Real-Time Linux on a warship!

- **Navy contracted to Raytheon to design the computer systems on the next generation Destroyer, or the DDG-1000 project.**
  - http://www.naval-technology.com/projects/dd21
- **Real-time Java from IBM running on Real-time Linux was key to Raytheon choice to partner with IBM instead of Sun.**
- **IBM Blades and real-time JVM used to build the central data centers.**

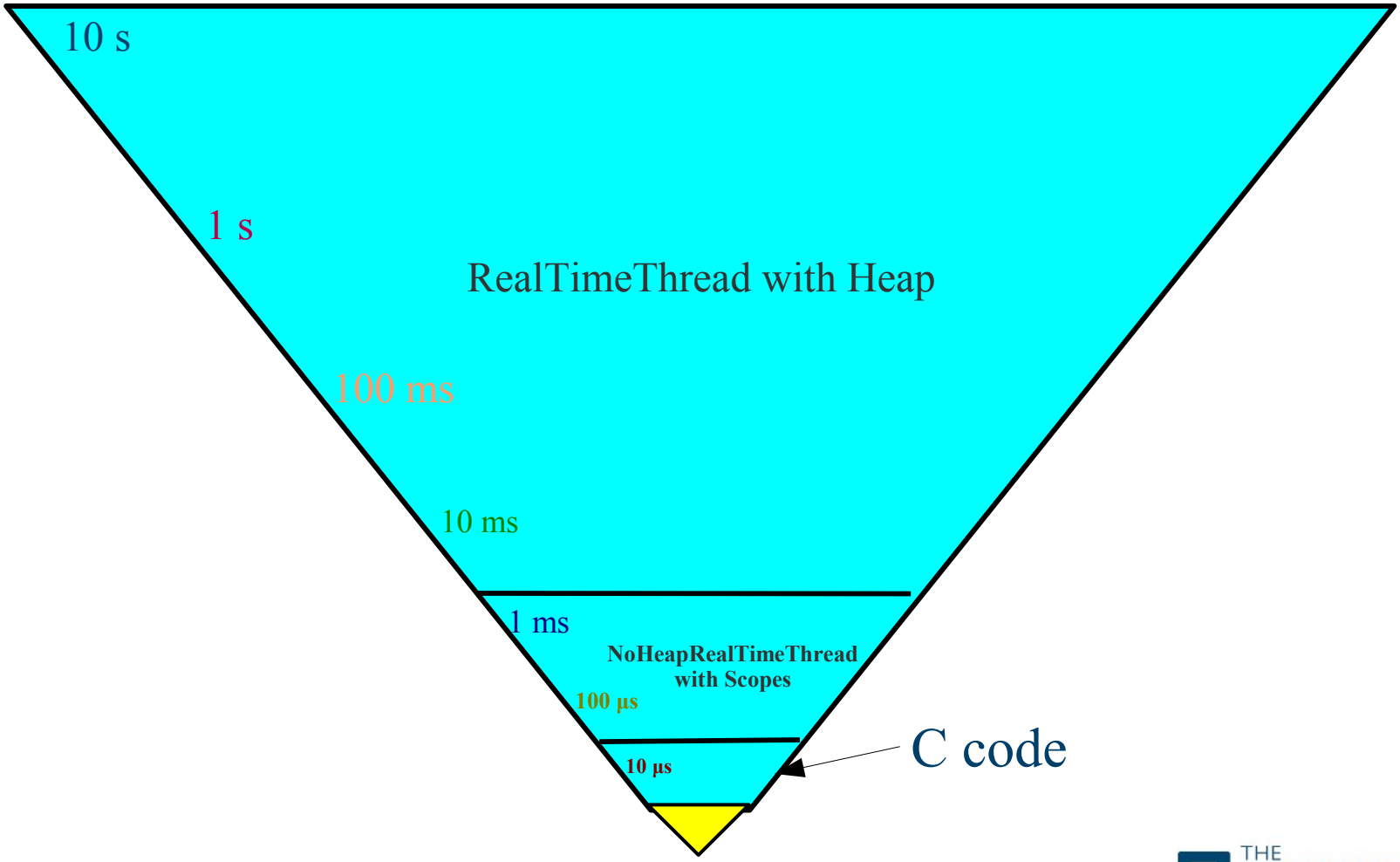
IBM's Real-Time Java product

- **Websphere Real-Time**
  - Websphere Real-Time 1.0 first delievered a fully supported GA product in August 2006
  - Paired with “patches and scripts” which extend a RHEL4 32-bit x86 installation to provide real-time functionality
- **WRT provides a High Predictable Java runtime**
  - Real-time garbage collection
    - Real time latency down to ~1ms with garbage collection
    - Latency using NHRT down to 50us
  - Static and Dynamic Compilation
  - Full support for RTSJ (JSR #1)
  - Java SE 5.0 Compliant
  - Rigorously tested on an RT Linux OS and IBM System X

Without Websphere Real-Time



With Websphere Real-Time



IBM's Real-Time Linux Extensions

- **Patches and scripts to update a 32-bit x86 RHEL4U2 system**
  - 2.6.16 Linux kernel with CONFIG_PREEMPT_RT patch from Ingo Molnar
    - high resolution timers
    - kernel and userspace priority mutexes
    - hardware and software interrupts run as kernel threads
    - latency measured at 25us (to six 9's)
  - 2 new interfaces added to glibc to support priority inheritance
  - enhancements to the PAM libraries to allow non-root users access to real-time facilities
  - bug fixes and stabilization patches
  - **Full binary compatibility with RHEL 4 userspace applications**
  - **Support provided via IGS Supportline and the IBM Linux Technology Center**
- **Development time: 9 months, from start to 1.0 GA release**

Result: A Happy Customer (and a lot of HW sales)

Two Chips per Blade (Quad Core in 2008)

- DS4700- 4.8 TB in 3U space
- Scales up to 33.6 TB in 42U

- Blade Center
- Up to 14 blades per chassis
- Opteron or PowerPC
- RT Linux

- TS3100 Tape Library
- LTO-3

The Bottom Line...

- **A Lot of Value Locked inside Linux and Its Ecosystem**
  - Almost one and a half billion dollars in the kernel
  - Over 10 billion in a Linux Distribution
- **Not all of that value will be suitable for your needs**
  - But a lot of it can be!
  - There are people who are quite happy and willing to point out where the value might be for you, If you can describe your problem in general terms to them.
- **“Stone Soup” rules apply**
  - The “price” for taking advantage of this value...
  - Some of these rules are legally enforceable, some are community standards
  - It is generally in your best interest to follow them anyway

And now a word from our sponsor...

- **Linux Foundation Collaboration Summit**
  - Will be in San Francisco, April 8-10th
  - Colocated with:
    - Linux Filesystem and Storage Workshop
    - CELF Embedded Linux Conference
    - Moblin Developer Summit
  - Training will be available at the event
    - Embedded Linux
    - Writing Device Drivers
    - Real-time Linux
  - Meet with the best minds in the Linux industry, from business to community leaders

THE LINUX FOUNDATION

Other Linux Foundation Activities

- **Other events, including:**
  - Linux Kernel Summit
  - Plumber's Conference
  - Legal Summit
- **Collaboration Forums**
  - End User Council
  - Vendor Advisory Council
  - Technical Advisory Board
  - Technical Workgroups: LSB, Accessibility, Carrier Grade Linux, Driver Backport, Desktop, Green Linux, Open Printing, FOSSbazaar
- **Promoting Linux by serving as a neutral spokesperson for Linux and generating original content that advances the understanding of the Linux platform**
- **Protecting Linux**
  - Sponsors key Linux developers through the Fellowship program
  - Administers the Linux trademark

Legal Statement

- This work represents the view of the author(s) and does not necessarily represent the view of IBM or of the Linux Foundation.
- IBM is a registered trademark of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Other company, product, and service names may be trademarks or service marks of others.