Item 283
git_comments:

git_commits:

1. **summary:** TS-13 problem with AS_CASE not supported in autoconf 2.59 on RHEL4
   message: TS-13 problem with AS_CASE not supported in autoconf 2.59 on RHEL4 using the expanded marco in its place git-svn-id: https://svn.apache.org/repos/asf/incubator/trafficserver/traffic/trunk@835138 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** x86_64 linux port
   description: Need to compile and run in x86_64 mode on 64-bit linux

jira_issues_comments:

1. This is an initial cut at a 64-bit linux port. It runs, good enough, to surf through and the debug build did 1300 ops/sec against 2 cooked linux disks /dev/sdb,/dev/sdd for a couple minutes. I am looking for a committer to help with code review, tracking down things I might have missed and to discuss the approach to atomics. I am using the gnuc builtin atomics __sync_XXXXX which gives portability to all GCC-based ports without requiring the assembly language that was used before.
2. Paul Querna had also started porting atomics to use the gcc builtins, and we should compare those two. One issue is that we need to support gcc compiler versions which do not support this. In particular, RHEL4 still uses gcc 3.x, and we unfortunately have to support this for the time being at least. Paul had suggested incorporating some stuff from APR, and use the builtins when possible, and APR otherwise. We should coordinate these two efforts.
3. Discussing with John for portability there is the HP research 'atomic_ops' project http://www.hpl.hp.com/research/linux/atomic_ops/ I haven't looked at it thoroughly but it supports quite a few platforms (X86, Itanium, Alpha, PA-RISC, PowerPC, and SPARC) and OSs (Linux, Microsoft Windows, HP/UX, Solaris, and MACOSX). They do indicate some ports are more complete than others. The license is mixture of MIT-style for code linked in with the client with some GPL stuff pulled into a separate library.
4. I left in the old code which will work for 32-bit machines using gcc 3.x, so the only folks who will be out will be those who for some reason are running 64-bit but don't want to update their compiler. I did break the makefile for the other platforms as I removed the .s file since I am not an automake guru. Perhaps someone could tell me how to include that conditionally? (i.e. on __i386__ platforms)? The new c++0x standard will have atomic operations built in, and when that comes out we should use that. In the meantime, the GCC ones are also supported by Intel which covers another major compiler. The HP effort is a good idea for other platforms, as I have used it before and it is of good quality. Personally, I would just use the code below which is small, and passed the test_atomic regression until the c++0x or someone wants another platform/compiler in which case I would go with the HP code. Also, I am assuming sequential(esque) consistency which is true for i386, x86_64 and Sparc, but should be conditioned for on other architectures like Power. #if defined(__GNUC__) && (__GNUC__ >= 4) && (__GNUC_MINOR__ >= 1) // see http://gcc.gnu.org/onlinedocs/gcc-4.1.2/gcc/Atomic-Builtins.html static inline ink32 ink_atomic_swap(pvink32 mem, ink32 value) { return __sync_lock_test_and_set(mem, value); } static inline ink64 ink_atomic_swap64(pvink64 mem, ink64 value) { return __sync_lock_test_and_set(mem, value); } static inline void *ink_atomic_swap_ptr(void

*mem, void *value) { return __sync_lock_test_and_set((void**)mem, value); } static inline int ink_atomic_cas(pvink32 mem, int old, int new_value) { return __sync_bool_compare_and_swap(mem, old, new_value); } static inline int ink_atomic_cas_ptr(pvvoidp mem, void* old, void* new_value) { return __sync_bool_compare_and_swap(mem, old, new_value); } static inline int ink_atomic_increment(pvink32 mem, int value) { return __sync_fetch_and_add(mem, value); } static inline int ink_atomic_increment64(pvink64 mem, ink64 value) { return __sync_fetch_and_add(mem, value); } static inline void *ink_atomic_increment_ptr(pvvoidp mem, intptr_t value) { return __sync_fetch_and_add((void**)mem, value); } static inline int ink_atomic_cas64(pvink64 mem, ink64 old, ink64 new_value) { return __sync_bool_compare_and_swap(mem, old, new_value); } // not used for Intel Processors which have sequential(esque) consistency #define INK_WRITE_MEMORY_BARRIER #define INK_MEMORY_BARRIER #else ... the old code ...

5. The comment is wrong as the Itanium has a relaxed consistency model. It should be x86 x86_64 Sparc.
6. Fixed issue with ink_queue. The freelists requires specialized code for each architecture to insert the version into the head pointer in unused bits. This code now runs under high load for extended periods of time in forward proxy mode with optimization turned on.
7. **body:** I am testing out the patch. I have only one minor fix to the patch where one of our Makefile.am files hard codes -ltcl instead of using $(LIBTCL). This broke the build on Fedora 11 core 64-bit. There looks to be an incompatibility with the librecords files and I got a core when trying to read the "32-bit" librecords files. I removed the old db files and let the 64-bit code recreate them. We might want to look into this more or just tell people that the 32-bit and 64-bit files are incompatible. I still need to test building the code as 32-bit and see if that didn't break.
   label: code-design
8. **body:** Patch that fixes a hard coded -ltcl in our Makefile.am. Breaks when trying to do a 64bit build on Fedora 11 core 64bit.
   label: code-design
9. **body:** There is a problem with this patch on i386 (32-bit) platforms. When we use things like ink_atomic_cas64(long*, long, long);, gcc can't generate the atomic, and instead generates a symbol like __sync_bool_compare_and_swap_8 This causes the link phase to fail (obviously), since the symbol is undefined. Not sure what options we have, but possibly we'll have to #ifdef around the (new) code that now uses long (i.e. ink64) and the 64-bit versions of the atomic operations. Also, I'm not sure what is more portable, but we already have code like #include <limits.h> #if (__WORDSIZE == 64) ... #endif No matter what, I think we should standardize on one way to detect 64-bit, maybe in one of our include files we create another define, e.g. #include <limits.h> #if (__WORDSIZE == 64) #define INK_64BIT_PLATFORM 1 #endif or some such (and use that new define consistently throughout the code). Thoughts? -- Leif
   label: code-design
10. the problem with __syn_bool_XX is that those instructions are not supported on i386 and the x86_64 build didn't like -mpentium. A solution is to add: +AC_CHECK_SIZEOF(int *) # test for 64-bit pointers, this permits CFLAGS=-m32 for 32-bit compiles on 64-bit systems +if test "${ac_cv_sizeof_int_p}" = "4"; then + CFLAGS="$CFLAGS -march=pentium" + CXXFLAGS="$CXXFLAGS -march=pentium" +else + CFLAGS="$CFLAGS" + CXXFLAGS="$CXXFLAGS" +fi to configure.ac The advantage of this is that it permits 32 bit compiles on native 64-bit systems by just doing configure CFLAGS=-m32 CXXFLAGS=-m32 v3 of the patch has this change and the corrected LIBTCL change. I am still looking at librecord... which db was it that had to be deleted? I don't remember this code, must have been after my time.
11. fixed 32-bit compiles
12. The only db file I remember is the 'host.db' file.
13. **body:** hmmm, so, this latest patch does work on i386, but... What if I wanted to compile with -march=core2 for example? Is that going to work too ? E.g. I'd do maybe CXXFLAGS='-march=core2' ./confgure As for the hostdb's etc, we added some stuff to make versioning of these things actually work (so, it'll self-destruct if it's an old version). Maybe also add something here for 64-bit vs 32-bit in the header?
    label: code-design
14. I meant to say, in my example case, the compiler gets invoked with -march=core2 -march=pentium which one takes precedence here ? I'm assuming -march=pentium is the "minimum" architecture to get the appropriate builtins?
15. > > -march=core2 -march=pentium Humm... good point. The original configure.ac just put -march=pentium in every time. It also hard coded the gnu c++ location to be /usr/bin/g++. I have an updated version of gcc in /usr/local/bin that is ahead in my PATH which I think it should take so I

changed this to be just g++. Similarly, we should probably check to see if an march is already specified and only if it is not add -march=pentium for 32-bit builds (as this is the minium required to get the atomics to work). That requires a substring search of CFLAGS which strains my autoconf/bash abilities. Any suggestions would be welcome. > As for the hostdb's etc, we added some stuff to make versioning of these things actually work (so, it'll self-destruct if it's an old version). Maybe also add something here for 64-bit vs 32-bit in the header. > I'm looking into this to see if I can just make them compatible. If not, then a flag in the header.

16. This might fix the HostDB problem. The HostDBInfo struct wasn't padded to a multiple of 8 bytes. diff --git a/iocore/hostdb/I_HostDBProcessor.h b/iocore/hostdb/I_HostDBProcessor.h index e1b0c36..75754d0 100644 --- a/iocore/hostdb/I_HostDBProcessor.h +++ b/iocore/hostdb/I_HostDBProcessor.h @@ -242,6 +242,7 @@ struct HostDBInfo unsigned int ip; int hostname_offset; // int srv_host_offset; + inku64 dummy_pad; } data; unsigned int srv_weight:16; Since I haven't seen the problem I haven't tested it. If soneone who has seen it could test or tell me how that would be great.
17. One option would be to have a --with-architecture or some such as a switch in configure. E.g. AC_ARG_WITH(architecture, [AC_HELP_STRING([--with-architecture=<arch>], [explicitly specify the architecture to use with various compilers])], ac_architecture=$withval) (not sure the above works, but something like that). And then use $ac_architecture if specified with the gcc compiler switches, and if not, default to some other "heuristics" like we did before (we'll want to support other -march= values for other platforms later on I'm sure). This would allow me to configure TS with something like --with-architecture=core2 and be all set :). Thoughts?
18. Attached is a patch that provides: 1) --with-architecture=<arch>, which can be used to explicitly set the architecture for -march with configure 2) Make sure that CFLAGS and CXXFLAGS don't append a -march=pentium if there already is a value for -march= in there. #1 provides a convenience feature, and #2 should make this more compatible with typical usage of configure (as above).
19. Applying the ts-64bit-v3.patch allows 64-bit compilation. Confirmed on Fedora 11/x86_64
20. reviewed patch 0003-TS-13-Add-support-to-only-add-march-pentium-if-CFLAG.patch and the patch looks good. sr=bcall
21. On a "clean" 64-bit box, without 32-bit packages, our m4/tcl.m4 script can not locate tclConfig.sh. Attaching a proposed patch, which will search /usr/lib64 as well. This patch + v3 compiles fine on my box (with the exception that the v3 patch is no longer in sync with the trunk).
22. 0001-Properly-search-usr-lib64-for-tclConfig.sh.patch looks good to me. No conflicts with Tcl license to make that change, either. In case you were concerned. Cheers, Andrew
23. **body:** John, For the 'ts-64bit-v3.patch': adjust the $CFLAGS and $CXXFLAGS in your 'ac_cv_sizeof_int_p' test block so that the flags are appended to the string instead of prefixing the string. In this way, you will then be able to let the package builder override the architecture values like so: ./configure CFLAGS=-march=core2 CXXFLAGS=-march=core2 The g++ commands will come out a little redundant: g++ -DPACKAGE_NAME=... -march=pentium -march=core2 ... But g++ will honor the last param passed of the same type so core2 ends up taking effect over i386. Cheers, Andrew
    label: code-design
24. Andrew, Leif's patch deals with this by not adding the -march=pentium if there is already an -march supplied. john
25. includes all patches up to through 0001-Properly-search-usr-lib64-for-tclConfig.sh.patch also sync'd with mainline
26. To make tracking of the patchset easier I have a git server: git://jplevyak.homeip.net/trafficserver If you add this as an additional remote you can keep in sync with both the x86_64 patchset and the current trunk (if I fall behind) using git-merge. I will try not to let it fall too far behind the trunk.
27. **body:** v4 worked fine on my 32-bit box, but a few comments: 1) Can we restore the -g for the release build? -ggdb3 is extremely verbose, making all the binaries 3x as large. 2) If we're no incorporating the patches I did for -march= detection, I think we need to do what Andrew suggested, and change the order in configure.ac, e.g. CCFLAGS="-march=pentium $CCFLAGS" CXXFLAGS="-march=pentium $CXXFLAGS" 3) Finally, I did a test with -march=i486 with the v4 patch, and it does not work. So -march=pentium is a minimum. I think we should change this to -march=i586 though, it's more in-line with the naming conventions used by the distros. I'm doing a second review of the actual code changes too, hopefully we can land this soon.
    label: code-design
28. Errors I got from gmake -k when trying to compile on RHEL4 32bit: [yts_dev bcall@armybright release]$ uname -a Linux armybright.corp.yahoo.com 2.6.9-78.EL #1 Wed Jul 9 15:27:01 EDT 2008 i386 [yts_dev bcall@armybright release]$ gcc --version gcc (GCC) 3.4.6 20060404 (Red Hat 3.4.6-10) Copyright (C) 2006 Free Software Foundation, Inc. This is free software; see the source for copying

conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. [yts_dev bcall@armybright release]$ cat /etc/redhat-release Red Hat Enterprise Linux AS release 4 (Nahant Update 7)

29. **body:** This allows building 32 bits with patch #4 applied. It uses full memory barrier for 64 bit atomic operations but cas64 is a hack
    label: test
30. **body:** The old 32-bit atomic code should probably be used for the atomics for gcc3. These is stored in the file: ink_atomic_solaris_i86pc.s which should probably be renamed. I am testing a patch to restore these and will have a new rollup out shortly.
    label: code-design
31. newest patch with pentium->i586 + all the old patches + restoring old 32-bit atomics
32. Oh, and I restored -g for release.
33. I was still getting the problem: ../../../external-apache/iocore/cluster/ClusterCache.cc:1794: warning: cast from pointer to integer of different size Attached is a patch to fix this issue. It now compiles and runs under rhel4 32bit.
34. Bryan, could you check out this patch, it avoids the #ifdef.
35. Yeah, that was another way I was thinking it would work. Worked well with RHEL4 32bit and fc11 64bit with and without -m32.
36. I committed the patches last night (v5 and cluster). John, Did you want to give it a once over and close out the bug? Leif and I tested it out on every available Linux distro we had installed (fedora 11 32bit and 64bit; rhel 4 and 5 32bit) There might have been an Ubuntu in there too, but I don't remember now...
37. I've tested it out Ubuntu 9.04 64bit and so far no issues.
38. Reviewed and tested. Looks good so far, so I am closing this issue.