

GROMACS - Bug #91

Problem with Gromacs intallation on IBM p655 series Power4+ SMP

07/11/2006 07:11 PM - Eugene empty

Status: Closed	
Priority: High	
Assignee: Erik Lindahl	
Category: mdrun	
Target version: 3.3.1	
Affected version - extra info:	Difficulty: uncategorized
Affected version:	
Description	
Hi,	
My configuration was:	
<pre>./configure --enable-mpi --disable-nice --program-suffix=_mpi --prefix=\$HOME/gromacs --disable-float</pre>	
The error is below:	
ERROR: 0031-808 Hostfile or pool must be used to request nodes configure:6801: error: cannot compute sizeof (int), 77	
So, MPI program cannot be run from the command line in the same way as serial programs (through lsubmit or poe only). How can I avoid those test runs in the configure file and get a working mpi-version of Gromacs?	
Thanks, Eugene	
config.log file:	
..... . . . configure:6352: checking for int configure:6384: mpcc -c -O3 -qarch=pwr3 -qtune=pwr3 -qmaxmem=16384 - /usr/local/include confstest.c >&5 configure:6390: \$? = 0 configure:6394: test -z test ! -s confstest.err configure:6397: \$? = 0 configure:6400: test -s confstest.o configure:6403: \$? = 0 configure:6414: result: yes configure:6417: checking size of int configure:6784: mpcc -o confstest -O3 -qarch=pwr3 -qtune=pwr3 -qmaxmem=16384 - /usr/local/include -L/usr/local/lib -bmaxdata:0x80000000 confstest.c >&5 configure:6787: \$? = 0 configure:6789: ./confstest ERROR: 0031-808 Hostfile or pool must be used to request nodes configure:6792: \$? = 255 configure: program exited with status 255 configure: failed program was:	
<pre>/* confdefs.h. */ #define PACKAGE_NAME "GROMACS" #define PACKAGE_TARNAME "gromacs" #define PACKAGE_VERSION "3.3.1" #define PACKAGE_STRING "GROMACS 3.3.1" #define PACKAGE_BUGREPORT " gmj-users@gromacs.org" #define PACKAGE "gromacs" #define VERSION "3.3.1" #define GMX_DOUBLE #define GMX_FORTRAN #define GMX_SOFTWARE_SQRT #define GMX_QMMM_GAUSSIAN #define BUILD_TIME "Tue Jul 11 12:03:47 EDT 2006" #define BUILD_USER "eugene@cmms23.cmms.pitt.edu" #define BUILD_MACHINE "AIX 2 000729294C00"</pre>	

```

| #define GMX_MPI | #define F77_FUNC(name,NAME) name | #define F77_FUNC_(name,NAME) name | #define
F77_OR_C_FUNC(name,NAME) F77_FUNC(name,NAME) | #define F77_OR_C_FUNC_(name,NAME) F77_FUNC_(name,NAME) |
#define GMX_INTEGER_BIG_ENDIAN | #define HAVE_SYS_TYPES_H 1 | #define HAVE_SYS_STAT_H 1 | #define
HAVE_STDLIB_H 1 | #define HAVE_STRING_H 1 | #define HAVE_MEMORY_H 1 | #define HAVE_STRINGS_H 1 | #define
HAVE_INTTYPES_H 1 | #define HAVE_STDINT_H 1 | #define HAVE_UNISTD_H 1 | /* end confdefs.h. */ | #include <stdio.h> | #if
HAVE_SYS_TYPES_H | # include <sys/types.h> | #endif | #if HAVE_SYS_STAT_H | # include <sys/stat.h> | #endif | #if
STDC_HEADERS | # include <stdlib.h> | # include <stddef.h> | #else | # if HAVE_STDLIB_H | # include <stdlib.h> | # endif | #endif |
#if HAVE_STRING_H | # if !STDC_HEADERS && HAVE_MEMORY_H | # include <memory.h> | # endif | # include <string.h> |
#endif | #if HAVE_STRINGS_H | /* end confdefs.h. */ | #include <stdio.h> | #if HAVE_SYS_TYPES_H | # include <sys/types.h> |
#endif | #if HAVE_SYS_STAT_H | # include <sys/stat.h> | #endif | #if STDC_HEADERS | # include <stdlib.h> | # include <stddef.h> |
#else | # if HAVE_STDLIB_H | # include <stdlib.h> | # endif | #endif | #if HAVE_STRING_H | # if !STDC_HEADERS &&
HAVE_MEMORY_H | # include <memory.h> | # endif | # include <string.h> | #endif | #if HAVE_STRINGS_H | # include <strings.h> |
#endif | #if HAVE_INTTYPES_H | # include <inttypes.h> | #else | # if HAVE_STDINT_H | # include <stdint.h> | # endif | #endif | #if
HAVE_UNISTD_H | # include <unistd.h> | #endif | # include <unistd.h> | #endif | long longval () { return (long) (sizeof (int)); } |
unsigned long ulongval () { return (long) (sizeof (int)); } | #include <stdio.h> | #include <stdlib.h> | #ifdef F77_DUMMY_MAIN | | #
ifdef __cplusplus | extern "C" | # endif | int F77_DUMMY_MAIN() { return 1; } | | #endif | int | main () | { | | FILE *f = fopen
("confest.val", "w"); | if (!f) | exit (1); | if (((long) (sizeof (int))) < 0) | { | long i = longval (); | if (i != ((long) (sizeof (int)))) |
exit (1); | fprintf (f, "%ld\n", i); | } | else | { | unsigned long i = ulongval (); | if (i != ((long) (sizeof (int)))) | exit (1); |
fprintf (f, "%lu\n", i); | } | exit (ferror (f) || fclose (f) != 0); | | ; | return 0; | }
configure:6801: error: cannot compute sizeof (int), 77
See `config.log' for more details.
.
.
.
.
.....

```

History

#1 - 07/12/2006 11:05 AM - Erik Lindahl

Hi,

I experienced exactly this problem myself 4-5 years ago, and worked around it by introducing a `CC_FOR_BUILD`, which is separate from the compiler used for building the final executable.

Normally, all autoconf tests are cross-compile-compliant, so they shouldn't have to execute any code for the tests. Unfortunately it seems as if `AC_CHECK_SIZEOF()` has changed since then.

In the immediate term, you can try one of:

1. Request an interactive node through the queue system and compile there
2. Remove the three `AC_CHECK_SIZEOF()` lines from `configure.ac`, rerun 'autoconf', run the configure step, and then manually insert the values for `SIZEOF_INT`, `SIZEOF_LONG_INT`, `SIZEOF_LONG_LONG_INT` into `src/config.h`.

I'll try to find a better macro for the checking, but that will involve contacting autoconf authors, and likely take a while...

/Erik

#2 - 07/21/2006 11:44 PM - Eugene empty

(In reply to comment [#1](#))

Hi,
I experienced exactly this problem myself 4-5 years ago, and worked around it

by introducing a

`CC_FOR_BUILD`, which is separate from the compiler used for building the final

executable.

Normally, all autoconf tests are cross-compile-compliant, so they shouldn't

have to execute any code

for the tests. Unfortunately it seems as if `AC_CHECK_SIZEOF()` has changed

since then.

In the immediate term, you can try one of:

1. Request an interactive node through the queue system and compile there
2. Remove the three AC_CHECK_SIZEOF() lines from configure.ac,

rerun 'autoconf', run the configure

step, and then manually insert the values for SIZEOF_INT, SIZEOF_LONG_INT,

SIZEOF_LONG_LONG_INT

into src/config.h.

I'll try to find a better macro for the checking, but that will involve

contacting autoconf authors, and

likely take a while...

/Erik

Hi Erik,

I took the second way, edited configure.ac file, then reran 'autoconf', and manually defined SIZEOF_INT, SIZEOF_LONG_INT, SIZEOF_LONG_LONG_INT values in config.h file. After that everything went smoothly and I got a working parallel version of Gromacs.

Thanks,
Eugene