Redbooks Paper

Qiyou Jiang

Porting and tuning the Amsterdam Density Functional package on IBM p575+ Linux Platforms

The Amsterdam Density Functional (ADF) package is a highly requested application in our work with IBM® benchmarks. However, there was no official Linux® on POWER™ version of the ADF package in 2006. So, we ported and tuned the ADF package to our IBM p575+ Linux platform.

This IBM Redpaper describes our porting experience and compares the performance of the ADF package on both Linux and AIX® to demonstrate the impact of different interconnects, file systems, and kernels. Our results show that the ADF package has great performance and scalability on both platforms. In addition, we include tips for troubleshooting and achieving better performance. There is now an official Linux on POWER version available from SCM.

Introduction

The ADF package is a quantum chemistry Fortran program that is based on the Density Functional Theory (DFT). The commercial version of the package is maintained and distributed by Scientific Computing and Modelling NV (SCM), a company that is based in Amsterdam.[1]

SCM has ported the ADF package to many different platforms, such as IBM AIX (IBM POWER3™, IBM POWER4™, and IBM POWER5™), HP Itanium2 (HP-UX and Linux), SGI Itanium2 Linux (Altix), SGI IRIX 6.5, Mac OS X, PC Linux (RedHat 9 or later), PC Linux 64 (Nocona and Opteron), and others. These versions are official and supported by SCM. ADF users can obtain these official versions by sending requests to support@scm.com.

In recent years, we have used the official ADF AIX version for many of our benchmark requests, and it has shown great performance on IBM AIX platforms.[2]

Because we received many requests for benchmarking ADF Linux on POWER platforms in 2006, we ported the ADF package to Linux on POWER. We made some successful runs and verified the correctness of each result. In 2007, SCM has also ported the ADF package to Linux on POWER, and this official version should be used.

This IBM Redpaper describes the procedures in porting ADF to Linux on POWER and provides performance tips based on our benchmarking experience. Other ADF users and IBM teams might find this information useful.

Porting the ADF package to IBM Linux platforms

Our porting work is based on the development version of the ADF package, which is an all-source-code package provided by SCM. This development version includes the zipped tar file fix2006.tar.gz.

To create the parallel 64-bit addressing ADF Linux on POWER version (using poe), follow these steps:

1. Set up environment variables. Environment variables with typical values using the Korn-shell syntax are:

   ```
   export ADFHOME $HOME/fix2006
   export ADFBIN $ADFHOME/bin
   export ADFRESOURCES $ADFHOME/atomicdata
   export SCMLICENSE $ADFHOME/license
   export NSCM 64 (The default number of parallel processes to use)
   export SCM_TMPDIR $ADFHOME/tmp
   export SCM_USETMPDIR yes
   export SCM_IOBUFFERSIZE=512
   ```

2. Run gunzip and untar fix2006.tar.gz.
3. Execute the following command:

   ```
   cd $ADFHOME
   ```

[1] http://www.scm.com/

[2] Qiyou Jiang, *Amsterdam Density Functional (ADF) Benchmarking on IBM Platforms*, REDP-4128, which is available at: http://www.redbooks.ibm.com/abstracts/redp4128.html/

4. This step is the most important. Under the Install directory, there is a subdirectory for each platform. Each subdirectory includes some platform-dependent files such as TclTk, Makeflags, and so on. You can either create a new subdirectory for Linux on POWER (then modify the configuration file accordingly) or, to simplify the porting efforts, just adapt an existing subdirectory, such as ibm64_xlf for Linux on POWER.

   The main porting work involved here is as follows:

   a. Rebuild the TclTk package on Linux on POWER. Then, create a zipped tar file tcltk.tgz that includes the binaries, header files, and libraries of TclTk. Place the zipped tar file in the Linux on POWER subdirectory that you created.

   b. Make the following modifications to Makeflags.mpi:

      i. Use `mpfort` instead of `mpif77` or `mpif90`.
      ii. Use `mpcc` instead of `mpicc`.
      iii. Add `-L/usr/X11R6/lib64 -lX11 -lresolv -ldl` to LIBEXTRA.

      > **Note:** You need to link `libresolv.a` and `libdl.a` from /usr/lib on Linux on POWER. They are not required on AIX platforms.

5. Run configuration:

   a. Execute the following command:

   ```
   cd $ADFHOME
   ```

   b. Run Install/configure, by making the following choices:

      i. mpi
      ii. 64 (maximum number of processors for your case)
      iii. ibm64 (bin/yam, bin/scu, settings, and Makeflags are created)

6. To compile the program using your own settings, modify Makeflags (change or add options for compiling or linking) by using the following commands:

   ```
   mkdir tmp
   bin/yam -c -v -t
   ```

   > **Note:** In these commands, the options mean:
   >
   > `-v`: verbose, print compile, archive, and link commands
   > `-c`: clean—remove *all* compilation information and executable
   > `-t`: print timing info

7. To run ADF:

   ```
   $ADFBIN/adf <in>out
   ```

Performance and troubleshooting considerations

This section includes some tips based on our experiences when running the benchmark tests.

XLF10.1

xlf10.1 ends with an Internal compiler error while compiling some ADF files. The problem is that xlf10.1 appears to have issues compiling the following Fortran structure:

```
if(kfexvr(iu,trim(vibstr))) call kfdlvr(iu,trim(vibstr))
```

xlf10.1 does not show any issue if this structure is transformed to another identical form in terms of Fortran syntax:

```
if(kfexvr(iu,trim(vibstr))) then
   call kfdlvr(iu,trim(vibstr))
endif
```

The following files in the ADF package are involved with this structure:

- adf/vibron/RWcgrad.d90
- adf/vibron/ReadExSym.d90
- adf/vibron/transf_h_tm.d90
- adf/vibron/SaveRawDat.d90
- adf/cycle/ReadPot.d90

Apparently, xlf10.1 can compile ADF successfully after this structure is transformed in these files, but even then, the code will terminate abnormally during runtime and this issue appears to be very difficult to track. We conclude it must be an xlf10.1 compiler bug, because xlf9.1 does not have any problem in compiling and running ADF. We have opened a program modification request (PMR) to report this xlf10.1 problem to the IBM compiler team. As of xlf10.1.0.3, a fix has been included that works fine with ADF.

MPITRACE

Frequently, we need to collect MPI elapsed-time measurements for analyzing performance issues or for projection purposes. Normally, all we need to do is to link Bob Walkup's libmpitrace.a when generating binaries for our application and we get the text files mpi_profiles.0, mpi_profile.1, and so on, when we run the code. However, on Linux platforms (including Linux on POWER), our experiments show that this library must be in the last position in the linking order, or else mpi_profiles might not be created. For the ADF package, we add this library in the file Install/yam.tcl after `-lablas`.

Another issue that we need to address that the user might still have an issue obtaining the mpi_profiles (and also the profiling outputs in the next session) for the ADF package. The reason and solution for this issue is as follows. In an execution of ADF, the code first copies all the files from the working directory (defined by SCM_WD) to the temporary directory (defined by SCM_TMPDIR) of each process, and each process in turn uses this temporary directory as its temporary working directory. After the run finishes, the result files are stored back in the original working directory and these temporary directories are immediately deleted. Most importantly, the new working directory is redirected to the root directory.

The mpi_profiles and the profiling outputs are normally created after all calculations are finished. Therefore, ADF tries to write these outputs in the root directory, which is never allowed. A simple solution for this is to change the working directory back to the original directory (defined by SCM_WD) after the run finishes, by modifying libtc/pp/pptempdir.cd like this:

- Original: sprintf (tmpdirnam, "/")
- New: sprintf (tmpdirnam, getenv("SCM_WD"))

Profiling

For better performance, we identify the files with *hot spots* by profiling. We then optimize these files aggressively or even hand-tune the code if necessary.

On AIX, the ADF package has a runtime issue if compiled with **`-pg`**, but appears to be all right with **`-p`**. We can also use **`tprof`** for profiling. On Linux on POWER, the closest tool is **`oprofile`**.

Different input files might give different profiling outputs. Table 1 and Table 2 show the most time-consuming routines that we identified by profiling using the two examples that we list in the "Appendix" on page 10.

Table 1 Profiling outputs for Example 1

Subroutine	%	Source
.__m_bas_MOD_calcbas	24.5	M_bas.f90
.calc_density_block	7.3	Calc_density_block.f90
.cfita	5.2	cfita.f90

Table 2 Profiling outputs for Example 2

Subroutine	%	Source
.__Calcfockmat2	10.4	CalcFockMat2.f90
.__mkorbmod	7.5	mkorb.f90
.calfov	7.2	calfov.f90

In the ADF package, a convenient way to apply a more aggressive optimization strategy to these files is to modify the routine CompilerSrc in the file Install/yam.tcl. Our experiments show that this strategy can lead to 5% to 10% performance gains.

Processor binding

IBM POWER5 processor core supports both Simultaneous Multi-Threading (SMT) and Single Threaded (ST) operation modes. In the SMT mode, one physical processor core may have two processes running. In our experiments, we observed that the ADF package consistently performs better with ST mode for a certain number of tasks. If we use the processor binding strategy to ensure that no more than one process is running in each physical processor core in SMT mode, the ADF package can give similar performance as in the ST mode. The ADF package does not benefit much from processor binding in ST mode. Our experiments show that this processor binding strategy can give the ADF package about 10% performance improvements for many cases in the SMT mode.

On Linux on POWER, processor binding can be done by using the **taskset** command. The easiest way to do this for the ADF package is to add the **taskset** command in the file bin/start.

Large pages and medium pages

POWER5 AIX supports 4 KB page size (small page), 64 KB page size (medium page) and 16 MB page size (large page). In our experiments, the ADF package does not show much difference between using large pages and using medium pages, but large pages and medium pages can give the ADF package 2% to 3% better performance than small pages.

Others

There are many other variations that can have an impact on the performance, such as the environment variables SCM_IOBUFFERSIZE and SCM_VECTORLENGTH,[3] the compiler and its options, the kernel, the file system (gpfs, nfs, local,and so on), the interconnect (HPS, InfiniBand, Myrinet, Ethernet, and so on), and others. We must do a full investigation for each specific case in order to get the best performance.

Performance comparison between Linux and AIX platforms

This section includes the results for our performance comparisons between Linux and AIX, both running on an IBM p575+. We used the development version fix2006 for both operating systems. The code was compiled and run in 64-bit mode. Because the Linux OS does not support large pages and medium pages, we used small pages for all runs on both operating systems.

System configurations

We configured the systems as shown in Table 3.

Table 3 Hardware and software configuration

System	Dual Core p575+ Linux (Linux on POWER)	Dual core p575+ AIX
Processor	POWER5 at 1.9 GHz smt off	POWER5 at 1.9 GHz smt off
Number of processor cores	16	16
Interconnect	IBx4 SDR	HPS
Memory	31 GB	31 GB
Large pages	0 GB	16
Kernel	2.6.5-7.244-pseries64	64 bit
Operating system	Sles9 sp3	AIX 5.3
POE	4.2.0.1	4.2.2.4
Compilers	xlf 9.1.0.0	xlf 9.1.0.6

[3] Jiang, ibid

System	Dual Core p575+ Linux (Linux on POWER)	Dual core p575+ AIX
Performance libraries	► MASS 4.3.0.0 ► ESSL 4.2.3.0	► MASS 4.3.0.1 ► ESSL 4.2.0.0
File systems	► NFS ► Local	► NFS ► GPFS

Case 1

The input file in Example 1 in the “Appendix” on page 10 has relatively light I/O activities and we find that the impacts from different file systems are very mild. We use Example 1 to show the effects from different interconnects (also kernels and possibly others) on the two different platforms.

Figure 1 shows the timings and scalabilities for Example 1. Only one node is used for both 8-way and 16-way runs, while two nodes are used for the 32-way run. It appears that the interconnect has some impacts on the performance, since the performance difference between the two systems for a 32-way run is larger than that of 16-way and 8-way runs. In Figure 1, using 32 processors, the Linux job ran in 655 seconds while the AIX job took only 449 seconds.

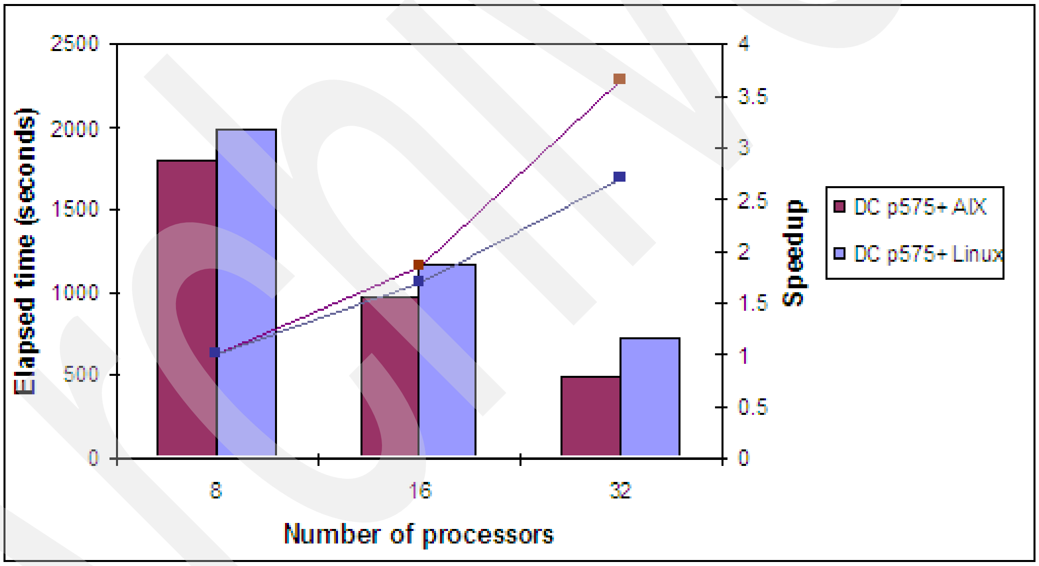

Figure 1 Comparison of timings and scalabilities on two platforms

Case 2

The input file in Example 2 in the “Appendix” on page 10 has much heavier I/O activities than Example 1. Figure 2 demonstrates that the different file systems result in big impacts on performance in this case, especially for the runs that use all processors in a node. In Figure 2, using 32 processors, the NFS job ran in 20855 seconds while the local job took only 16800 seconds.

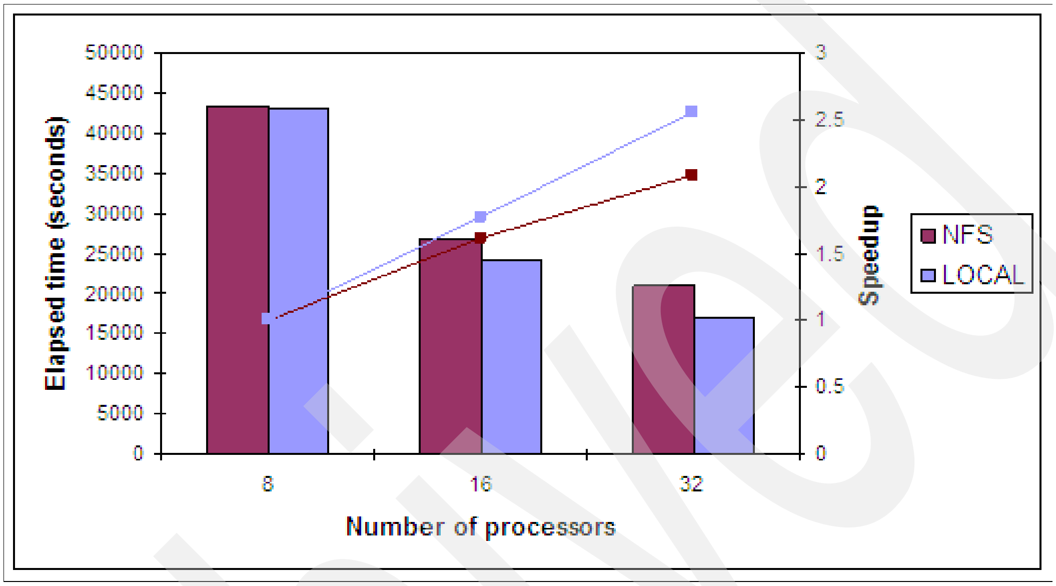

Figure 2 Comparison of timings and scalabilities on Linux on POWER

Summary

This paper describes the procedures to port the ADF package on IBM Linux on POWER platform. The paper also provides tips for troubleshooting and improving performance. The test results show that the ADF package can yield great performance and scalability on IBM platforms under either light or heavy workloads, while the strategies to achieve the best performance are very much case-related.

The author of this redpaper

Qiyou Jiang is a High Performance Computing specialist at the IBM HPC Benchmark Center in Poughkeepsie, New York. Qiyou joined IBM in 2001, and his areas of expertise includes porting, tuning, and benchmarking scientific and technical applications on System p™ and x. He holds a Ph.D. in Chemical Engineering and a Master of Science degree in Computer Science.

Acknowledgements

The author would like to thank the support from the SCM staff, IBM HPC colleagues, system administration team, and management.

References

- *ADF User's Guide (Manual)*, Scientific Computing & Modelling, 2006

  `http://www.scm.com/Doc/Doc2006.01/ADF/`
- *Installation Manual*, Scientific Computing & Modelling, 2006

  `http://www.scm.com/Doc/Doc2006.01/Install/`
- Qiyou Jiang, *Amsterdam Density Functional (ADF) Benchmarking on IBM Platforms*, REDP-4128

  `http://www.redbooks.ibm.com/abstracts/redp4128.html/`

Appendix

The examples that we list here are the input files that we used in our performance comparisons.

Example 1   Input file

```
${ADFBIN}/adf
title
print timingdetail
atoms cartesian
 (24 atom line are omitted)
end

Basis
  Type DZP
  Core none
end

XC
 LDA Xonly
End

symmetry nosym
integration   5.0
linearscaling 8.0

AnalyticalFreq
End

endinput
eor
```

Example 2   Input file

```
$ADFBIN/adf <<eor
create H file=$ADFRESOURCES/TZP/H

XC
   GGA becke perdew
END
eor

mv TAPE21 t21.H

$ADFBIN/adf <<eor
create V file=$ADFRESOURCES/TZP/V.2p

XC
   GGA becke perdew
END
eor

mv TAPE21 t21.V
```

```
$ADFBIN/adf <<eor
create C file=$ADFRESOURCES/TZP/C.1s

XC
   GGA becke perdew
END
eor

mv TAPE21 t21.C

$ADFBIN/adf <<eor
Title
Fragments
V  t21.V
C  t21.C
H  t21.H
END

Atoms
 (34 atom line are omitted)
End

Unrestricted

Charge 0 4

XC
   GGA becke perdew
END

SCF
  Iterations 1000
End

Integration 5.0

Geometry
   Frequency
End

End Input
eor
```

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document created or updated on February 7, 2007.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo) ™
AIX®
IBM®
POWER™
POWER3™
POWER4™
POWER5™
Redbooks™
System p™

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Mac OS is trademark of Apple Computer Inc. in the United States, other countries, or both.

Opteron is a trademark of Advanced Micro Devices, Inc. in the United States, other countries or both.

Altrix and IRIX are trademarks of Silicon Graphics (SGI) in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.