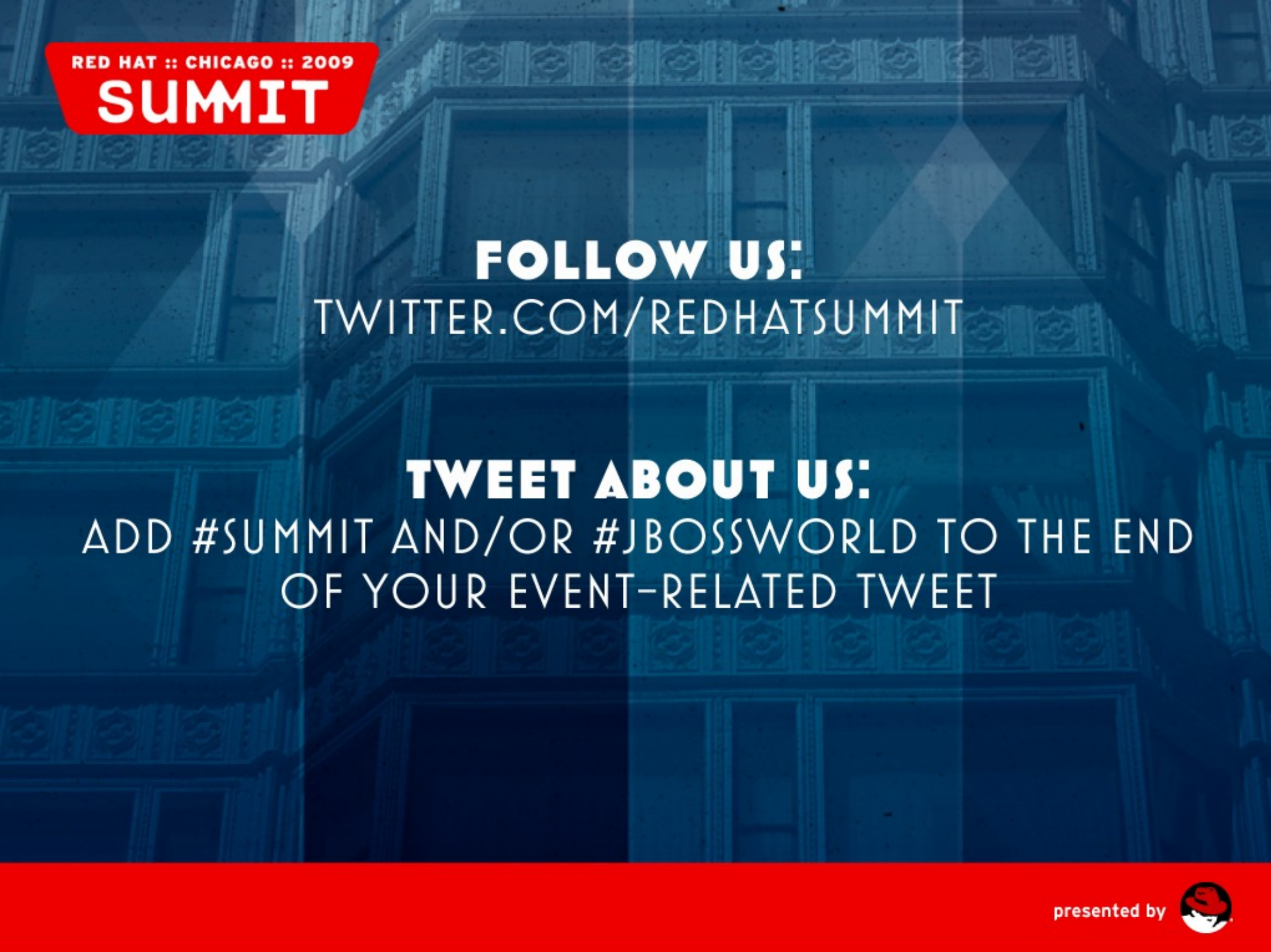
RED HAT :: CHICAGO :: 2009
SUMMIT
FOLLOW US:
TWITTER.COM/REDHATSUMMIT
TWEET ABOUT US:
ADD #SUMMIT AND/OR #JBOSSWORLD TO THE END
OF YOUR EVENT-RELATED TWEET
presented by

RED HAT :: CHICAGO :: 2009
SUMMIT

Creating A Low Cost Enterprise Linux Deployment

Andrew Hecox
Technical Account Manager
Red Hat
September 2 2009

presented by

Creating a Low Cost RHEL Deployment

- Topics for today:
  - Your organizational build
  - Collecting core files
  - System logging
  - System resource monitoring
  - Interactive performance monitoring
  - Interactive problem troubleshooting
  - Putting it all together

Your Organizational Build

- De-couple your application, operating environment, and hardware
  - PXE, kickstart, configs management, app packaging
- Sometime after a release's GA, snap a gold master
  - Selectively pull back critical fixes
- Test targeted apps early on RHEL betas
  - When possible, on each new Fedora release as well

Collecting Crashes

- A core file is a dump of a process' memory plus some metadata
- Generated when a program is in a exceptional state
- The kernel creates userspace cores
- The kernel creates kernel cores too (via kexec-kdump)
- Cores can be collected on demand...

Collecting Cores (Userspace)

- Set “core” to “unlimited” in /etc/security/limits.conf
- A core file is created in the process' cwd
- You can also collect cores manually with gcore
- You only need debuginfo on a system to read a core, not collect one

Collecting Cores (Kernel)

- Second kernel is stored in memory; on panic, the main kernel jumps to the second one to do the dump
- System memory is getting large -- it's not infrequent for memory to be larger than local disk
- You can create a stripped file: this is a sparse file with holes for uninteresting data
  - See --sparse options to cp, tar, etc.
- Recommended config is something like:
  - ext3 /dev/vg/var
  - path /crash
  - core_collector makedumpfile -c -d 31

System Logging

- Currently using a long-ago branch of sysklogd
- Very reliable, simple system and kernel logger
- By default, fsync(3) is called with each write
- Append '-' to each local file name to skip the fsync
- Keep a central syslog server (sync'd or not)
- In RHEL6 (and as a tech preview in RHEL5), rsyslog is the future

Resource Monitoring

- Done locally via sar, in the sysstat package
- You may want to change the duration files are held by changing /etc/sysconfig/sysstat
- Can use at increased sampling frequency if needed
- Also collect resource data remotely via snmp
  - But generally customers run a full-blown monitoring application

Interactive Performance Troubleshooting

- Many, many tools: pick a few to use consistently
  - My personal selection:
    - vmstat -a 2
    - iostat -xm <dev1> <dev2> <...> 10
    - top (with a custom .toprc)
- oprofile provides system-wide sampling data, include both application and function, if debuginfo is installed
  - Most helpful to the code's developer
  - Beyond CPU usage, can track events like cache misses or pipeline stalls

Interactive Troubleshooting

- strace, ltrace to watch a process's interactions
- tcpdump on the server, wireshark on your workstation for packet capture and analysis
- blktrace for I/O capture and analysis
- gdb, crash to debug your code (or ours)
- systemtap on your development boxes
- systemtap-runtime in production

Putting It All Together

- Identify a core build management team and structure
- Keep just a couple supported builds at once
  - EUS can help minimize the churn
- Use Fedora daily
- If your organization is large enough to employ a team of RHEL engineers, consider a TAM support option

QUESTIONS?

TELL US WHAT YOU THINK:
REDHAT.COM/SUMMIT-SURVEY