Our Troubles with Linux and Why You Should Care

Ashif Harji, Peter Buhr and Tim Brecht

University of Waterloo

David R. Cheriton School of Computer Science
Our Troubles with Linux

• Encountered during Ashif Harji’s Ph.D.
  – Over a number of years (N)
  – N much larger because of Linux kernel bugs
• Build, compare, improve high-perf web servers
• Found 3 Linux kernel bugs
  – Usually required rerunning previous experiments

• Serious implications
Why you Should Care

• Performance bugs exist
  – Harder to identify & fix than correctness bugs
• Can be serious (e.g., non-repeatable experiments)
  – Your results and those of others may be bogus
• They can be long lived (e.g., > 3 years)

• Serious implications
Why you Should Care

• Performance bugs exist
  – Harder to identify & fix than correctness bugs
• Can be serious (e.g., non-repeatable experiments)
  – Your results and those of others may be bogus
• They can be long lived (e.g., > 3 years)

• Serious implications

Likely true of other open-source software projects without rigorous performance regression testing
Linux as a Research Platform

+ Open source
+ Used in production environments
+ Rapid kernel development cycle

- Large and complex
- Difficult to properly configure and tune
- Rapid kernel development cycle
Linux as a Research Platform

+ Open source
+ Used in production environments
+ Rapid kernel development cycle

– Large and complex
– Difficult to properly configure and tune
– Rapid kernel development cycle

Can introduce bugs / performance regressions
Our Experiences with Kernel Bugs

- **Bug 1**: Small file evictions

  *Kernel versions affected*: 2.6.11 to 2.6.21.17

  *Duration*: 02-Mar-2005 to 04-Aug-2007 (~2.4 years)
Our Experiences with Kernel Bugs

- **Bug 2**: Prefetching Disabled

  *Kernel versions affected*: 2.6.12 to 2.6.22.19
  
  *Duration*: 17-Jun-2005 to 26-Feb-2008  (~ 2.7 years)
  
  
  \[
  rc = \text{sendfile}(fd, \ sd, \ \&\text{offset},\text{bytes} \ldots)
  \]
Our Experiences with Kernel Bugs

• **Bug 2**: Prefetching Disabled

  *Kernel versions affected*: 2.6.12 to 2.6.22.19

  *Duration*: 17-Jun-2005 to 26-Feb-2008  (~ 2.7 years)

  \[ rc = \text{sendfile}(fd, sd, \&\text{offset}, \text{bytes} \ldots) \]
Our Experiences with Kernel Bugs

- **Bug 2**: Prefetching Disabled

  *Kernel versions affected*: 2.6.12 to 2.6.22.19

  *Duration*: 17-Jun-2005 to 26-Feb-2008  (~ 2.7 years)

  
  \[
  r_{c} = \text{sendfile}(fd, sd, &\text{offset}, \text{bytes} ...) 
  \]

  
  Disk: 0 – 128 KB
Our Experiences with Kernel Bugs

- **Bug 2**: Prefetching Disabled

  *Kernel versions affected*: 2.6.12 to 2.6.22.19
  
  *Duration*: 17-Jun-2005 to 26-Feb-2008 (~ 2.7 years)

  \[ rc = \text{sendfile}(fd, sd, &\text{offset}, \text{bytes} \ldots) \]

Disk : 0 – 128 KB

Sock : 60 KB
Our Experiences with Kernel Bugs

• **Bug 2**: Prefetching Disabled

  *Kernel versions affected*: 2.6.12 to 2.6.22.19

  *Duration*: 17-Jun-2005 to 26-Feb-2008 (~ 2.7 years)

  \[
  rc = \text{sendfile}(fd, \, sd, \, \&\text{offset,bytes} \, \ldots)
  \]

  Disk: 0 – 128 KB,  Req: 60 KB – end. Not from last read
Our Experiences with Kernel Bugs

- **Bug 2**: Prefetching Disabled

  *Kernel versions affected*: 2.6.12 to 2.6.22.19

  *Duration*: 17-Jun-2005 to 26-Feb-2008 (~ 2.7 years)

  \[ rc = \text{sendfile}(fd, sd, &offset, bytes \ldots) \]

  ![Diagram of disk and sockettransfer]

  Disk: 0 – 128 KB,  Req: 60 KB – end. Not from last read

  Sock: 60 KB

  Not interpreted as seq.

  Disable Prefetching
Our Experiences with Kernel Bugs

• **Bug 3**: Erratic Page Evictions

*Kernel versions affected*: 2.6.23 to at least 2.6.36.2

*Duration*: 09-Oct-07 to at least 09-Dec-10 (> 3 years)
Erratic Page Evictions

![Graph showing erratic page evictions over different requests per second. The x-axis represents requests per second ranging from 40000 to 70000, and the y-axis represents Mbps ranging from 0 to 5000. Two lines are shown: one for 2.6.24.3+patch, indicating a pattern of evictions.]
Erratic Page Evictions

![Graph showing erratic page evictions with two lines representing different versions of software. The green line indicates 2.6.24.3+patch, and the pink line indicates 2.6.36.2 run 1. The x-axis represents requests per second (Requests/s), and the y-axis represents Mbps.](image)
Erratic Page Evictions

![Graph showing Mbps vs Requests/s with different line styles for 2.6.24.3+patch, 2.6.36.2 run 1, and 2.6.36.2 run 2.]
Erratic Page Evictions

![Graph showing erratic page evictions with different configurations and data points.](image)
Erratic Page Evictions
Others’ Experiences

Source: http://kernel-perf.sourceforge.net
Others’ Experiences

Source: http://kernel-perf.sourceforge.net
Consequences

• Used patched 2.6.24.3 kernel, 1 new project FreeBSD
• Problems in published papers
• Must find underlying cause for performance change
  – May be just masking a kernel bug
• Fixing problems / kernel bugs
• Kernel upgrading problems
  – Sometimes old bugs not fixed and new bugs appear
  – Mistaken belief that newer kernels are better
  – Reviewers “results are not valid, use latest kernel”
Consequences

• Used patched 2.6.24.3 kernel, 1 new project FreeBSD
• Problems in published papers
• Must find underlying cause for performance change
  – May be just masking a kernel bug
• Fixing problems / kernel bugs
• Kernel upgrading problems
  – Sometimes old bugs not fixed and new bugs appear
  – Mistaken belief that newer kernels are better
  – Reviewers “results are not valid, use latest kernel”
Fixing a Kernel Bug (in Principle)

- Track the bug into the kernel
- Determine fix and performance test it (rerun expts!)
- See if the bug exists in the most recent kernel
- Try the fix in the most recent kernel
  - Often code has changed so patch fails
  - Expertise lost, understand new code, design fix
- Write stand-alone pgm to demonstrate bug
- Submit bug report with example pgm
Fixing a Kernel Bug (in Practice)

- Not everyone can find and fix kernel bugs
- Trying to fix a moving target
- Thankless, little credit, not publishable

- Getting a fix into mainline kernel is not easy
- Performance bug reports may not be well received
  - Need to have a thick skin
  - Need to be persistent
Fixing a Kernel Bug (in Practice)

• Not everyone can find and fix kernel bugs
• Trying to fix a moving target
• Thankless, little credit, not publishable

• Getting a fix into mainline kernel is not easy
• Performance bug reports may not be well received
  – Need to have a thick skin
  – Need to be persistent

Very difficult, time consuming, few benefits
Our Experience

• 1\textsuperscript{st} bug : fix made it into the mainline kernel
• 2\textsuperscript{nd} bug : gave up part way through, new kernel changed significantly
• 3\textsuperscript{rd} bug : didn’t even try
Possible Recommendations

• Kernel developers:
  – Need systematic, sustained, perf regression testing

• Researchers:
  – Have a look at Enterprise distributions?
  – Check performance regression tests (on web sites)
  – Recheck results with each kernel upgrade
  – Check results for repeatability
  – Appeal to intuition, are results as expected
Summary

• Find underlying cause of performance changes
• Latest kernel not always the best or most appropriate
  – Reviewers – justify why latest kernel is required
  – Authors – explain why latest kernel is not required
• Papers exist that are likely incorrect or flawed
• CS needs to embrace scientific method
  – Validate / verify results of others
  – Understand when results are and are not valid
  – Publishable in main tracks of top conferences