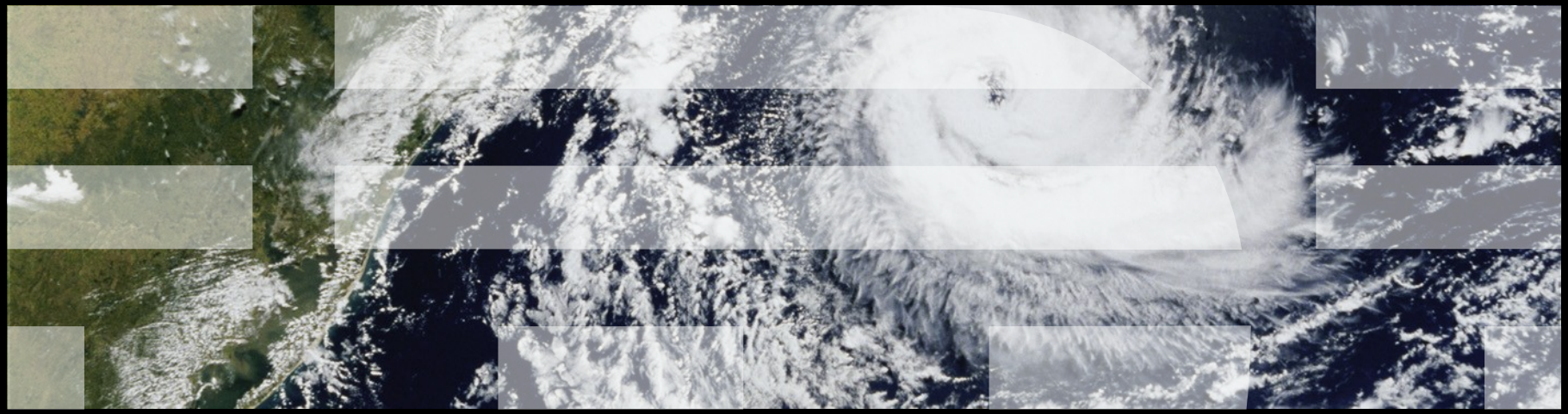
Paul E. McKenney – Distinguished Engineer, Linux Technology Center. Linaro Technical Steering Committee Member

4 April 2012

Whacking Droids: *

How to Extract Requirements from Flame Wars



* With apologies to Jon Corbet.

© 2009 IBM Corporation

Such A Small Thing To Cause So Much Trouble!!!

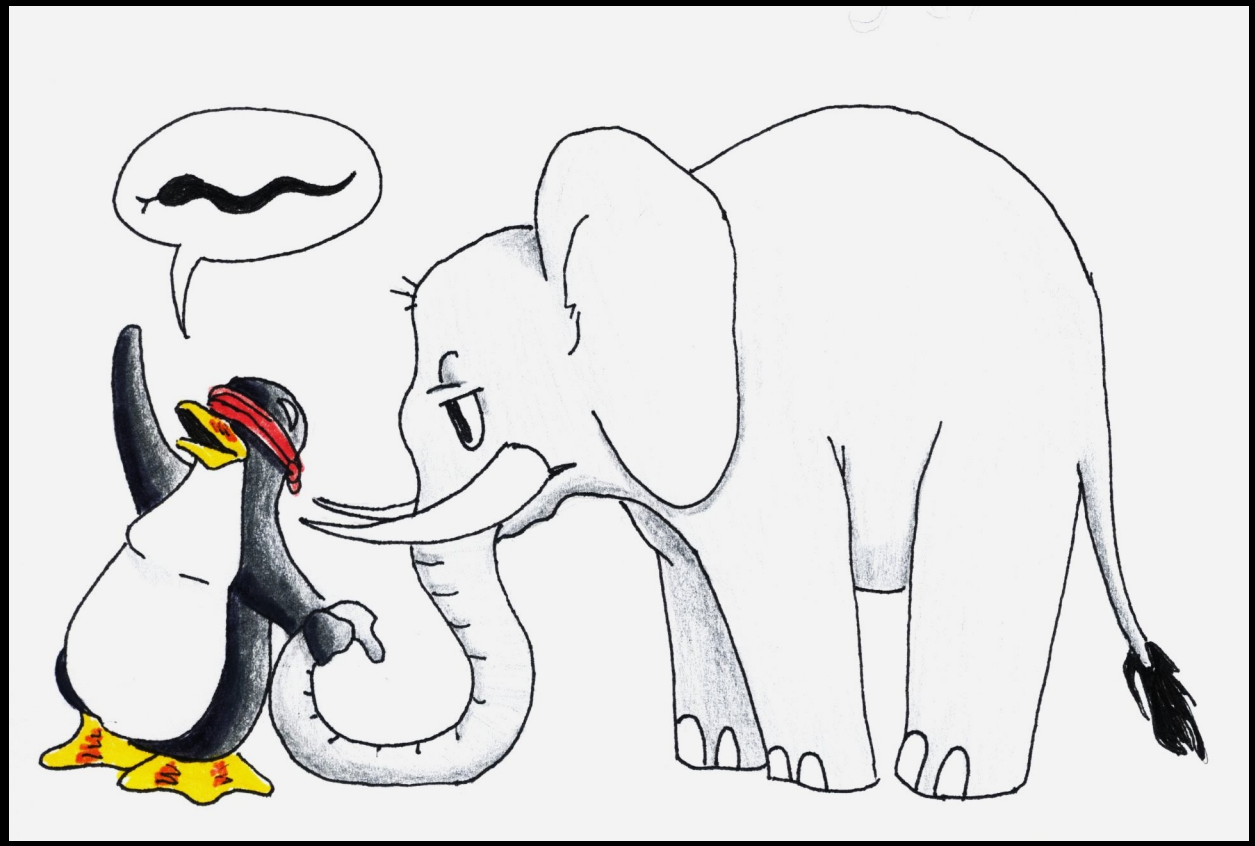


© 2009 IBM Corporation

The Parable of the Six Blind Penguins and the Elephant

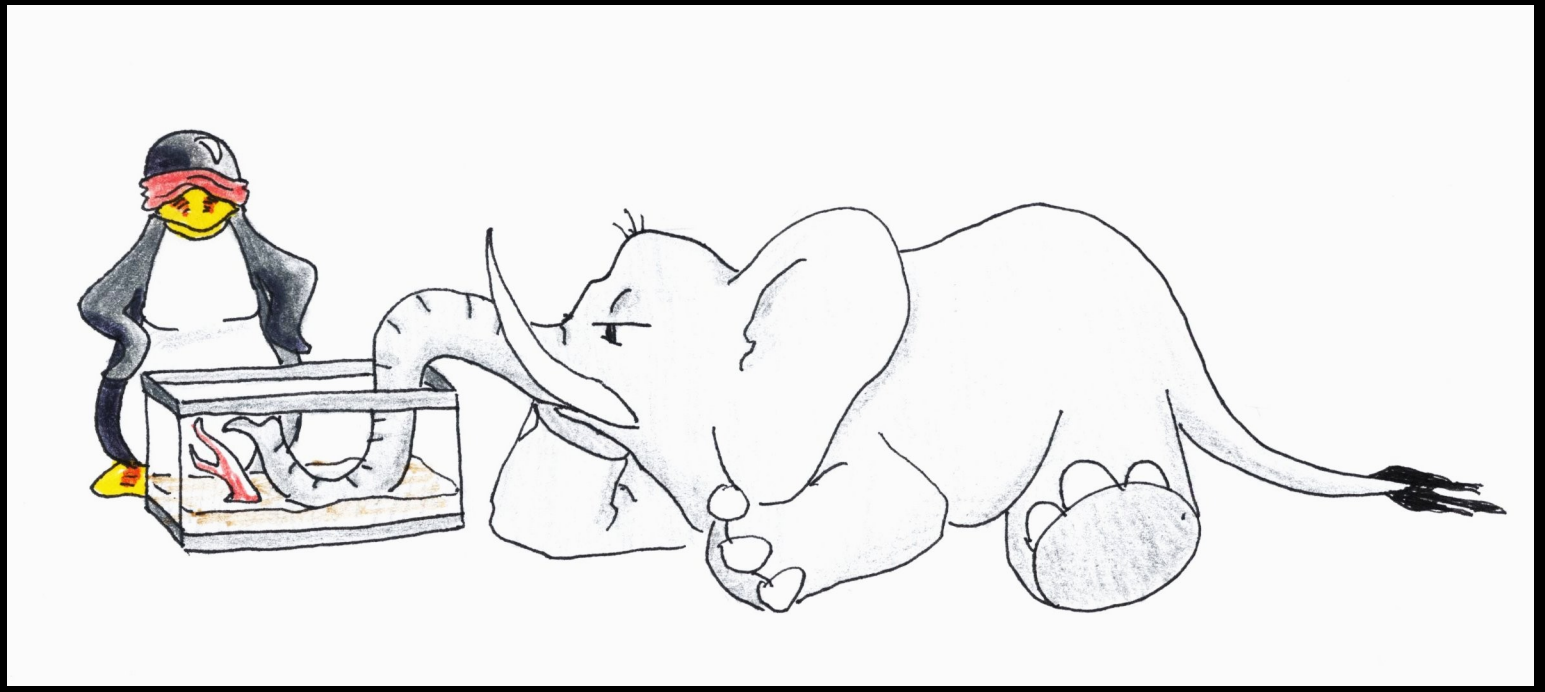
© 2009 IBM Corporation

Proprietary Programming: Requirements



© 2009 IBM Corporation

Proprietary Programming: “Solution”



But sooner or later...

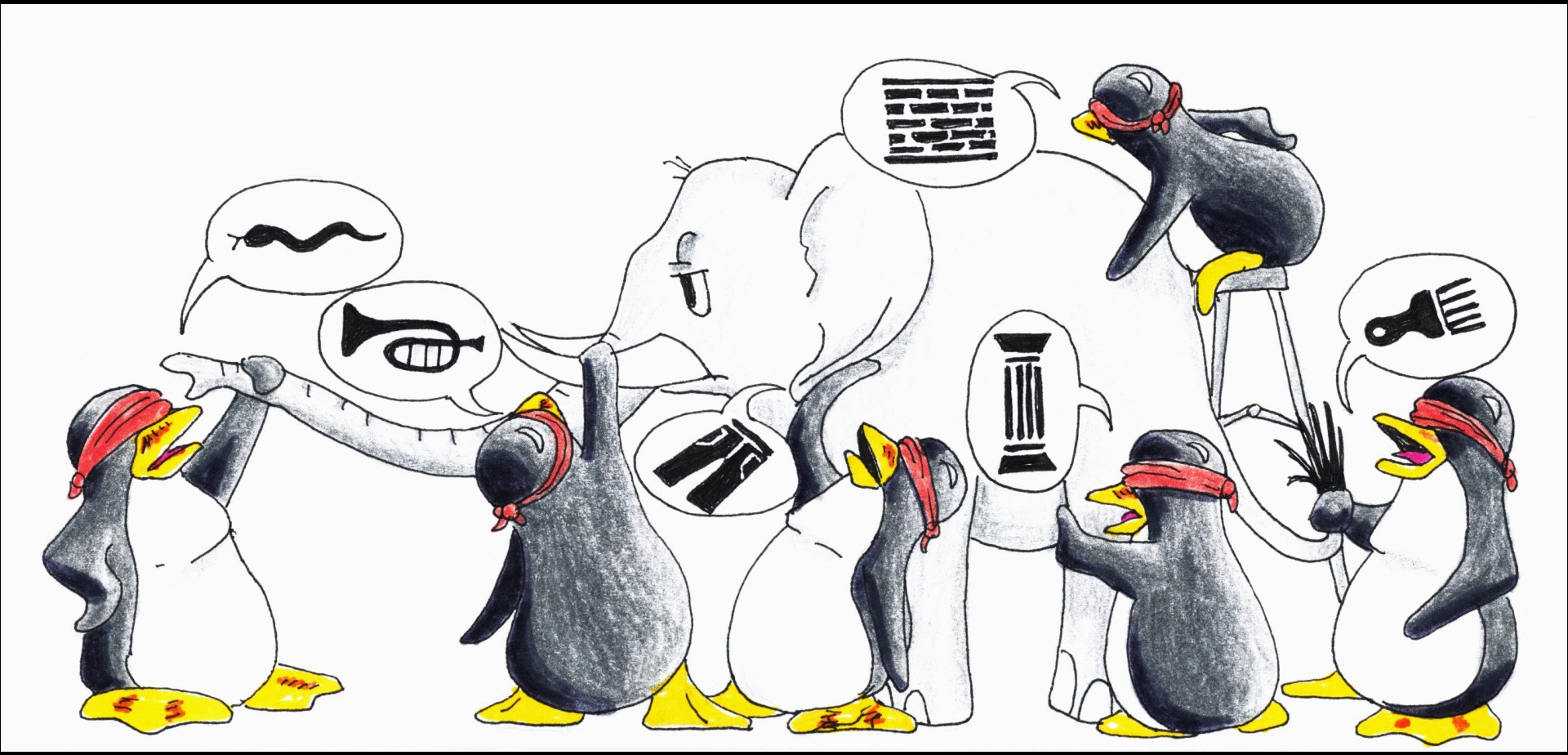
© 2009 IBM Corporation

The Rest of the Elephant Will Make Itself Known...



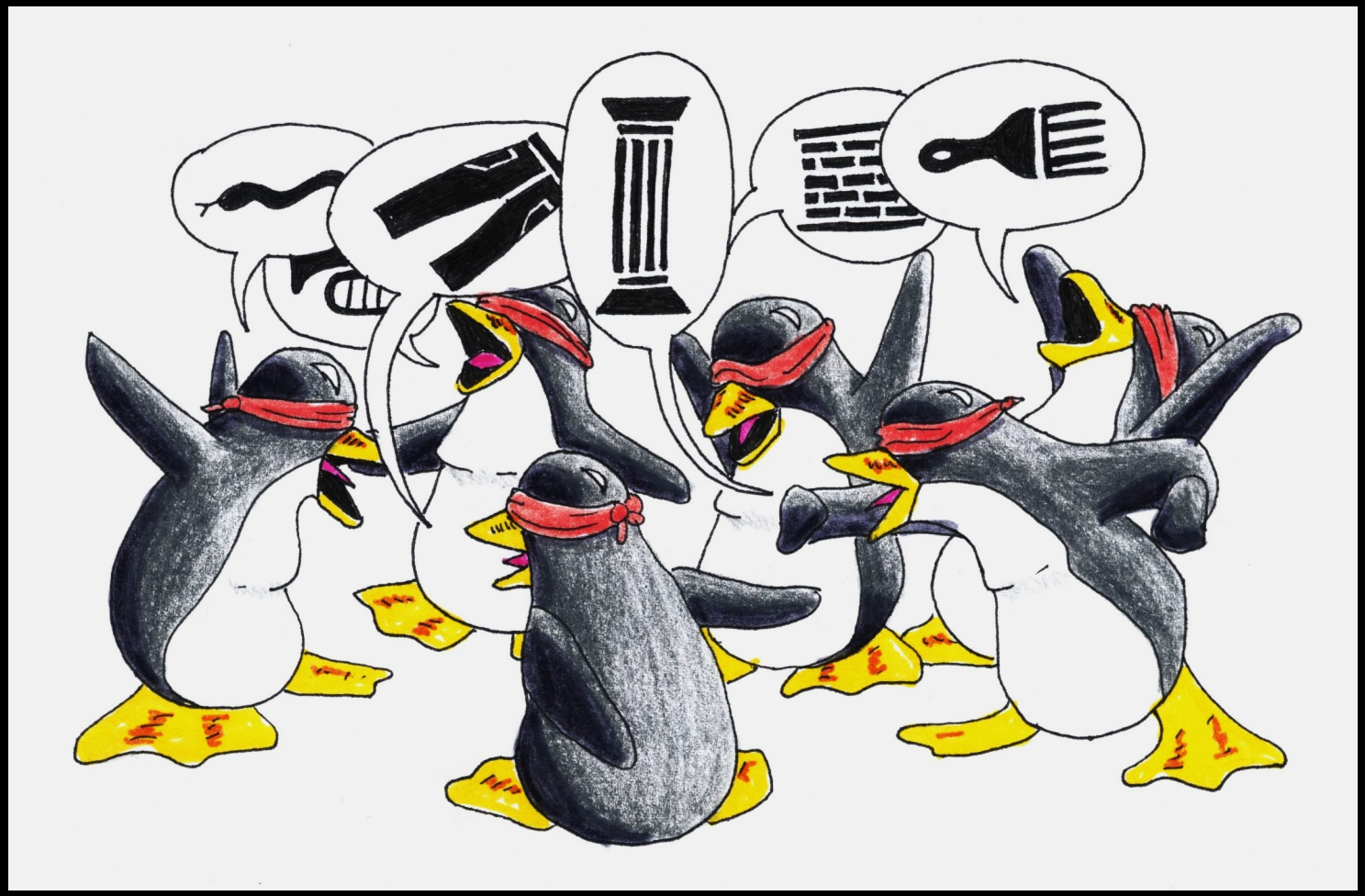
© 2009 IBM Corporation

FOSS Programming: Requirements



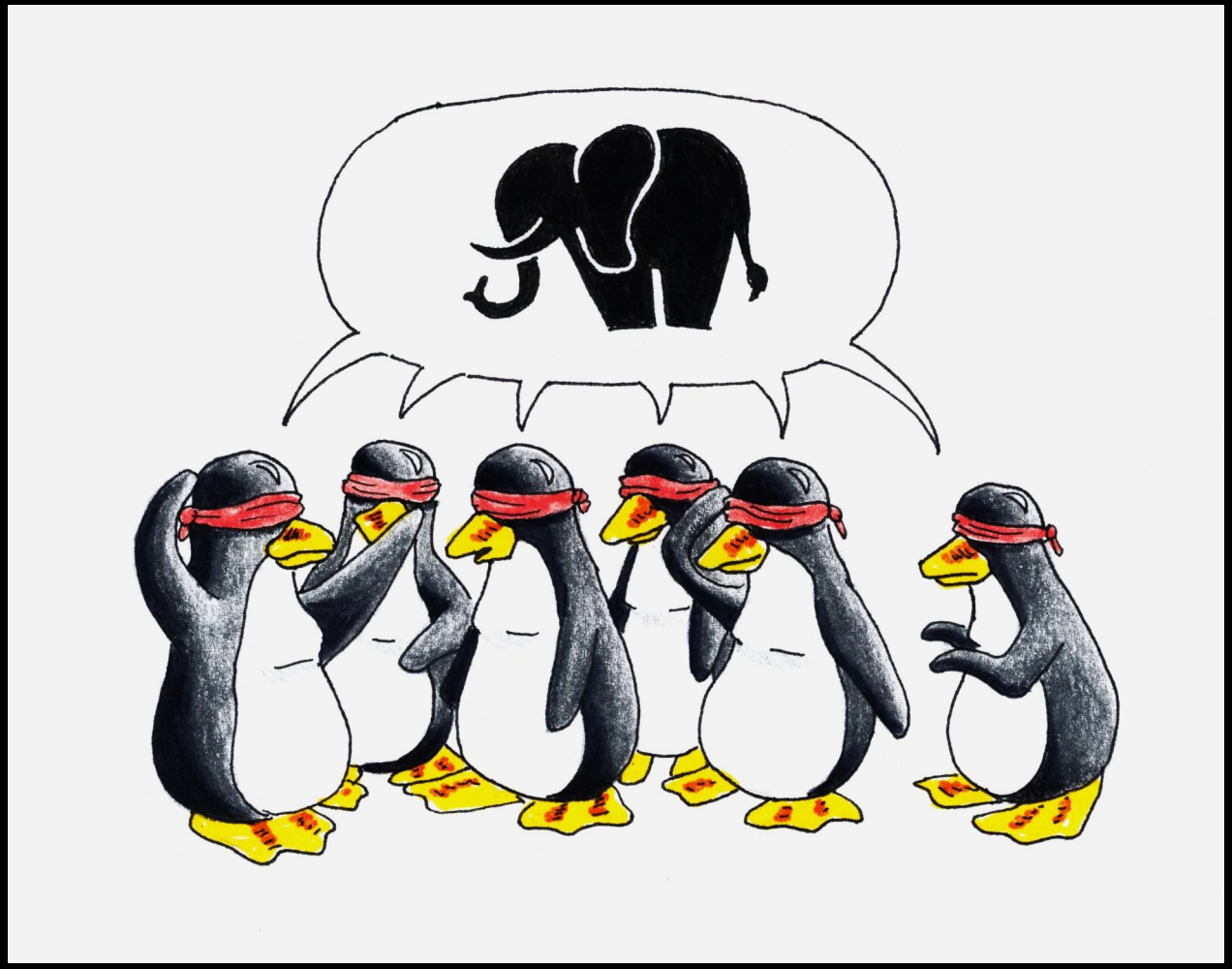
© 2009 IBM Corporation

Just Another Day on LKML...



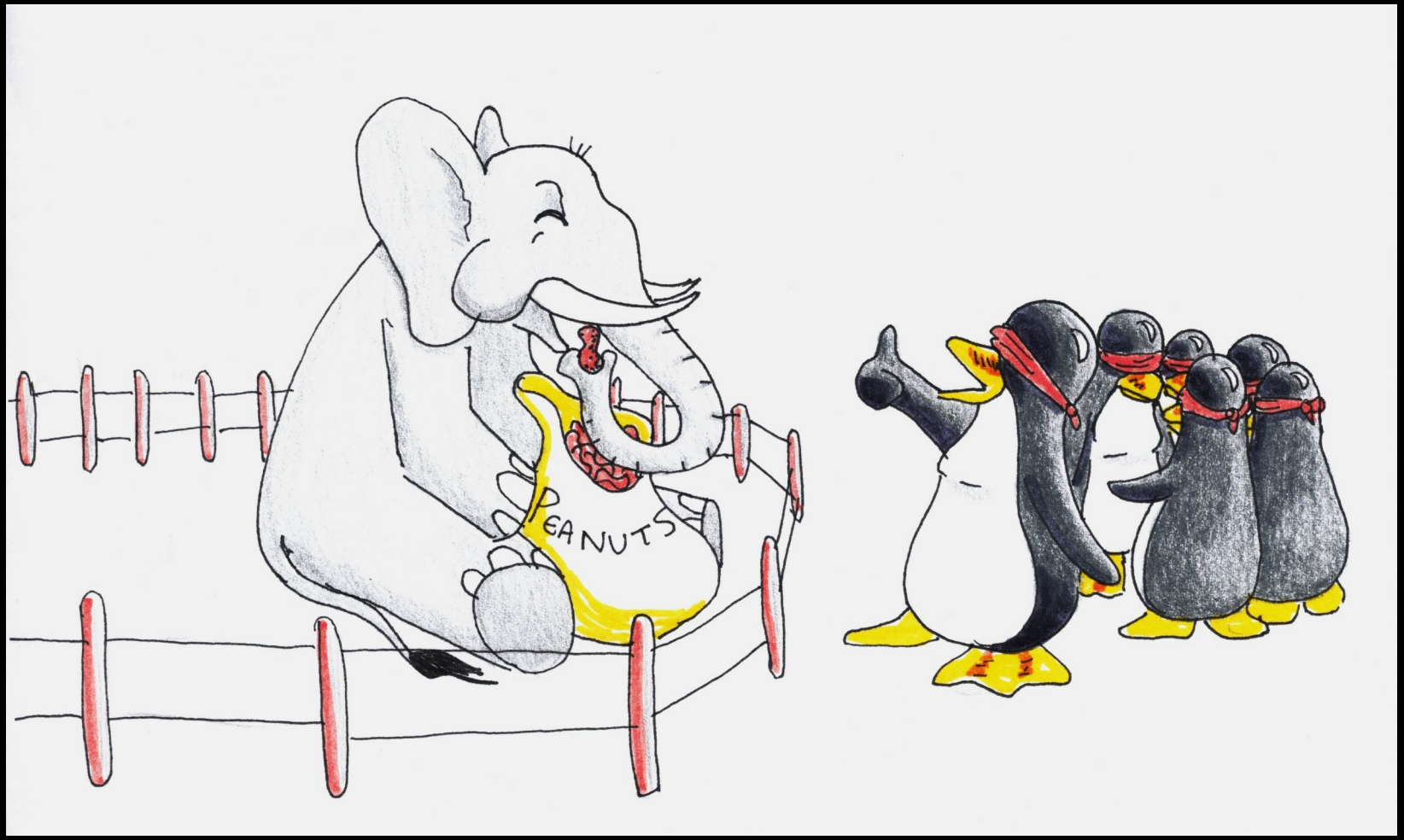
© 2009 IBM Corporation

But Sometimes Consensus is Achieved



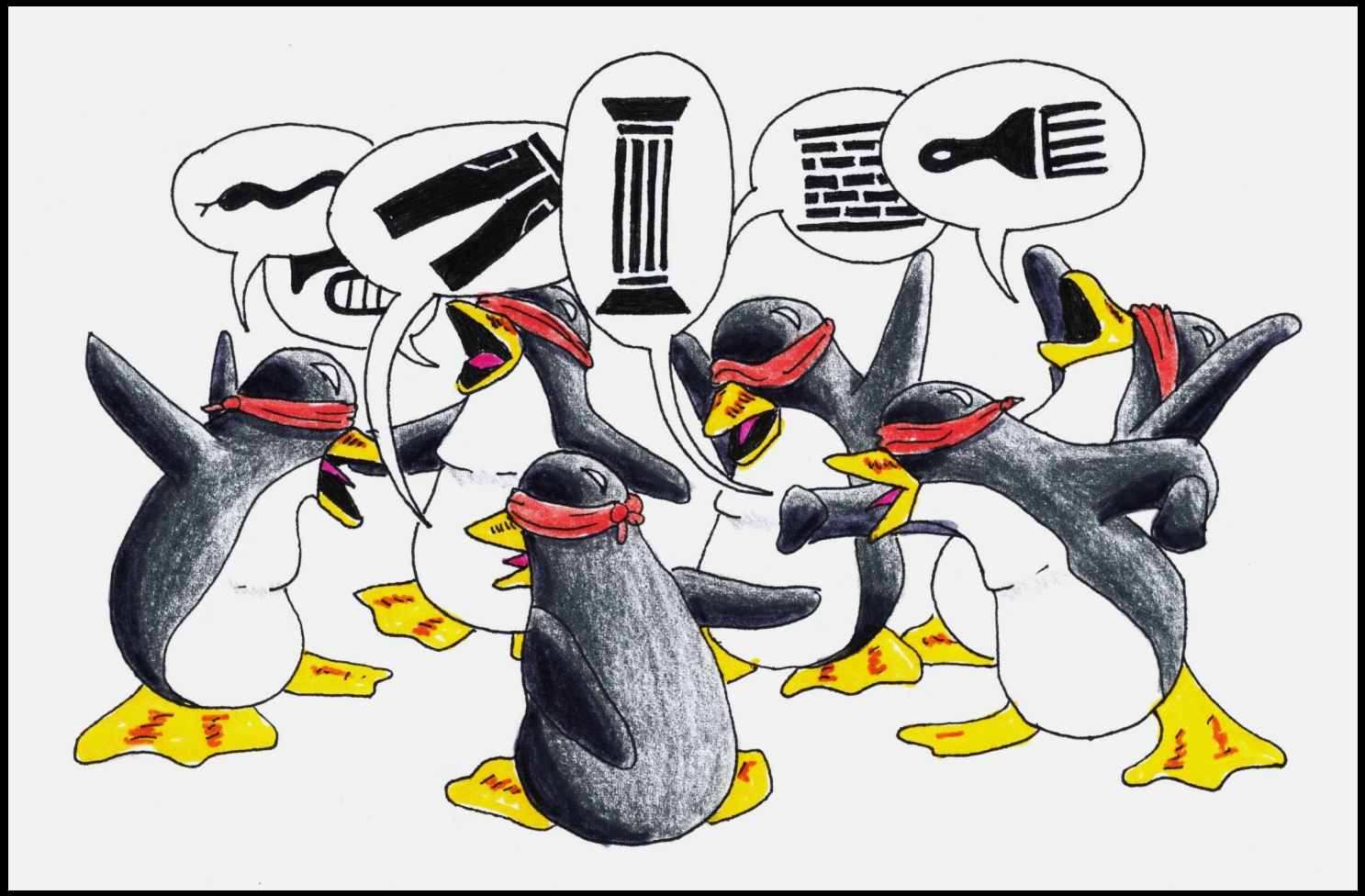
© 2009 IBM Corporation

And an Appropriate Solution Produced Thereby



© 2009 IBM Corporation

But the Android Discussions Got Stuck Right Here for A Few Years:



© 2009 IBM Corporation

More Recently, There Has Been Great Progress

- Rafael Wysocki: pm_qos-based wakelocks
- John Stultz: Alarm Timers & Ashmem upstreaming
- Marek Szyprowski: memory allocation for device use
- Tim Bird: Logger upstreaming, Android upstreaming summits
- Anton Vonrontsov: Low-memory killer & interactive cpufreq governor
- For more information:
  - https://lwn.net/Articles/484196/

© 2009 IBM Corporation

More Recently, There Has Been Great Progress

- Rafael Wysocki: pm_qos-based wakelocks
- John Stultz: Alarm Timers & Ashmem upstreaming
- Marek Szyprowski: memory allocation for device use
- Tim Bird: Logger upstreaming, Android upstreaming summits
- Anton Vonrontsov: Low-memory killer & interactive cpufreq governor
- For more information:
  - https://lwn.net/Articles/484196/
- But it would have been good to have started sooner...

© 2009 IBM Corporation

Table of contents

- Choose Your Flamewar
- Know Your Goals
- Do Your Homework
- Analyze The Flamewar
- Lessons Learned

© 2009 IBM Corporation

Choose Your Flamewar

© 2009 IBM Corporation

Flamewar Tradeoffs

- Shorter flamewars require more expertise on your part
  - The shorter the flamewar, the less time you have to study the topic
  - In contrast, you might calm a years-long flamewar simply by writing the combatants' positions in a neutral tone
  - The Android suspend-blocker flamewar was almost 2 years old
    - My several weeks of study were negligible by comparison
- The more people talk past each other, the more help needed
  - But don't expect them to welcome such help
  - Or to thank you for it

    “People really need help but may attack you if you do help them.”

    “Help people anyway.”
  - A thick skin: Don't leave home without it!

http://www.paradoxicalcommandments.com/

© 2009 IBM Corporation

Know Your Goals

© 2009 IBM Corporation

Why Would You Need A Goal Beyond Having Fun?

- If all you are doing is having fun, you will be just another flamer

© 2009 IBM Corporation

Why Would You Need A Goal Beyond Having Fun?

- If all you are doing is having fun, you will be just another flamer
- Which might be OK, but not the topic for this talk

© 2009 IBM Corporation

What Goals Might You Have?

- Learn more about the inflamed topic
- Understand the positions of the various flamers
- Present a neutral view of one of the positions
- Advocate for one of the positions
- Present a neutral view of all of the positions
- Present a neutral view of all of the positions along with a critique of each
- Propose a solution designed to meet all participants' needs

© 2009 IBM Corporation

What Goals Might You Have?

- Learn more about the inflamed topic
- Understand the positions of the various flamers
- Present a neutral view of one of the positions
- Advocate for one of the positions
- Present a neutral view of all of the positions
- Present a neutral view of all of the positions along with a critique of each
- Propose a solution designed to meet all participants' needs

© 2009 IBM Corporation

Do Your Homework

© 2009 IBM Corporation

Types Of Homework

- Review textbooks (if any)
- Read datasheets and relevant technical web sites
  - It is amazing how many flamewars are missing a few hard facts
  - But be aware of any axe-grinding that might be taking place
- Read the flamewar messages
  - You will need ~10x the writing time!
    - Extracting technical content from screaming and shouting is not always easy...
  - Keep track of unstated assumptions and consequences
  - Keep a written log of technical content, including URLs if possible
  - Avoid taking sides
    - This is harder than it looks

© 2009 IBM Corporation

“Whacking Droids” Homework

- Review textbooks (if any)
  - I couldn't find any, but several of the Linaro folks gave me a tutorial
- Read datasheets and relevant technical web sites
  - The LWN writeups were extremely helpful (see next slide)
- Read the flamewar messages
  - Accumulated almost 1,600 lines of commentary
  - Not counting the actual requirements themselves

© 2009 IBM Corporation

“Whacking Droids” Homework: LWN Articles &c

- Android Wakelock Java-Level Documentation
  - http://developer.android.com/reference/android/os/PowerManager.WakeLock.html
- Wakelocks and the embedded problem (LWN)
  - February 10, 2009: http://lwn.net/Articles/318611/
- From wakelocks to a real solution (LWN)
  - February 18, 2009: http://lwn.net/Articles/319860/
- Suspend block (LWN)
  - April 28, 2010: http://lwn.net/Articles/385103/
- Blocking suspend blockers (LWN)
  - May 18, 2010: http://lwn.net/Articles/388131/
- Suspend blocker suspense (LWN)
  - May 26, 2010: http://lwn.net/Articles/389407/
- What comes after suspend blockers (LWN)
  - June 1, 2010: http://lwn.net/Articles/390369/
- A suspend blockers post-mortem (LWN)
  - June 2, 2010: http://lwn.net/Articles/390392/

© 2009 IBM Corporation

“Whacking Droids” Homework: Relevant Experience RCU and Energy Efficiency

- State in 2003: RCU wakes up each CPU at least once per grace period to check for quiescent states
- Wastes power, so dyntick-idle API was added
- This proved buggy (failed to spot RCU read-side critical sections in interrupts from dyntick idle)
  - Buggy, but extremely reliable...
  - No one noticed for 5 years...
- Reliable dyntick idle in preemptible RCU in 2007
  - Appeared in TREE_RCU in 2008
- At this point, RCU leaves sleeping CPUs lie

- So RCU is perfectly energy efficient, right?

© 2009 IBM Corporation

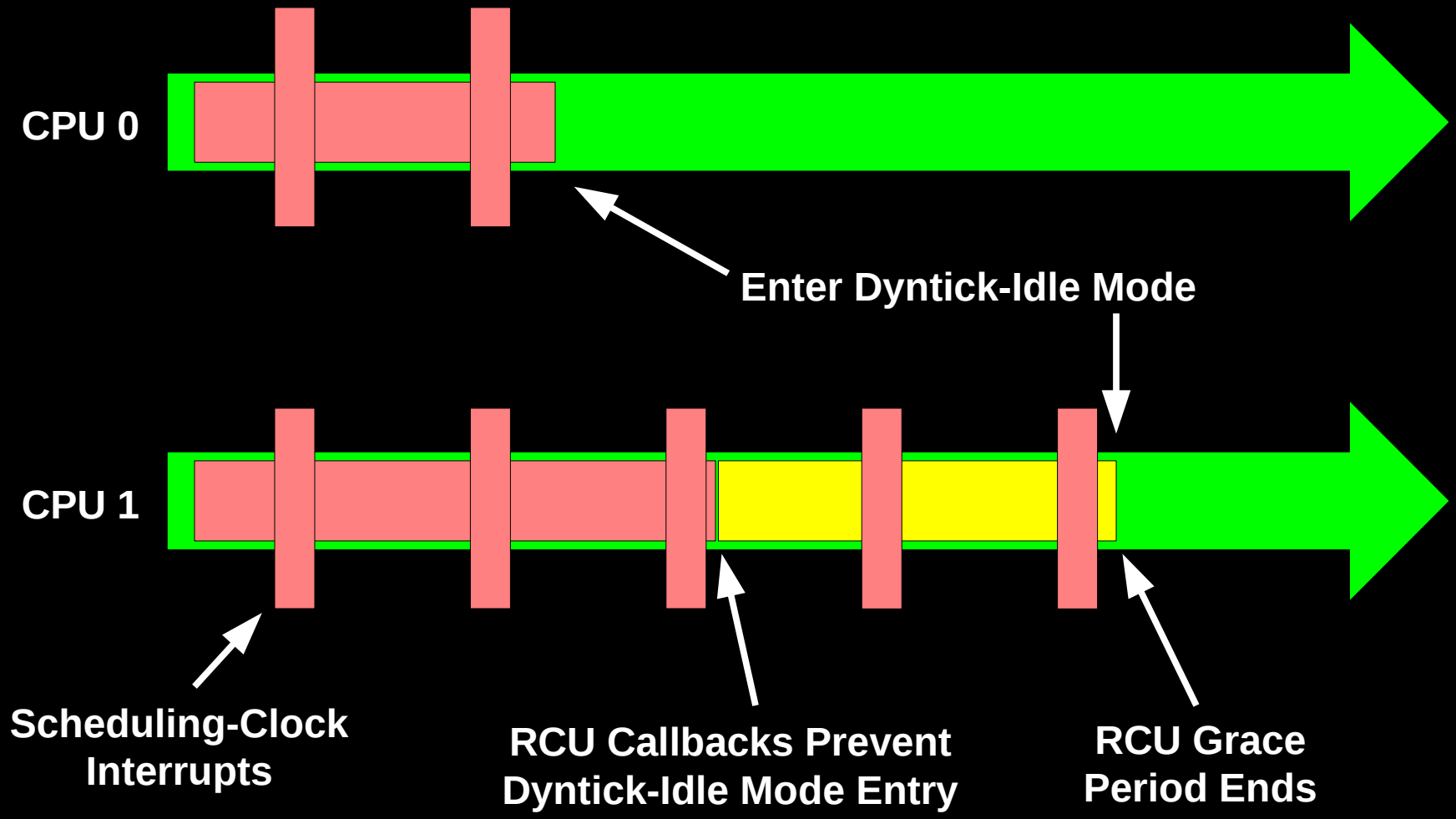
"Whacking Droids" Homework: Relevant Experience RCU and Energy Efficiency

- Well, I thought that RCU was energy efficient
- Then I got a call from someone with a battery-powered multicore system
- And he was ***very*** upset with RCU

- Why?

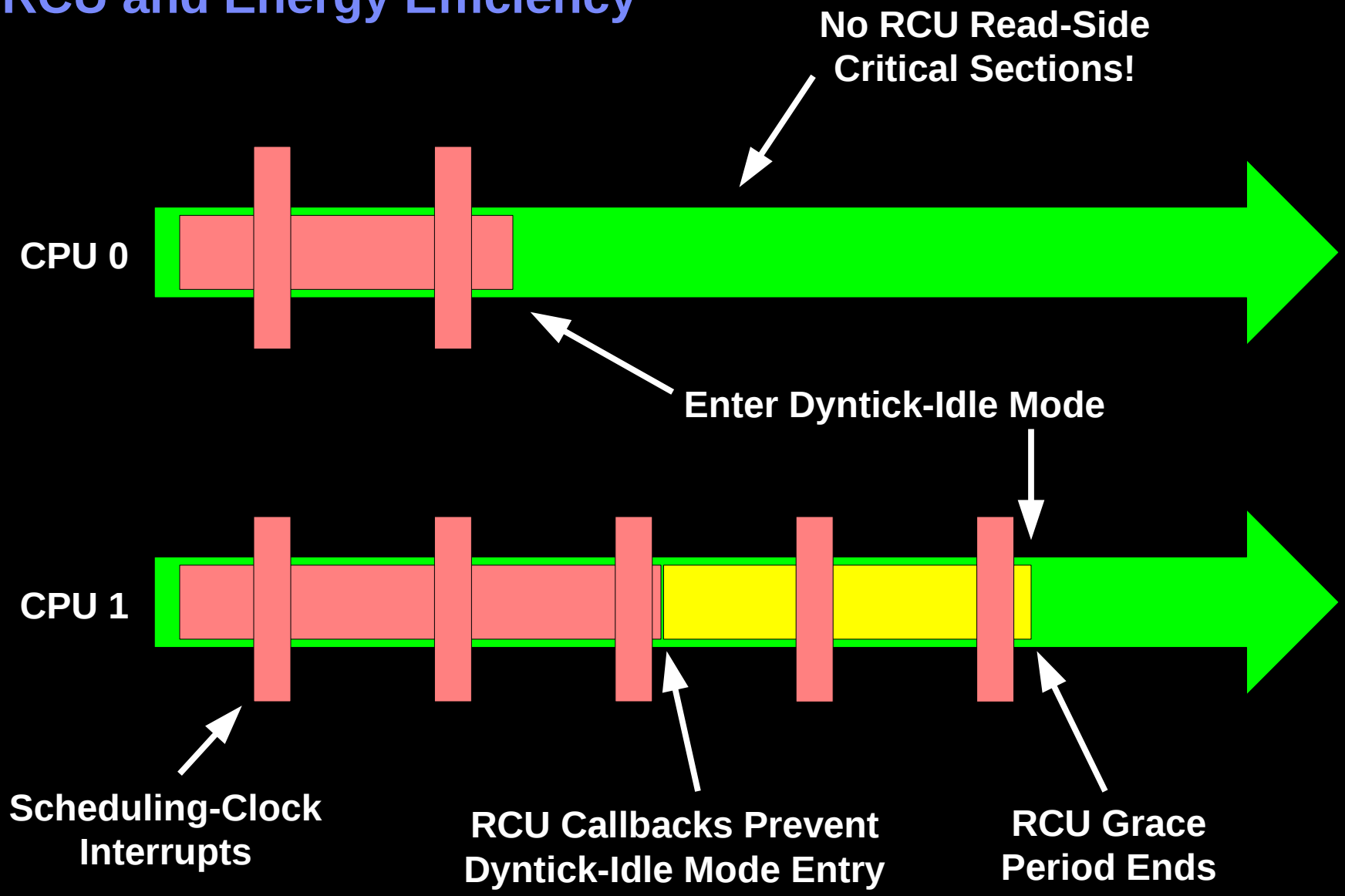
© 2009 IBM Corporation

RCU and Energy Efficiency



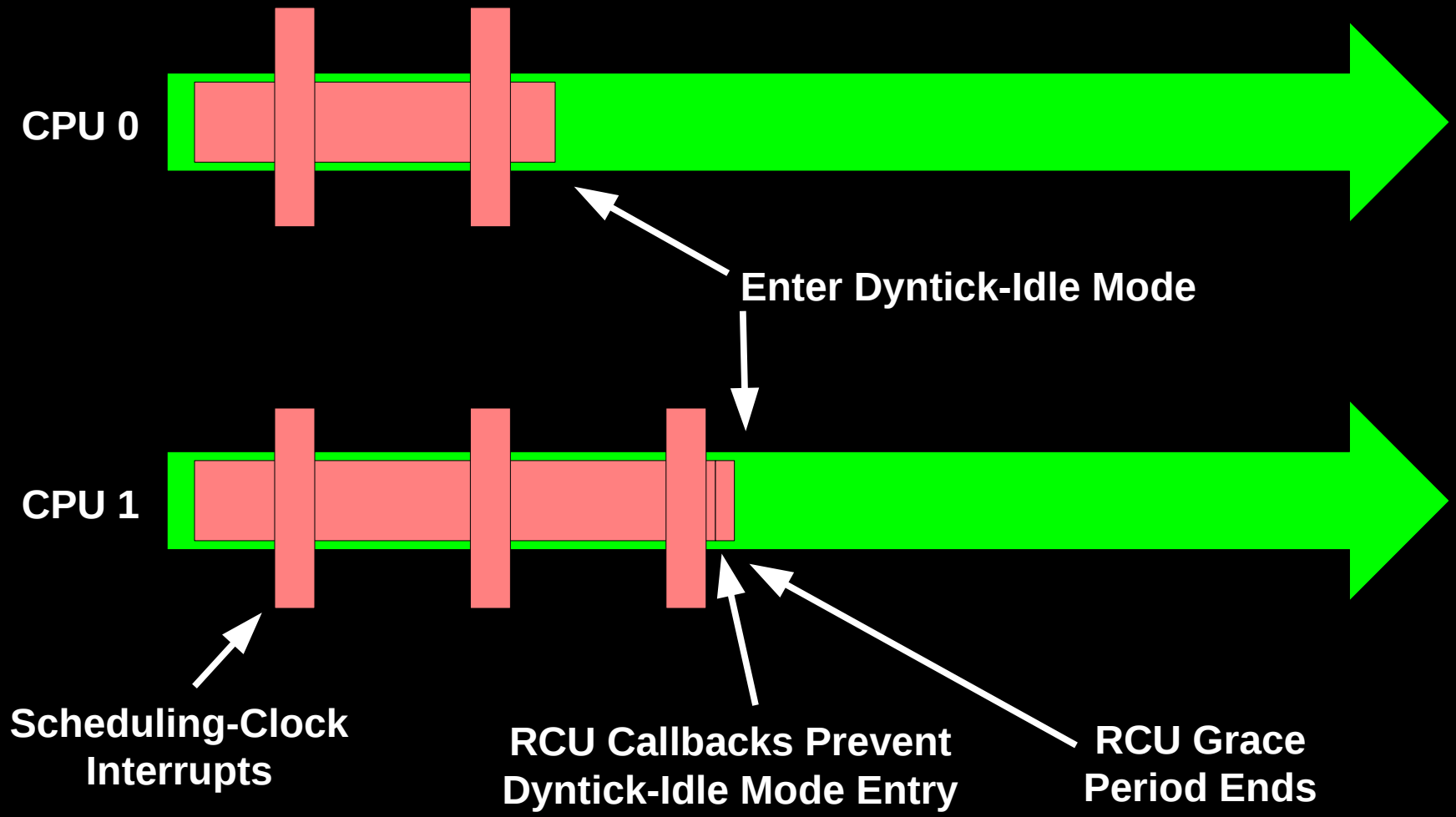
© 2009 IBM Corporation

RCU and Energy Efficiency



© 2009 IBM Corporation

RCU and Energy Efficiency: Desired State: CONFIG_RCU_FAST_NO_HZ



© 2009 IBM Corporation

RCU and Energy Efficiency: Servers vs. Embedded

Servers might have energy efficiency as a hard requirement, but...

© 2009 IBM Corporation

RCU and Energy Efficiency: Servers vs. Embedded

Of course, the situation is reversed for scalability...

© 2009 IBM Corporation

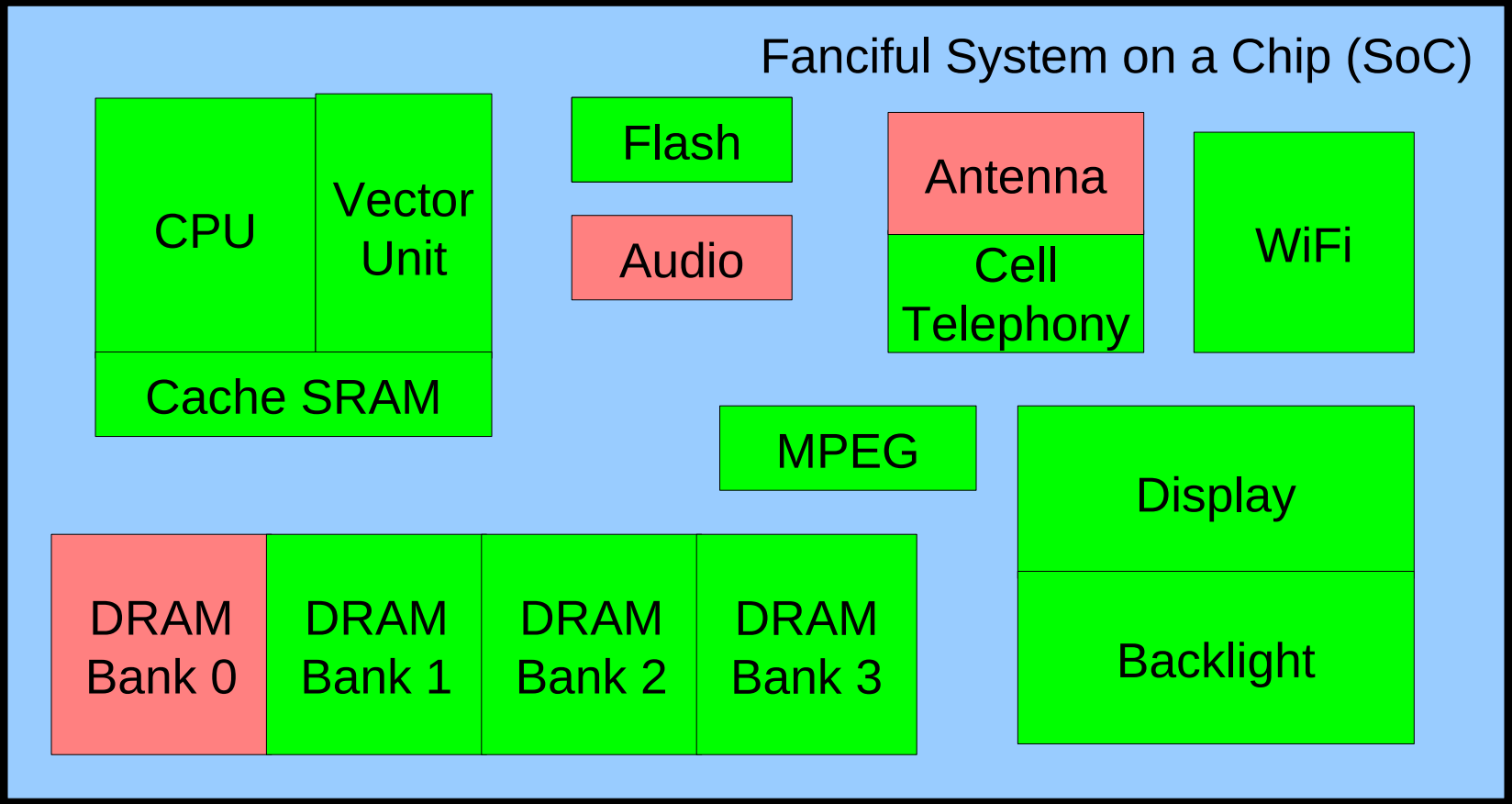
"Whacking Droids" Homework: Datasheets

- Server power consumption: 100 watts/socket
- Low-power server: 10 watts/socket
- Extreme low-power server: 2 watts/socket
- High-power embedded: 500 milliwatts
- Ditto in low-power mode: 20 milliwatts
- Battery-life power limit: 500 microwatts

How can this *possibly* work???

© 2009 IBM Corporation

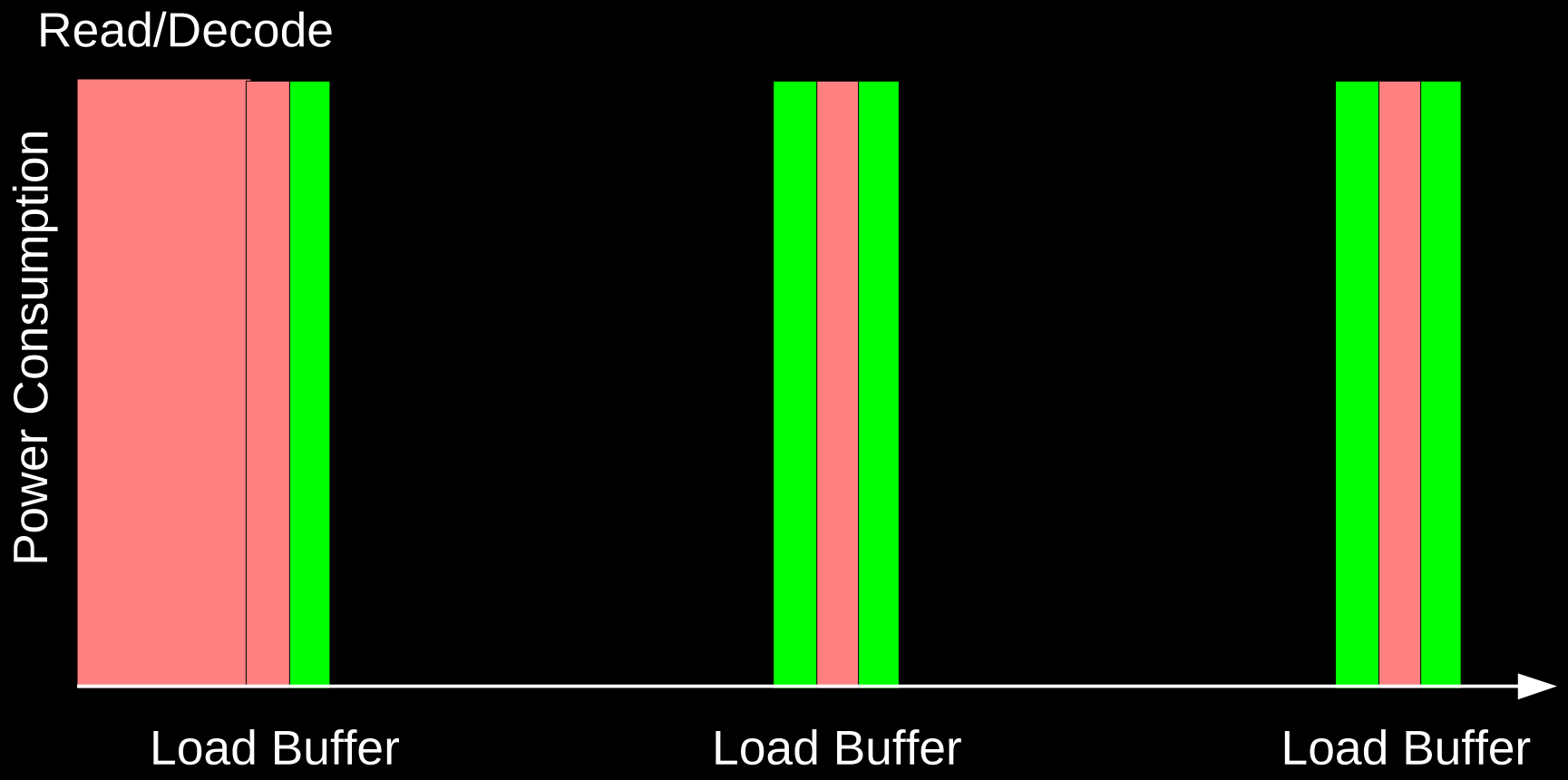
“Whacking Droids” Homework: Datasheets
Audio Playback Application



Much of the time, most of the chip powered down.
Must periodically power up CPU and flash to decode more sound.

© 2009 IBM Corporation

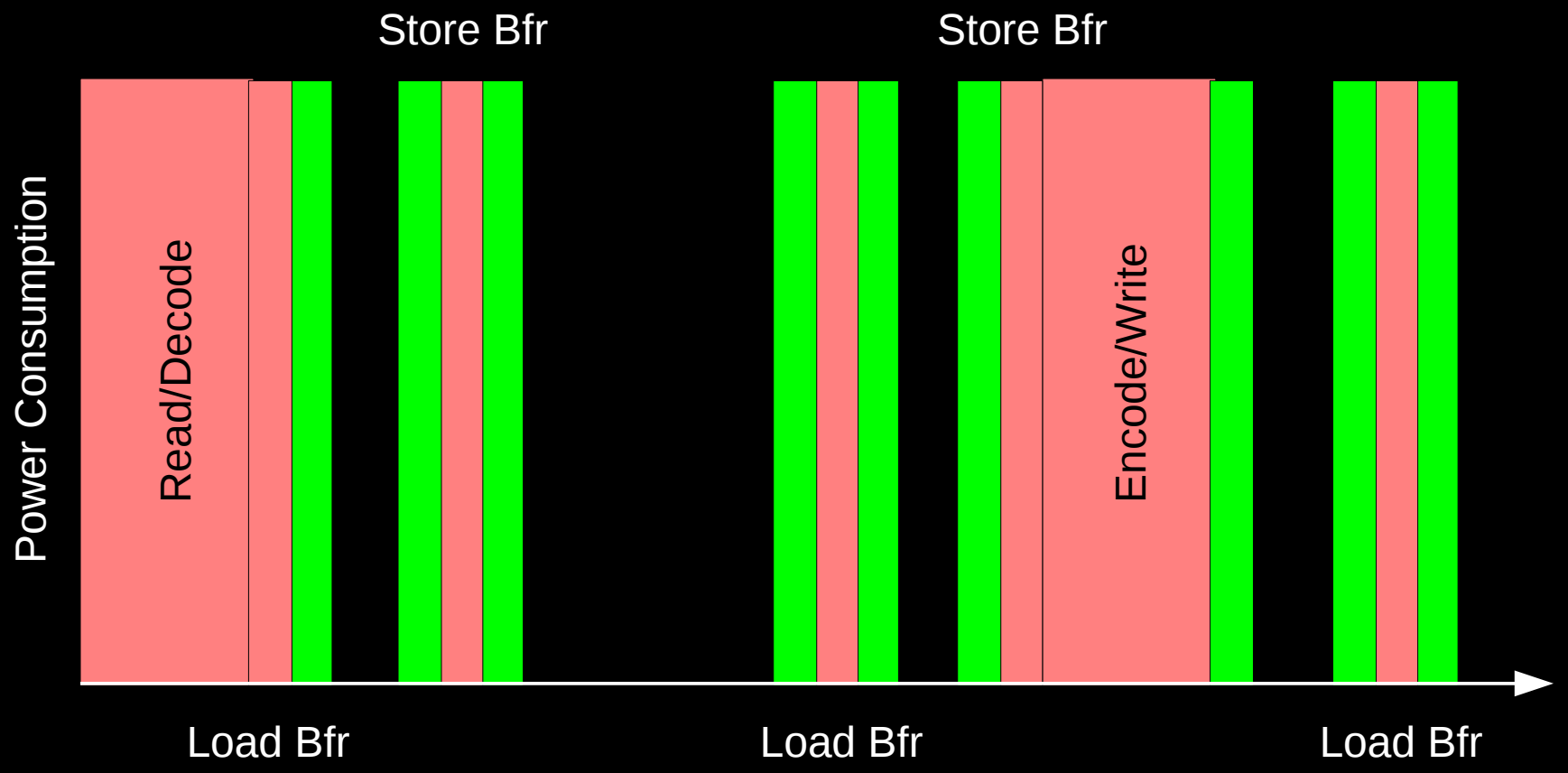
“Whacking Droids” Homework: Datasheets
Audio Playback Application: Power Consumption Profile



Backlight and display assumed to be powered off.
What do the green bars signify?

© 2009 IBM Corporation

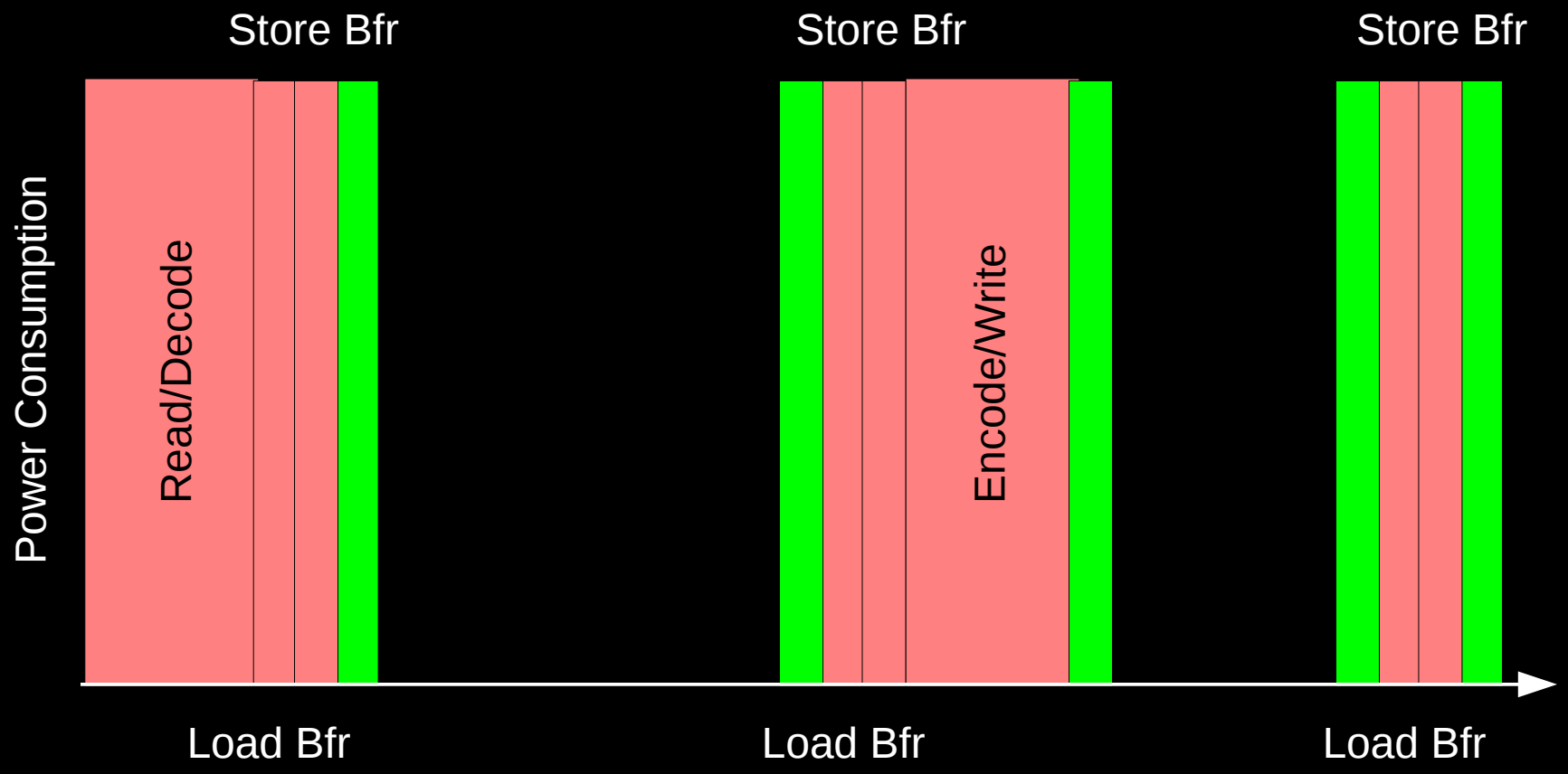
“Whacking Droids” Homework: Datasheets
Audio Playback Plus Recording: Power Consumption Profile



Combining Two Power-Efficient Applications Results in Power Inefficiency!!!

© 2009 IBM Corporation

“Whacking Droids” Homework: Datasheets Audio Playback Plus Recording: Energy-Efficient Operation



Power efficiency is a system attribute, not just an application attribute

© 2009 IBM Corporation

"Whacking Droids" Homework: Summary

- In some ways, power efficiency is more challenging than is concurrency
  - Power efficiency is a system attribute
  - Power efficiency thus less amenable to localized approaches
- I learned a huge amount while doing the homework
  - I expect that I have a huge amount more to learn
  - And I suspect that many other server-centric hackers also have much to learn as well
- But this does not necessarily prove the Android guys correct

© 2009 IBM Corporation

Analyze The Flamewar

© 2009 IBM Corporation

Example Flames For Your Analysis

I don't think x86 is relevant anyway, it doesn't suspend/resume anywhere near fast enough for this to be usable.

My laptop still takes several seconds to suspend (Lenovo T500), and resume (aside from some userspace bustage) takes the same amount of time. That is quick enough for manual suspend, but I'd hate it to try and auto-suspend.

© 2009 IBM Corporation

Example Flames For Your Analysis

I don't think x86 is relevant anyway, it doesn't suspend/resume anywhere near fast enough for this to be usable.

My laptop still takes several seconds to suspend (Lenovo T500), and resume (aside from some userspace bustage) takes the same amount of time. That is quick enough for manual suspend, but I'd hate it to try and auto-suspend.

Hidden assumption: opportunistic suspend is to have same function on laptops, desktops and servers as on Android smartphones.

© 2009 IBM Corporation

Example Flames For Your Analysis

There is no other solution (beisdes suspend blockers).

© 2009 IBM Corporation

Example Flames For Your Analysis

```
There is no other solution (beisdes suspend
blockers).
```

Hidden assumption: “I have not thought of anything else besides suspend blockers, so there must not be anything else.”

© 2009 IBM Corporation

Example Flames For Your Analysis

Solutions to the problem already exist.

© 2009 IBM Corporation

Example Flames For Your Analysis

```
Solutions to the problem already exist.
```

Hidden assumption: the guy posting this believes he understands the problem that the Android guys are trying to solve. Maybe he does, maybe he doesn't.

© 2009 IBM Corporation

Example Flames For Your Analysis

I still don't see how blocking applications will cause missed wakeups in anything but a buggy application at worst, and even those will eventually get the event when they unblock.

© 2009 IBM Corporation

Example Flames For Your Analysis

```
I still don't see how blocking applications will
cause missed wakeups in anything but a buggy
application at worst, and even those will eventually
get the event when they unblock.
```

Hidden assumptions: (1) the poster's idea of “buggy” coincides with that of the Android community and (2) it is OK for the event to be delayed until the application finally unblocks.

© 2009 IBM Corporation

Example Flames For Your Analysis

The whole concept sucks, as it does not solve anything. Burning power now or in 100ms is the same thing power consumption wise.

© 2009 IBM Corporation

Example Flames For Your Analysis

```
The whole concept sucks, as it does not solve
anything. Burning power now or in 100ms is the same
thing power consumption wise.
```

Hidden assumption: (1) An AC power adapter won't enter the picture in the next 100ms. (2) Power cannot be deferred for seconds, minutes, hours, or days – only for about 100ms.

© 2009 IBM Corporation

Example Flames For Your Analysis

```
> On some platforms (like those with ACPI), deeper
> power-savings are available by using forced
> suspend than by using idle.

Sounds like something that's fixable, doesn't it?
```

© 2009 IBM Corporation

Example Flames For Your Analysis

```
> On some platforms (like those with ACPI), deeper
> power-savings are available by using forced
> suspend than by using idle.

Sounds like something that's fixable, doesn't it?
```

Hidden assumptions: (1) if a hardware platform can achieve lower power dissipation in suspend than in idle, this constitutes a hardware bug (2) apps should be totally responsible for their energy efficiency.

© 2009 IBM Corporation

Example Flames For Your Analysis

> How long do you wait for applications to respond
> that they've stopped drawing? What if the
> application is heavily in swap at the time?

Very realistic scenario on a mobile phone.

© 2009 IBM Corporation

Example Flames For Your Analysis

```
> How long do you wait for applications to respond
> that they've stopped drawing? What if the
> application is heavily in swap at the time?

Very realistic scenario on a mobile phone.
```

Hidden assumption: opportunistic suspend is useful only on Android smartphones.

© 2009 IBM Corporation

Example Flames For Your Analysis

> So either we get the packet before suspend, and we
> cancel the suspend, or we get it after and we wake
> up. What's the problem, and how does that need
> suspend blockers?

In order to cancel the suspend we need to keep track of whether userspace has consumed the event and reacted appropriately. Since we can't do this with the process scheduler, we end up with something that looks like suspend blockers.

© 2009 IBM Corporation

Example Flames For Your Analysis

```
> So either we get the packet before suspend, and we
> cancel the suspend, or we get it after and we wake
> up. What's the problem, and how does that need
> suspend blockers?

In order to cancel the suspend we need to keep track of
whether userspace has consumed the event and reacted
appropriately. Since we can't do this with the process
scheduler, we end up with something that looks like
suspend blockers.
```

Hidden assumption: all the non-Android people understood the interaction between suspend blockers and I/O. (I certainly did not!)

© 2009 IBM Corporation

Example Flames For Your Analysis

Nobody besides Android needs suspend blockers.

© 2009 IBM Corporation

Example Flames For Your Analysis

```
Nobody besides Android needs suspend blockers.
```

An earlier hidden assumption is finally made explicit. And if suspend blockers can help non-Android platforms, this guy posting won't be the one to figure it out, now will he?

© 2009 IBM Corporation

Lessons Learned

© 2009 IBM Corporation

General Lessons Learned

- It is not enough to listen to what people say
  - You must strive to understand what they are thinking
  - So you must understand all aspects of the technology in play
- If understanding all aspects was easy...
  - There probably wouldn't be any flaming to begin with
- Those strongly opposed to something won't think about it
  - And therefore are unlikely to come up with a use case
  - And furthermore might not be happy with use cases from others
- Those wedded to a solution won't think about other solutions
  - And therefore are unlikely to come up with alternative solutions
  - And furthermore might not be happy with solutions from others

© 2009 IBM Corporation

Android Lessons Learned

- Reasonable Android wakelock requirements exist
  - https://wiki.linaro.org/WorkingGroup/KernelConsolidation/Projects/AndroidSuspendBlockers
  - This lists 19 requirements, 2 nice-to-haves, and 3 non-requirements
- Wakelocks are likely to have server/desktop uses
  - Especially given alarm-timers interfaces
    - See http://lwn.net/Articles/429925/ for more information
  - Of course, it remains to be seen whether this use case is compelling to the Linux kernel community at large
  - Or whether solutions developed along these lines address all 19 of the above requirements
- Some pm_qos functionality is under development
  - Might not meet all requirements, but should allow merging drivers “real soon now”

© 2009 IBM Corporation

Legal Statement

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

© 2009 IBM Corporation

Questions?

© 2009 IBM Corporation