TEST REPORT
FEBRUARY 2007

Linpack HPL performance on Intel- and AMD-processor-based servers running Red Hat Enterprise Linux v.4.4

Executive summary

Intel Corporation (Intel) commissioned Principled Technologies (PT) to measure the LINPACK HPL performance of dual-processor servers using the following systems running Red Hat Enterprise Linux v.4.4:

- Supermicro SuperServer 6025B-TR+V with dual-core AMD Opteron processor model 2220 SE
- Supermicro SuperServer 6025B-TR+V with Quad-Core Intel Xeon processor X5355

KEY FINDINGS

- The Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s delivered 227 percent higher peak performance on the Linpack WR00C2C4 test than the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs (see Figure 1).

In this section, we discuss the best results for each server. For complete details of the performance of each benchmark for each server, see the Test results section.

Linpack HPL produces performance results as the number of billions of floating point operations per second (Gigaflops per second, or Gflops per second, or Gflops/s).

Figure 1 shows the Linpack Benchmark results, in Gflops/s, of a sample test WR00C2C4 (one of the 18 Linpack benchmarks we ran) for the test servers. To obtain the median results, we performed three runs with each server and selected the benchmark on which the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs produced its highest score. That benchmark was WR00C2C4. We then took the median result from the three runs of that benchmark for both servers. A higher Linpack score is better, because it indicates that the server is faster at floating point operations.

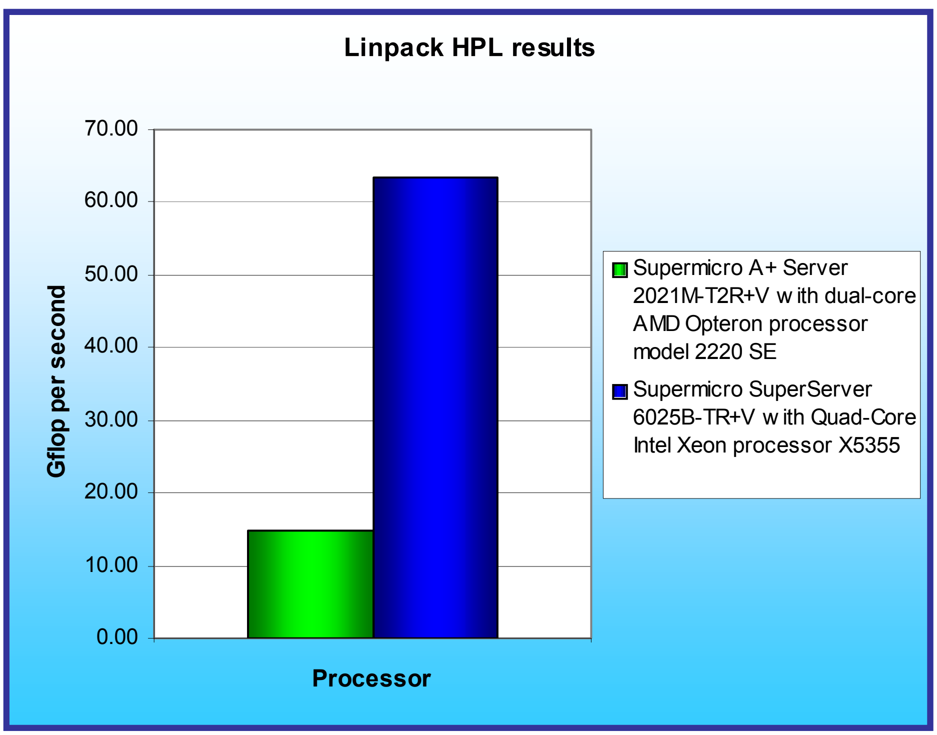

Figure 1: Linpack HPL results, in Gflops per second of a sample test WR00C2C4 for the test servers. Higher numbers of Gflops per second are better.

The Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s produced the higher results, 63.47 Gflops/s, while the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs achieved 19.41 Gflops/s. The Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s thus delivered a 227 percent performance increase over the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs.

Figure 2 shows the complete median Linpack HPL run results, in Gflops/s, for each of the 18 benchmarks we ran on both servers.

Test name	Supermicro A+ Server 2021M-T2R+V with dual-core AMD Opteron processor model 2220 SE	Supermicro SuperServer 6025B-TR+V with Quad-Core Intel Xeon processor X5355
WR00L2L2	19.35	63.47
WR00L2L4	19.50	63.53
WR00L2C2	19.45	63.42
WR00L2C4	19.42	63.52
WR00L2R2	19.43	63.45
WR00L2R4	19.28	63.55
WR00C2L2	19.35	63.30
WR00C2L4	19.49	63.49
WR00C2C2	19.44	63.33
WR00C2C4	19.41	63.47
WR00C2R2	19.50	63.55
WR00C2R4	19.49	63.59
WR00R2L2	19.43	63.43
WR00R2L4	19.42	63.45
WR00R2C2	19.41	63.39
WR00R2C4	19.39	63.57
WR00R2R2	19.48	63.53
WR00R2R4	19.51	63.57

Figure 2: Median Linpack HPL run results of all 18 benchmarks, in Gflops per second, for each server. Higher numbers are better.

Workload

The Linpack benchmark is an industry-standard benchmark created in 1979 by Jack Dongarra. The Linpack benchmark solves linear equations and uses the speed of the system under test at that task as a measure of the system's floating-point performance. Linpack reports its results in billions of floating point operations per second, or Gflops/s.

We used the HPL version of the Linpack Benchmark. HPL is a portable implementation of the High Performance Computing Linpack benchmark that generates, solves, checks, and times the solution process of a random dense linear system of equations. The HPL software package uses 64-bit floating point arithmetic and portable routines for linear algebra operations and message passing. The HPL code offers the advantage of allowing testers to select from among multiple factorization algorithms.

Test results

Figure 3 shows the Linpack HPL results for the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs for all three runs. Run 2 produced the median results.

Supermicro A+ Server 2021M-T2R+V with dual-core AMD Opteron processor model 2220 SE			
Test name	**Run 1**	**Run 2**	**Run 3**
WR00L2L2	19.24	19.35	19.29
WR00L2L4	19.37	19.50	19.46
WR00L2C2	19.53	19.45	19.37
WR00L2C4	19.49	19.42	19.40
WR00L2R2	19.44	19.43	19.44
WR00L2R4	19.48	19.28	19.40
WR00C2L2	19.42	19.35	19.36
WR00C2L4	19.49	19.49	19.51
WR00C2C2	19.44	19.44	19.45
WR00C2C4	19.53	19.41	19.38
WR00C2R2	19.41	19.50	19.41
WR00C2R4	19.49	19.49	19.30
WR00R2L2	19.47	19.43	19.38
WR00R2L4	19.39	19.42	19.35
WR00R2C2	19.39	19.41	19.44
WR00R2C4	19.47	19.39	19.48
WR00R2R2	19.37	19.48	19.38
WR00R2R4	19.45	19.51	19.41

Figure 3: Linpack HPL benchmark results, in Gflops per second, for the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs. Higher numbers are better.

Figure 4 shows the Linpack HPL results for the Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s for all three runs. Run 3 produced the median results.

Supermicro SuperServer 6025B-TR+V with Quad-Core Intel Xeon processor X5355			
Test name	Run 1	Run 2	Run 3
WR00L2L2	63.52	62.92	63.47
WR00L2L4	63.81	63.37	63.53
WR00L2C2	63.47	63.14	63.42
WR00L2C4	63.90	63.45	63.52
WR00L2R2	63.76	63.17	63.45
WR00L2R4	63.83	63.40	63.55
WR00C2L2	63.67	63.16	63.30
WR00C2L4	63.69	63.29	63.49
WR00C2C2	63.66	63.01	63.33
WR00C2C4	63.69	63.07	63.47
WR00C2R2	63.93	63.28	63.55
WR00C2R4	63.83	63.32	63.59
WR00R2L2	63.65	62.98	63.43
WR00R2L4	63.70	63.14	63.45
WR00R2C2	63.61	63.07	63.39
WR00R2C4	63.86	63.21	63.57
WR00R2R2	63.63	62.93	63.53
WR00R2R4	63.74	63.31	63.57

Figure 4: Linpack HPL benchmark results, in Gflops per second, for the Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s. Higher numbers are better.

Test methodology

Figure 5 summarizes some key aspects of the configurations of the server systems; Appendix A provides detailed configuration information.

Server	Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs	Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s
Processor frequency (GHz)	2.8 GHz	2.66 GHz
System bus	2000 MHz HyperTransport	1333 MHz
Number of processor packages	2	2
Number of cores per processor package	2	4
Number of hardware threads per core	1	1
Motherboard	Super H8DME-2	Supermicro X7DBE+
Chipset	NVIDIA MCP55 Pro	Intel 5000P Chipset
RAM (8GB in each)	PC2-5300	PC2-5300 FBDIMM
Hard Drive	Western Digital WD740ADFD 74 GB 10,000 RPM	Western Digital WD740ADFD 74 GB 10,000 RPM
NICs	NVIDIA MCP55 Pro Chipset Dual-Port Ethernet Controller	Intel PRO/1000 EB Network Dual Port Network Connection

Figure 5: Summary of some key aspects of the server configurations.

Intel configured and provided both servers.

With the following exceptions, we used the default BIOS settings on each server: on the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs, we changed the OS installation option to Linux.

We began by installing a fresh copy of Red Hat Enterprise Linux v.4.4 on both servers. We installed each system with the default operating system (OS) installation options.

Linpack configuration

We used the following three software components for this test:

- MVAPICH2-0.9.8 (source: http://nowlab.cse.ohio-state.edu/projects/mpi-iba/)
- GotoBLAS (source: http://www.tacc.utexas.edu/resources/software/)
- HPL (source: http://www.netlib.org/benchmark/hpl/)

For both servers and all software components, we compiled the components from source. We accepted the default file system layout for both the Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s and the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs. We then created our work directory under the root directory on each server. To simplify the description below, we will refer to the working directory as $HPL_HOME.

To build MVAPICH2, we did the following:

1. Unpack the MVAPICH2-0.9.8.tar.gz archive in $HPL_HOME. This step creates the mvaich2-0.9.8 directory.
2. Type "cd to $HPL_HOME/mvapich2-0.9.8".
3. Type “./make.mvapich2.tcp”. (Note that the two test systems did not have host channel adaptors, which is why we built the program on TCP.)
4. Create the “.mpd.password” and “.mod.conf” files as the MVAPICH documentation specifies.
5. Put a single entry in the mpd.hosts file, “localhost".
6. Create a setup.sh script to set the PATH environment variable to include “$HPL_HOME/mvapich2-0.9.8/bin”.
7. Verify MVAPICH2-0.9.8 is working by typing the following commands:
   - . ./setup.sh (Note the necessary space after the first dot.)
   - mpd & (launches the mp daemon)
   - mpdtrace (will show you the systems it's running on)
   - mpdallexit (kills the daemon)

To build GotoBLAS, we did the following:

1. Unpack the GotoBLAS-1.10.tar.gz archive in $HPL_HOME. This creates the GotoBLAS directory.
2. Type "cd to $HPL_HOME /GotoBLAS".
3. Edit the Makefile.rule file and make the following changes:
   - BINARY64=1
   - SMP=1
   - MAX_THREADS=# of cores – 8 for the Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s server and 4 for the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs. (The maximum number of threads reflects the number of available execution units on each system.)
4. Run “./quickbuild.64bit”.
5. Copy the “libgoto_<arch>-r1-.10.a” file to $HPL_HOME. On the Supermicro SuperServer 6025B-TR+V Red Hat system, this file's name will be “libgoto_core2p-r1-.10.a”. On the Supermicro A+ Server 2021M-T2R+V system, the file name will be “libgoto_opteronp-r1-.10.a”.

To build HPL, we did following:

1. Unpack the hpl.gz archive in $HPL_HOME/mvapich2-0.9.8. This step creates the hpl directory.
2. Type "cd to $HPL_HOME/mvapich2-0.9.8/hpl".

3. Edit the Makefile and set arch to “em64t” (for the Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s server) or “opt64” (for the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs).
4. Create the Make.em64t or Make.opt64 file, as appropriate for the system (see Appendix B).
5. Type “make”.
6. After you install the benchmark, use the HPL.dat file in Appendix B.

To run HPL, we did the following:

1. Type "cd to /$HPL_HOME/mvapich2-0.9.8".
2. Type “. ./setup.sh “. (Note the necessary space after the first dot.)
3. Type “mpd &” to start the mpi daemon.
4. Type “cd hpl/bin/em64t“ (for the Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s server) or “cd hpl/bin/opt64“ (for the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs).
5. Type “mpirun -n 2 ./xhpl".
6. By default, the output goes to the screen. Either redirect the output to a file to save it or copy it from the console window after the test.

Appendix A – Test server configuration information

This appendix provides detailed configuration information about each of the test server systems, which we list in alphabetical order.

Systems	Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs	Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s
General processor setup		
Number of processor packages	2	2
Number of cores per processor package	2	4
Number of hardware threads per core	1	1
CPU		
Vendor	AMD	Intel
Name	dual-core AMD Opteron processor model 2220 SE	Quad-Core Intel Xeon processor X5355
Stepping	2	7
Socket type	F	LGA 771
Core frequency (GHz)	2.8 GHz	2.66 GHz
Front-side bus frequency (MHz)	2000 MHz HyperTransport	1333 MHz
L1 Cache	64 KB + 64 KB (per core)	32 KB + 32 KB (per core)
L2 Cache	2 x 1 MB	2 x 4MB (each 4MB shared by 2 cores)
Platform		
Vendor and model number	dual-core AMD Opteron processor model 2220 SE-based server	Quad-Core Intel Xeon processor X5355-based server
Motherboard model number	Super H8DME-2	Supermicro X7DBE+
Motherboard chipset	NVIDIA MCP55 Pro	Intel 5000P Chipset
Motherboard revision number	A2	92
Motherboard serial number	Q5785G16010104	TM66S06520
BIOS name and version	American Megatrends Inc. AMIBIOS 08.00.14 11/28/06	Phoenix BIOS DB8A026 Rev 1.1c
BIOS settings	OS installation Linux	Default
Memory module(s)		
Vendor and model number	Hynix HYMP525P72BP4-Y5	Kingston KVR667D2D4F5/2G
Type	PC-5300	PC2-5300 FBDIMM
Speed (MHz)	667 MHz	667 MHz
Speed in the system currently running @ (MHz)	667 MHz	667 MHz
Timing/Latency (tCL-tRCD-iRP-tRASmin)	5-5-5-15	5-5-5-15
Size	8186 MB	8196 MB
Number of RAM modules	4	4
Chip organization	Double-Sided	Double-Sided
Hard disk		
Vendor and model number	Western Digital Raptor WD740ADFD	Western Digital Raptor WD740ADFD
Number of disks in system	1	1
Size	74 GB	74 GB

Buffer Size	8 MB	8 MB
RPM	10,000	10,000
Type	SATA	SATA
Controller	NVIDIA MCP55 Pro SATA2 Controller	Intel 3100 Chipset SATA Controller
Controller driver	sata_nv	Ata_piix
Operating system		
Name	Red Hat Enterprise Linux 4 Advanced Server	Red Hat Enterprise Linux 4 Advanced Server
Build number	v.4.4	v.4.4
File system	Ext3	Ext3
Kernel	2.6.9-42.ELsmp	2.6.9-42.ELsmp
Language	English	English
Graphics		
Vendor and model number	ATI ES1000	ATI ES1000
Chipset	ATI ES1000 PCI	ATI ES1000 PCI
Type	Integrated	Integrated
Resolution	1024 x 768	1024 x 768
Driver	ATI ES1000	ATI ES1000
Network card/subsystem		
Vendor and model number	NVIDIA MCP55 Pro Chipset Dual-Port Ethernet Controller	Intel PRO/1000 EB Network Dual Port Network Connection
Type	Integrated	Integrated
Driver	eth0	eth0
Optical drive		
Vendor and model number	Matshita DVD-ROM SR-8178	Matshita DVD-ROM SR-8178
Type	DVD-ROM	CD/DVD
Interface	Internal	Internal
Dual/Single layer	Single	Single
USB ports		
Number	4	4
Type	USB 2.0	USB 2.0

Figure 6: Detailed system configuration information for the test servers.

Appendix B – Linpack HPL configuration files

This appendix contains the benchmark configuration files we used to test the servers.

Makefile.em64t: Configuration file for the Supermicro SuperServer 6025B-TR+V with two Quad-Core Intel Xeon processor X5355s

```
#
# -- High Performance Computing Linpack Benchmark (HPL)
#    HPL - 1.0a - January 20, 2004
#    Antoine P. Petitet
#    University of Tennessee, Knoxville
#    Innovative Computing Laboratories
#    (C) Copyright 2000-2004 All Rights Reserved
#
# -- Copyright notice and Licensing terms:
#
# Redistribution  and  use in  source and binary forms, with or without
# modification, are  permitted provided  that the following  conditions
# are met:
#
# 1. Redistributions  of  source  code  must retain the above copyright
# notice, this list of conditions and the following disclaimer.
#
# 2. Redistributions in binary form must reproduce  the above copyright
# notice, this list of conditions,  and the following disclaimer in the
# documentation and/or other materials provided with the distribution.
#
# 3. All  advertising  materials  mentioning  features  or  use of this
# software must display the following acknowledgement:
# This  product  includes  software  developed  at  the  University  of
# Tennessee, Knoxville, Innovative Computing Laboratories.
#
# 4. The name of the  University,  the name of the  Laboratory,  or the
# names  of  its  contributors  may  not  be used to endorse or promote
# products  derived   from   this  software  without  specific  written
# permission.
#
# -- Disclaimer:
#
# THIS  SOFTWARE  IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES,  INCLUDING,  BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
# A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE UNIVERSITY
# OR  CONTRIBUTORS  BE  LIABLE FOR ANY  DIRECT,  INDIRECT,  INCIDENTAL,
# SPECIAL,  EXEMPLARY,  OR  CONSEQUENTIAL DAMAGES  (INCLUDING,  BUT NOT
# LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA OR PROFITS; OR BUSINESS INTERRUPTION)  HOWEVER CAUSED AND ON ANY
# THEORY OF LIABILITY, WHETHER IN CONTRACT,  STRICT LIABILITY,  OR TORT
# (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# ######################################################################
#
# ----------------------------------------------------------------------
# - shell --------------------------------------------------------------
# ----------------------------------------------------------------------
#
SHELL        = /bin/sh
#
CD           = cd
CP           = cp
LN_S         = ln -s
MKDIR        = mkdir
RM           = /bin/rm -f
TOUCH        = touch
#
# ----------------------------------------------------------------------
```

```
# - Platform identifier ------------------------------------------------
# ----------------------------------------------------------------------
#
ARCH         = em64t
#
# ----------------------------------------------------------------------
# - HPL Directory Structure / HPL library ------------------------------
# ----------------------------------------------------------------------
#
#TOPdir       = $(HOME)/mvapich/hpl
TOPdir       = /hpl_stuff/mvapich2-0.9.8/hpl
INCdir       = $(TOPdir)/include
BINdir       = $(TOPdir)/bin/$(ARCH)
LIBdir       = $(TOPdir)/lib/$(ARCH)
#
HPLlib       = $(LIBdir)/libhpl.a
#
# ----------------------------------------------------------------------
# - Message Passing library (MPI) --------------------------------------
# ----------------------------------------------------------------------
# MPinc tells the  C  compiler where to find the Message Passing library
# header files,  MPlib  is defined  to be the name of  the library to be
# used. The variable MPdir is only used for defining MPinc and MPlib.
#
#MPdir        = /opt/mvapich-0.9.7-1/gnu
MPdir        = /hpl_stuff/mvapich2-0.9.8
MPinc        = -I$(MPdir)/include
MPlib        = -L$(MPdir)/lib $(MPdir)/lib/libmpich.a
#
# ----------------------------------------------------------------------
# - Linear Algebra library (BLAS or VSIPL) -----------------------------
# ----------------------------------------------------------------------
# LAinc tells the  C  compiler where to find the Linear Algebra  library
# header files,  LAlib  is defined  to be the name of  the library to be
# used. The variable LAdir is only used for defining LAinc and LAlib.
#
LAdir        = /hpl_stuff/mvapich2-0.9.8/hpl
LAinc        =
#LAlib        = $(LAdir)/libgoto_core2p-r1.03.a
#LAlib        = $(LAdir)/libgoto_opt64-r0.94-2.so
LAlib        = $(LAdir)/libgoto_core2p-r1.10.a
##
# ----------------------------------------------------------------------
# - F77 / C interface --------------------------------------------------
# ----------------------------------------------------------------------
# You can skip this section  if and only if  you are not planning to use
# a  BLAS  library featuring a Fortran 77 interface.  Otherwise,  it  is
# necessary  to  fill out the  F2CDEFS  variable  with  the  appropriate
# options.  **One and only one**  option should be chosen in **each** of
# the 3 following categories:
#
# 1) name space (How C calls a Fortran 77 routine)
#
# -DAdd_              : all lower case and a suffixed underscore  (Suns,
#                       Intel, ...),                           [default]
# -DNoChange          : all lower case (IBM RS6000),
# -DUpCase            : all upper case (Cray),
# -DAdd__             : the FORTRAN compiler in use is f2c.
#
# 2) C and Fortran 77 integer mapping
#
# -DF77_INTEGER=int   : Fortran 77 INTEGER is a C int,         [default]
# -DF77_INTEGER=long  : Fortran 77 INTEGER is a C long,
# -DF77_INTEGER=short : Fortran 77 INTEGER is a C short.
#
# 3) Fortran 77 string handling
#
# -DStringSunStyle    : The string address is passed at the string loca-
#                       tion on the stack, and the string length is then
```

```
#                     passed as  an  F77_INTEGER  after  all  explicit
#                     stack arguments,                       [default]
# -DStringStructPtr   : The address  of  a  structure  is  passed  by  a
#                     Fortran 77  string,  and the structure is of the
#                     form: struct {char *cp; F77_INTEGER len;},
# -DStringStructVal   : A structure is passed by value for each  Fortran
#                     77 string,  and  the  structure is  of the form:
#                     struct {char *cp; F77_INTEGER len;},
# -DStringCrayStyle   : Special option for  Cray  machines,  which  uses
#                     Cray  fcd  (fortran  character  descriptor)  for
#                     interoperation.
#
F2CDEFS      = -DF77_INTEGER=int
#
# ----------------------------------------------------------------------
# - HPL includes / libraries / specifics -------------------------------
# ----------------------------------------------------------------------
#
HPL_INCLUDES = -I$(INCdir) -I$(INCdir)/$(ARCH) $(LAinc) $(MPinc)
HPL_LIBS     = $(HPLlib) $(LAlib) $(MPlib) /hpl_stuff/mvapich2-0.9.8/hpl/src/xerbla.o
#
# - Compile time options -----------------------------------------------
#
# -DHPL_COPY_L           force the copy of the panel L before bcast;
# -DHPL_CALL_CBLAS       call the cblas interface;
# -DHPL_CALL_VSIPL       call the vsip  library;
# -DHPL_DETAILED_TIMING  enable detailed timers;
#
# By default HPL will:
#    *) not copy L before broadcast,
#    *) call the BLAS Fortran 77 interface,
#    *) not display detailed timing information.
#
#HPL_OPTS     = -DHPL_CALL_CBLAS
#
# ----------------------------------------------------------------------
#
HPL_DEFS     = $(F2CDEFS) $(HPL_OPTS) $(HPL_INCLUDES)
#
# ----------------------------------------------------------------------
# - Compilers / linkers - Optimization flags ---------------------------
# ----------------------------------------------------------------------
#
#CC          = /usr/bin/gcc
CC           = $(MPdir)/bin/mpicc
CCNOOPT      = $(HPL_DEFS)
CCFLAGS      = $(HPL_DEFS) -fomit-frame-pointer -O3 -funroll-loops -Bstatic
#CCFLAGS      = $(HPL_DEFS) -g *** For debugging
#
# On some platforms,  it is necessary  to use the Fortran linker to find
# the Fortran internals used in the BLAS library.
#
#LINKER       = /usr/bin/g77
LINKER       = $(MPdir)/bin/mpicc
#LINKFLAGS    = $(CCFLAGS) -Bstatic
LINKFLAGS    = $(CCFLAGS) -lm -B static
#LINKFLAGS    = $(CCNOOPT) -lm ***For debugging
#
ARCHIVER     = ar
ARFLAGS      = r
RANLIB       = echo
#
# ----------------------------------------------------------------------
```

Makefile.opt64: Configuration file for the Supermicro A+ Server 2021M-T2R+V with two dual-core AMD Opteron processor model 2220 SEs

```
#
#  -- High Performance Computing Linpack Benchmark (HPL)
#     HPL - 1.0a - January 20, 2004
#     Antoine P. Petitet
#     University of Tennessee, Knoxville
#     Innovative Computing Laboratories
#     (C) Copyright 2000-2004 All Rights Reserved
#
#  -- Copyright notice and Licensing terms:
#
#  Redistribution  and  use in  source and binary forms, with or without
#  modification, are  permitted provided  that the following  conditions
#  are met:
#
#  1. Redistributions  of  source  code  must retain the above copyright
#  notice, this list of conditions and the following disclaimer.
#
#  2. Redistributions in binary form must reproduce  the above copyright
#  notice, this list of conditions,  and the following disclaimer in the
#  documentation and/or other materials provided with the distribution.
#
#  3. All  advertising  materials  mentioning  features  or  use of this
#  software must display the following acknowledgement:
#  This  product  includes  software  developed  at  the  University  of
#  Tennessee, Knoxville, Innovative Computing Laboratories.
#
#  4. The name of the  University,  the name of the  Laboratory,  or the
#  names  of  its  contributors  may  not  be used to endorse or promote
#  products  derived   from   this  software  without  specific  written
#  permission.
#
#  -- Disclaimer:
#
#  THIS  SOFTWARE  IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#  ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES,  INCLUDING,  BUT NOT
#  LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
#  A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE UNIVERSITY
#  OR  CONTRIBUTORS  BE  LIABLE FOR ANY  DIRECT,  INDIRECT,  INCIDENTAL,
#  SPECIAL,  EXEMPLARY,  OR  CONSEQUENTIAL DAMAGES  (INCLUDING,  BUT NOT
#  LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
#  DATA OR PROFITS; OR BUSINESS INTERRUPTION)  HOWEVER CAUSED AND ON ANY
#  THEORY OF LIABILITY, WHETHER IN CONTRACT,  STRICT LIABILITY,  OR TORT
#  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
#  OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# ######################################################################
#
# ----------------------------------------------------------------------
# - shell --------------------------------------------------------------
# ----------------------------------------------------------------------
#
SHELL        = /bin/sh
#
CD           = cd
CP           = cp
LN_S         = ln -s
MKDIR        = mkdir
RM           = /bin/rm -f
TOUCH        = touch
#
# ----------------------------------------------------------------------
# - Platform identifier ------------------------------------------------
# ----------------------------------------------------------------------
#
ARCH         = opt64
#
```

```
# ----------------------------------------------------------------------
# - HPL Directory Structure / HPL library ------------------------------
# ----------------------------------------------------------------------
#
#TOPdir       = $(HOME)/mvapich/hpl
TOPdir       = /export/home/hpl_stuff/mvapich2-0.9.8/hpl
INCdir       = $(TOPdir)/include
BINdir       = $(TOPdir)/bin/$(ARCH)
LIBdir       = $(TOPdir)/lib/$(ARCH)
#
HPLlib       = $(LIBdir)/libhpl.a
#
# ----------------------------------------------------------------------
# - Message Passing library (MPI) --------------------------------------
# ----------------------------------------------------------------------
# MPinc tells the  C  compiler where to find the Message Passing library
# header files,  MPlib  is defined  to be the name of  the library to be
# used. The variable MPdir is only used for defining MPinc and MPlib.
#
#MPdir        = /opt/mvapich-0.9.7-1/gnu
MPdir        = /export/home/hpl_stuff/mvapich2-0.9.8
MPinc        = -I$(MPdir)/include
MPlib        = -L$(MPdir)/lib $(MPdir)/lib/libmpich.a
#
# ----------------------------------------------------------------------
# - Linear Algebra library (BLAS or VSIPL) -----------------------------
# ----------------------------------------------------------------------
# LAinc tells the  C  compiler where to find the Linear Algebra  library
# header files,  LAlib  is defined  to be the name of  the library to be
# used. The variable LAdir is only used for defining LAinc and LAlib.
#
LAdir        = /export/home/hpl_stuff/mvapich2-0.9.8/hpl
LAinc        =
#LAlib        = $(LAdir)/libgoto_core2p-r1.03.a
#LAlib        = $(LAdir)/libgoto_opt64-r0.94-2.so
#LAlib        = $(LAdir)/libgoto_core2p-r1.10.a
LAlib        = $(LAdir)/libgoto_opteronp-r1.10.a
##
# ----------------------------------------------------------------------
# - F77 / C interface --------------------------------------------------
# ----------------------------------------------------------------------
# You can skip this section  if and only if  you are not planning to use
# a  BLAS  library featuring a Fortran 77 interface.  Otherwise,  it  is
# necessary  to  fill out the  F2CDEFS  variable  with  the  appropriate
# options.  **One and only one**  option should be chosen in **each** of
# the 3 following categories:
#
# 1) name space (How C calls a Fortran 77 routine)
#
# -DAdd_              : all lower case and a suffixed underscore  (Suns,
#                       Intel, ...),                           [default]
# -DNoChange          : all lower case (IBM RS6000),
# -DUpCase            : all upper case (Cray),
# -DAdd__             : the FORTRAN compiler in use is f2c.
#
# 2) C and Fortran 77 integer mapping
#
# -DF77_INTEGER=int   : Fortran 77 INTEGER is a C int,         [default]
# -DF77_INTEGER=long  : Fortran 77 INTEGER is a C long,
# -DF77_INTEGER=short : Fortran 77 INTEGER is a C short.
#
# 3) Fortran 77 string handling
#
# -DStringSunStyle    : The string address is passed at the string loca-
#                       tion on the stack, and the string length is then
#                       passed as  an  F77_INTEGER  after  all  explicit
#                       stack arguments,                       [default]
# -DStringStructPtr   : The address  of  a  structure  is  passed  by  a
#                       Fortran 77  string,  and the structure is of the
```

```
#                    form: struct {char *cp; F77_INTEGER len;},
# -DStringStructVal   : A structure is passed by value for each  Fortran
#                    77 string,  and  the  structure is  of the form:
#                    struct {char *cp; F77_INTEGER len;},
# -DStringCrayStyle   : Special option for  Cray  machines,  which  uses
#                    Cray  fcd  (fortran  character  descriptor)  for
#                    interoperation.
#
F2CDEFS      = -DF77_INTEGER=int
#
# ----------------------------------------------------------------------
# - HPL includes / libraries / specifics -------------------------------
# ----------------------------------------------------------------------
#
HPL_INCLUDES = -I$(INCdir) -I$(INCdir)/$(ARCH) $(LAinc) $(MPinc)
#HPL_LIBS     = $(HPLlib) $(LAlib) $(MPlib) /export/home/hpl_stuff/mvapich2-0.9.8/hpl/src/xerbla.o
HPL_LIBS     = $(HPLlib) $(LAlib) $(MPlib)
#
# - Compile time options -----------------------------------------------
#
# -DHPL_COPY_L           force the copy of the panel L before bcast;
# -DHPL_CALL_CBLAS       call the cblas interface;
# -DHPL_CALL_VSIPL       call the vsip  library;
# -DHPL_DETAILED_TIMING  enable detailed timers;
#
# By default HPL will:
#    *) not copy L before broadcast,
#    *) call the BLAS Fortran 77 interface,
#    *) not display detailed timing information.
#
#HPL_OPTS     = -DHPL_CALL_CBLAS
#
# ----------------------------------------------------------------------
#
HPL_DEFS     = $(F2CDEFS) $(HPL_OPTS) $(HPL_INCLUDES)
#
# ----------------------------------------------------------------------
# - Compilers / linkers - Optimization flags ---------------------------
# ----------------------------------------------------------------------
#
#CC           = /usr/bin/gcc
CC           = $(MPdir)/bin/mpicc
CCNOOPT      = $(HPL_DEFS)
CCFLAGS      = $(HPL_DEFS) -fomit-frame-pointer -O3 -funroll-loops -Bstatic
#CCFLAGS      = $(HPL_DEFS) -g *** For debugging
#
# On some platforms,  it is necessary  to use the Fortran linker to find
# the Fortran internals used in the BLAS library.
#
#LINKER       = /usr/bin/g77
LINKER       = $(MPdir)/bin/mpicc
#LINKFLAGS    = $(CCFLAGS) -Bstatic
LINKFLAGS    = $(CCFLAGS) -lm -B static
#LINKFLAGS    = $(CCNOOPT) -lm ***For debugging
#
ARCHIVER     = ar
ARFLAGS      = r
RANLIB       = echo
#
# ----------------------------------------------------------------------
```

HPL.dat file (same on both systems)

```
HPLinpack benchmark input file
Principled Technologies, Inc
HPL.out       output file name (if any)
6             device out (6=stdout,7=stderr,file)
1             # of problems sizes (N)
30000         Ns
1             # of NBs
256           NBs
0             PMAP process mapping (0=Row-,1=Column-major)
1             # of process grids (P x Q)
1             Ps
2             Qs
16.0          threshold
3             # of panel fact
0             PFACTs (0=left, 1=Crout, 2=Right)
2             # of recursive stopping criterium
2 4           NBMINs (>= 1)
1             # of panels in recursion
2             NDIVs
3             # of recursive panel fact.
0 1 2         RFACTs (0=left, 1=Crout, 2=Right)
1             # of broadcast
0             BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1             # of lookahead depth
0             DEPTHs (>=0)
2             SWAP (0=bin-exch,1=long,2=mix)
64            swapping threshold
0             L1 in (0=transposed,1=no-transposed) form
0             U  in (0=transposed,1=no-transposed) form
1             Equilibration (0=no,1=yes)
8             memory alignment in double (> 0)
```

Principled Technologies, Inc.
1007 Slater Road, Suite 250
Durham, NC 27703
www.principledtechnologies.com
info@principledtechnologies.com