Analysis Plan for Migration of the Performance Assessment Codes to the Sun Solaris Blade Server Running with Intel Processors

Task 1.4.1.3

Effective Date: June 12, 2012
# Table of Contents

1 Introduction ..................................................................................................................... 3  
2 Objectives ....................................................................................................................... 4  
3 Scope ................................................................................................................................ 4  
4 Deliverables ..................................................................................................................... 4  
5 Tasks and Schedules ........................................................................................................ 4  
6 Infrastructure ................................................................................................................... 5  
   6.1 Hardware .................................................................................................................. 5  
   6.2 Other PA Components/Modules ............................................................................. 5  
   6.3 Software .................................................................................................................. 5  
   6.4 Training ................................................................................................................... 6  
7 Project Team Members .................................................................................................. 6  
8 Project Requirements ....................................................................................................... 6  
9 Software List ................................................................................................................... 7  
10 Approach ......................................................................................................................... 8  
   10.1 Porting of the Codes ............................................................................................... 8  
   10.2 Porting the Test Data .............................................................................................. 9  
   10.3 Providing the Scripts .............................................................................................. 9  
   10.4 Validation Test ....................................................................................................... 10  
      10.4.1 Test Procedure .......................................................................................... 10  
      10.4.2 Inputs and Outputs .................................................................................... 10  
      10.4.3 Acceptance Criteria ................................................................................... 10  
   10.5 Complete the Installation and Checkout Forms ................................................... 11  
   10.6 Provide a Summary Report ................................................................................... 11  
11 Special Considerations .................................................................................................. 12  
12 Applicable Procedures ................................................................................................... 12  
13 References ..................................................................................................................... 12
1 Introduction

This Analysis Plan (AP) is intended to provide an overview of the Migration of the Performance Assessment (PA) Computational Suite from an OpenVMS Alpha Cluster to a Sun Solaris Blade Server running with Intel processors. The primary motivation for the PA migration is diminishing technical and hardware support for OpenVMS and the Alpha servers. OpenVMS is a very stable operating system and has been successfully used in the WIPP Compliance environment. However, technical support of the hardware and OS is diminishing, and is becoming increasingly expensive. Additionally, augmentation of the current capabilities is restricted due to the availability of additional hardware, as well as the cost. Because of these reasons, SNL WIPP has decided to move towards Sun Solaris/Intel based platforms.

Solaris is a UNIX operating system developed by Sun Microsystems, and is now owned by Oracle Corporation after Oracle acquired Sun in January, 2010. The Solaris OS running on Sun hardware was picked as the operating system and hardware configuration used for future PA calculations for many reasons, including the following:

- A UNIX or Linux based operating system is readily usable by new and current employees as these platforms are widely used in academia and industry. Numerous training courses are available for the Solaris OS, as well as the chosen SUN hardware it will reside on in the migrated PA configuration, making it easy for new or inexperienced users of UNIX to become proficient in its use. Moreover, this training can be attended by system administrators new to SUN hardware as a means to become familiar with it. The availability of easily accessible training was a primary consideration in the choice of the Solaris/Sun operating system and hardware configuration.

- The Solaris/Sun software and hardware combination presented a greater degree of customization, providing increased flexibility in the final migrated platform.

- Solaris has attractive diagnostic tools (e.g. DTrace), allowing for faster resolution of software and hardware challenges encountered by future users and administrators of the migrated system.

- Sun has an extensive history developing/optimizing the type of server chosen for the migrated hardware platform. Developing and optimizing these systems has been a primary component of Sun’s business model for a long time.

- A learning curve associated with the use of a new software/hardware combination will likely be encountered regardless of the combination chosen. As a result, eventual expected ease of use, ease of system administration, training availability, and stability of the system once in place were considered to be more important factors to consider than the amount of initial learning curve encountered to users of the new system.

- The choice of a Solaris/Sun software and hardware configuration allows for support for the OS and hardware from one supplier. This is more attractive than trying to get support piecemeal from different vendors.

- The opinions of system administrators outside of Sandia were sought in regard to the “best” software and hardware configuration to choose for the PA migration. The consensus of these discussions was the Solaris/Sun combination would result in the fewest problems after everything was up and running.
2 Objectives

The primary objective of this task is to port and qualify those codes needed to perform performance assessments on the Sun Solaris cluster. In addition, the run control system will be simplified using improved scripting methods and some data will be exported directly to database tables for easier access by the analysts. The upgrade will facilitate the preservation of the current PA capabilities, as well as allow for increased capability in the future.

3 Scope

This AP will direct the migration of the performance assessment codes and supporting libraries from the current VMS Alpha platform to a Solaris platform on Sun hardware with Intel based processors. Therefore this is a compliance decision analysis.

4 Deliverables

The deliverables for this project will occur in a phased approach per the documentation requirements of NP 19-1 (Chavez 2009). The qualification of the codes is prioritized (Section 6.5) according to need. Individuals designated as “run masters” will perform and document as many of the regression tests as they can but will require that the results be reviewed and accepted by the code sponsor. The code sponsors will be required to take responsibility for validation tests where regression is not possible. In all cases the test runs will be made by an authorized user (a “run master”).

5 Tasks and Schedules

There are six main tasks for the migration of PA codes to the new UNIX platform. The performance of the first five of these tasks is controlled by the requirements of NP 19-1 (Chavez 2009).

Task 1: Migrate PA modules to the new UNIX platform. A Change Control form (NP 19-1-9) will be completed for each code. The codes will be modified as needed to run under the new operating system. A few additional changes may be made to 1) fix known bugs and 2) convert variables to double precision, 3) use FORTRAN standard dynamic arrays instead of the C-based memory allocation methods used with the FORTRAN-77 codes on VMS and 4) enhance output by storing some results into database tables. The codes will be compiled by an authorized “run master” and the executable files (“builds”) stored with the source code and run control files in CVS repositories. Production of these qualified builds will start immediately following the acceptance of this AP into record, projected to be before the end of June, 2012. The products of this task will be a Change Control form and an executable file, stored in CVS, for each code that is converted.

Task 2: Convert to ASCII, as needed, results from prior validation tests to allow for comparisons across the platforms and transfer those results to the Solaris platform. The VMS results will be stored in the CVS repositories with their associated code along with the results from the Solaris tests. This task will start immediately following acceptance of the AP. The products of this task will be the test results from the most current VMS validation test installed in CVS.

Task 3: Provide scripts for running the tests. The scripts will consist of Python scripts for managing the use of the CVS repositories and UNIX shell script “wrappers” for the Python scripts. The scripts will be stored in the CVS repository for the associated code.
**Task 4:** Run tests for validation. Similar results are expected for tests of the utility codes, and perhaps the pre- and post-processing codes, but may not be produced by those codes which require numerically intensive calculations involving convergence on solutions of differential equations as these kinds of problems can be sensitive to the design of the hardware and the floating point representation. Even when similar results are expected, regression analyses are not expected to produce exact matches in numerical output because of the differences in the compilers and particularly the floating point hardware. Allowances for differences will be made when comparing numerical results (Section 6.8.3). Results will be stored in the CVS repository of the code. Validation of a code will start following the qualified build of that code. The product of this task will be a Regression Testing Report or a Validation Document for each code. If regression testing was successful for some tests but validation against the criteria of the VVP was needed for others then the Validation Document will also contain the results of the Regression Testing rather than having a separate report for each approach.

**Task 5:** Complete Installation and Checkout forms for each code that is validated.

**Task 6:** A report will be written to document the completion of the migration of the codes to Solaris. This report will summarize the results of the regression and validation tests performed on the codes, including listing the documents produced. Deviations from this AP will also be discussed.

The estimated start date is June 30, 2012. However, Performance Assessments take precedence over the migration tasks. Given the upcoming PAs, the completion date for the migration is estimated to be December 2013.

### 6 Infrastructure

#### 6.1 Hardware

The following hardware and operating systems are used on the cluster and support server systems.

- The servers that run the MySQL database and that are used for code compilation are:
  - Hardware: Oracle SunFire X2270 server
  - Software: SunOS 5.11 11.0 i86pc i386 i86pc

- The computation nodes for the cluster are:
  - Hardware: Oracle Sun X6270 M2 Blade
  - Software: SunOS 5.11 11.0 i86pc i386 i86pc

#### 6.2 Other PA Components/Modules

There are two other components of the PA system involved in the migration. The PA parameter Database (PAPDB) has already been ported to MySQL on Solaris and qualified there (Kirchner and Long, 2012). Parameters used in a performance assessment are stored in the PAPDB. In addition, CVS (Concurrent Version Control) will be used for the file repository on Solaris. The current version of CVS is 1.12.13.

#### 6.3 Software

In addition to the codes to be migrated, the project will also use FORTRAN 90/95, C, C++, CVS, MySQL, Python, Python scripts, UNIX shell scripts and Fortran Flint. A Python module will be
provided for handling and simplifying the extraction, insertion and tagging of files into the CVS repositories.

6.4 Training

Training will be provided to the Technical Staff for Solaris/UNIX. Training on Solaris will be optional. Primary training will be hands on operation of the system. Training will be documented on a Training Record Form (NP 2-1-2).

7 Project Team Members

This project will involve nearly all of the PA staff since it involves nearly all of the PA codes. Code sponsors will have the responsibility of validating their codes, but as much help as possible will be provided by the authorized users (run masters) in setting up and running the tests. Amy Gilkey will be responsible for modifying and compiling all codes. Table 1 lists the personnel likely to participate in the tasks.

<table>
<thead>
<tr>
<th>Area of Responsibility</th>
<th>Team Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNL Management</td>
<td>Janis Trone</td>
</tr>
<tr>
<td>Project Coordinator</td>
<td>Tom Kirchner</td>
</tr>
<tr>
<td>Network Administrator</td>
<td>John Geilow</td>
</tr>
<tr>
<td>Run Masters¹</td>
<td>Amy Gilkey, Tom Kirchner, Jennifer Long</td>
</tr>
<tr>
<td>Analysts</td>
<td>Tom Kirchner, Chris Camphouse, Kris Kuhlman, Dwayne Kicker, Sungtae Kim, Jim Garner</td>
</tr>
<tr>
<td>SCM Coordinator</td>
<td>Jennifer Long</td>
</tr>
<tr>
<td>Database Administrator</td>
<td>John Geilow, Tom Kirchner, Jennifer Long</td>
</tr>
<tr>
<td>Software QA</td>
<td>Shelly Nielsen</td>
</tr>
<tr>
<td>Subject Matter Experts</td>
<td>Code Sponsors</td>
</tr>
</tbody>
</table>

¹ Run masters will be designated by the PA Manager.

There are no special training requirements for this testing. All participants, listed above, are already fully trained and will be functioning within their normal job descriptions.

8 Project Requirements

The PA computational cluster is comprised of a Sun Solaris Blade Server, a Sun Solaris server for the MySQL database, and a Linux file server. The ported codes will be assigned version numbers reflecting minor changes in the software. Validation tests as specified in the VVP for each code will be conducted on at least one node of the computational cluster. When feasible, regression tests will be performed to compare the results from the Solaris platform to those from the VMS platform. Otherwise the results will be compared to the criteria established in the Verification and Validation Plan (VVP) for the code. All documentation requirements specified in NP 19-1 will be met for each code. In general it is expected that changes to the documents will be accomplished using addenda given that the changes in the source code are minor and will not impact the equations used but only impact the input (command line arguments) and output (save to database tables) interfaces of the code, the way dynamic storage for data is provided, and the storage allocated to floating point variables (double versus single precision).
9 Software List

There are several kinds of codes used in PA. The “major” codes, such as BRAGFLO and CCDFGF, include the models and LHS, the Latin hypercube sampling program for parameter values. Many of these codes have pre- and post-processors that are used to set up the inputs and to filter or merge the outputs. There are also utility codes, like ALGEBRACDB and GROPECDB, which are used to examine or modify results or inputs. There are a few single-use utilities, which have very specific uses and which are used infrequently. Libraries are not codes but are collections of object files than can be linked to other object files to produce codes. Generally if a subroutine is used by more than one code its object is stored in a library to avoid redundancy. Libraries do not require separate validation because their subroutines are validated when the code into which they are linked is validated. Libraries will be built first since they are used to build the codes. The next priority is to build a set of utilities that are needed to prepare inputs for the codes (ALGEBRACDB, GENMESH, ICSET, MATSET, RELATE) or to merge or modify (SUMMARIZE) or examine (GROPECDB) their outputs. Many of these codes are needed because the output or transfer files that are used to couple the system together are unformatted (binary) files and hence not readable or editable directly. The major codes will be built next as requested by the sponsors who are responsible for their validation or the project coordinator. Pre- and post-processor codes for an application will, in general, be compiled and tested with the major code that they support. The plotting utilities are used to examine code output. They are not required for a PA and have been replaced by commercial-off-the-shelf (COTS, e.g. Sigma Plot) software in many cases. Finally there are codes which may not be needed, either because they will be replaced with COTS software (e.g. STEPWISE, PCCSRC), integrated into pre- or post-processors (LHSEDIT, VTRAN) or represent codes or versions of codes that may not be used in the future (e.g. CCDFGF Version 6.00). A new version of SECOTP2D is being developed on Solaris, so that version may be qualified under this AP should it be completed soon enough. In any case should a revision be made to the VMS version before the migration is completed then it is that version that will be qualified.

Table 2. Codes to be migrated.

<table>
<thead>
<tr>
<th>Program</th>
<th>VMS Version</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Libraries</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAMSUPES_LIB</td>
<td>2.23</td>
<td>library</td>
</tr>
<tr>
<td>CAMCON_LIB</td>
<td>2.21</td>
<td>library</td>
</tr>
<tr>
<td>CAMDAT_LIB</td>
<td>1.26</td>
<td>library</td>
</tr>
<tr>
<td>SDBREAD_LIB</td>
<td>4.00</td>
<td>library</td>
</tr>
<tr>
<td><strong>Utilities</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROPECDB</td>
<td>2.12</td>
<td>utility</td>
</tr>
<tr>
<td>ALGEBRACDB</td>
<td>2.35</td>
<td>utility</td>
</tr>
<tr>
<td>GENMESH</td>
<td>6.08</td>
<td>utility</td>
</tr>
<tr>
<td>ICSET</td>
<td>2.22</td>
<td>utility</td>
</tr>
<tr>
<td>MATSET</td>
<td>9.20</td>
<td>utility</td>
</tr>
<tr>
<td>RELATE</td>
<td>1.43</td>
<td>utility</td>
</tr>
<tr>
<td>SUMMARIZE</td>
<td>3.01</td>
<td>utility</td>
</tr>
<tr>
<td>SCREEN</td>
<td>1.0</td>
<td>utility</td>
</tr>
<tr>
<td><strong>Pre- and post-processors</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PreBRAG</td>
<td>8.00</td>
<td>pre/post</td>
</tr>
<tr>
<td>PostBRAG</td>
<td>4.00A</td>
<td>pre/post</td>
</tr>
<tr>
<td>PreCCDFGF</td>
<td>1.01</td>
<td>pre/post</td>
</tr>
<tr>
<td>PreLHS</td>
<td>2.40</td>
<td>pre/post</td>
</tr>
</tbody>
</table>
10 Approach

10.1 Porting of the Codes

Below are the changes that are recommended to be made to the codes.

- Fetch the source and include files from CMS by class using the /nohistory option (to strip CMS comments).
- Transfer the code to the Solaris systems.
- Rename files according to conventions: use lower-case names, remove the code prefixes, change .for to .f. For example, BF2_BRAGFLO.FOR becomes bragflo.f
- Create a makefile.
- Follow naming conventions for .include (.INC) files and change all INCLUDE statements to reference the renamed INC files.
- Strip the /nolist or /list for all INCLUDE statements.
Switch all REAL variables to DOUBLE PRECISION. Do not implement any changes with INTEGER variables. If any variable is REAL*4, check it quickly to see if there are interface issues, but change it unless there is a reason for *4.

Compiler error “Because of COMMON, the alignment of object is inconsistent with its type” means that character and real data are in the same COMMON block. Put the character data in its own COMMON block or arrange the order to correct alignment issues.

Change STATUS='NEW' to STATUS='REPLACE' (or default ‘UNKNOWN’) in OPEN statements, unless it would be better to abort the code than overwrite the file.

Look for logical names (e.g., BF2_BIN$OUTPUT) used for file names. This is likely to occur in INQUIRE or OPEN statements if the code does not get all its file names from the command line. Set up these codes to read file names from the command line if reasonable or from a file or similar scheme if the number of names is excessive for a command line. Document the command line arguments or logical names as they will be needed for the run script.

Examine all OPEN statements. Remove CARRIAGE_CONTROL or VMS-specific options, if given.

If the code uses its input file to specify file names, look at the parsing routine. If the routine translates strings into upper-case (as the current CAMCON_LIB free-field routines do), make sure the file name is not converted to upper-case.

Change CAMSUPES_LIB dynamic memory to F90 allocated arrays. If the conversion to F90 would require tricky coding, then retaining the CAMSUPES_LIB memory calls may result in more readable code. Convert arrays on an array-by-array basis, and only when the conversion is straightforward. This conversion may require moving the array allocation and adding or deleting subroutine arguments. Dynamic arrays may be declared and shared in a Fortran 90 Module.

Identify calls to MD/MC.
   - For each array, add a declaration (check type),
   - change the MxRSRV to an ALLOCATE (check the size and dimension),
   - change any MxDEL to a DEALLOCATE, and change references to the array.
   - Modify error checking and remove unneeded pointers, etc.
   - Calls to MxLONG, MxFIND, and others will require special handling.

Declare all variables, and force the declaration by adding “IMPLICIT NONE”.

The codes will be stored in the CVS repository and scripts written to check out the source code and compile it. Compilations will be done by an authorized “run master”. Version numbers for the codes will be provided by the SCM Coordinator.

10.2 Porting the Test Data

The input files and results of the most recent validation tests run on the VMS platform for each code will be transferred from VMS to the file server for the Solaris cluster using FTP or other suitable means. In some cases the input files may need to be converted from the VMS binary format to ASCII, transferred, then converted to binary files compatible with the Intel architecture.

10.3 Providing the Scripts

Scripts will be written to run the various tests of the code using Python or as shell scripts. These scripts will be preserved in the CVS repository with the test results.
10.4 Validation Test

This test will run all of the “test cases” identified by consulting the “Requirements Coverage by Test Case” table defined in RD/VVP document for each of the application codes listed in Table 2. Each of the runs will be made on the Solaris 11 OS on the Sun cluster, described in Section 6.3.1. This will be done using the qualified executable compiled and stored in a CVS repository. The objectives of this test are to demonstrate that the code executables will run correctly on the new OS and hardware. An additional purpose of the test is to demonstrate that the test results are equivalent to test results obtained under OpenVMS. Equivalence can be demonstrated in two ways: 1) the test results can be verified using the criteria established in the VVP for the code, or 2) the outputs of the code can be compared to the corresponding outputs from the most recent VMS validation and shown to be reasonably close. Exact matches are not expected, due to the change to double precision floating point variables in the codes, the change in the floating point representation, changes in the compilers and the differences in the CPU architectures. Section 6.8.3 discusses the acceptance criteria.

10.4.1 Test Procedure

The following steps will be performed prior to and during the test:

- Identify the test cases from the VVP for each code.
- Create the test scripts to provide the ability to fetch the code executable at least once and all inputs at runtime, and place any relevant outputs into an appropriate repository controlled by CVS prior to terminating the run script.
- Run the test cases on the Sun/Solaris cluster.
- Compare the output to the criteria specified in the VVP for the code. This could include a comparison of data in output files to results from VMS.
- Evaluate whether the regression test or VVP criteria were met.
- If problems arise that are attributed to the migration to the Sun/Solaris systems, then:
  1. Report problems using the NP 19-1 process.
  2. Identify the problem(s) in the Analysis Package.
  3. Identify the problem(s) to a SCMS Librarian for possible management actions in the repository.
  4. Identity the appropriate remediation.
  5. Retest the affected codes.

10.4.2 Inputs and Outputs

All input files used in validation testing and all output files generated during testing and used to evaluate the tests will be saved in the CVS repository of the code. The comparable output files from VMS will be saved in the CVS repository for the code as well if a regression test was performed successfully.

10.4.3 Acceptance Criteria

There will be some differences between the two sets of output files. The following types of differences are expected and are completely acceptable:

- Differences due to run dates and times
- Differences due to different file names
- Differences due to different directory names
- Differences due to different user names
• Differences due to platform and system version.

Also, reasonable differences in numeric values will be acceptable, since the change in platforms and floating point representations will make such differences inevitable. An examination of the values in the PA Parameter Database (PAPDB) show that as few as 1 significant digit is used to represent some parameters. However, such implied precision may not have been intended and may be the result of the way the data were entered and stored. Three or four significant digits are more common, with some parameters showing many more digits. For this analysis it will be assumed that all parameters for a code have at least four significant digits and therefore the precision on calculated values cannot exceed four significant digits. Thus relative percent differences (RPD) of 1e-4 or smaller should, in general, be classified as insignificant. The RPD is calculated as the difference divided by the mean of the two values, i.e.

\[ RPD = \frac{2(x_2 - x_1)}{x_2 + x_1} \]

The RPD is useful when the values are meaningful. However, the codes can sometimes produce values that are meaningless, such as when the values reflect little more than “numerical noise” representing the result of the accumulation of rounding and other numerical errors over large numbers of computations. Numerical noise is always a problem when performing operations on floating point numbers of greatly different magnitudes. Numerical methods, such as finding solutions to differential equations, have to deal with truncation errors as well, which arise not from the finite number of digits that can be stored in a computer but from approximations made when finding solutions numerically. Although good numerical techniques are designed to minimize truncation errors, they usually cannot be eliminated entirely.

It is difficult to specify a priori an absolute value below which results are basically indistinguishable from zero, because it depends on the relative magnitude of the values in the calculations. For example, the RPD between 1e-30 and 1e-40 is large, but if the value represents a release in EPA units then both of these values are insignificant compared to the 1e-4 lower limit on releases normally considered in an analysis. Determinations of whether differences in small values are insignificant will therefore be specified for each code as needed.

In those cases where a regression analysis seems unlikely to work, the criteria for acceptance will be those specified in the VVP for the code. Such cases can occur when a small numerical difference can cause differences in the path that calculations follow. For example, in CCDFGF a random number is generated to determine whether a drilling event occurs. A value that is slightly less than the test value on VMS but slightly larger on Solaris will produce two radically different paths of execution for a simulation. In such cases the code sponsors will be responsible for verifying that the test cases produce acceptable results.

10.5 Complete the Installation and Checkout Forms

Installation and Checkout Forms will be completed for each code following its validation.

10.6 Provide a Summary Report

The report for this AP will summarize the completion of the code migration effort. It will list the documentation for each ported code, explain any significant issues that arose during the migration, and document any deviations from this AP.
11 Special Considerations

The names of individuals who can serve as “run masters” will be provided by the PA Manager.

12 Applicable Procedures

The qualification of the codes on UNIX will follow the procedures of latest revision of NP 19-1 (Chavez 2009). Regression testing will follow the procedures of SP 19-1 using the criteria for comparison of numeric values outlined in Section 6.8.3. Software configuration management on Solaris will follow the guidance of Kirchner (2005), with the exception that a revised structure for the repositories will be used.

13 References


This work of authorship was prepared as an account of work sponsored by an agency of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so for United States Government purposes. Neither Sandia Corporation, the United States Government, nor any agency thereof, nor any of their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by Sandia Corporation, the United States Government, or any agency thereof. The views and opinions expressed herein do not necessarily state or reflect those of Sandia Corporation, the United States Government or any agency thereof.

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.

Parties are allowed to download copies at no cost for internal use within your organization only provided that any copies made are true and accurate. Copies must include a statement acknowledging Sandia Corporation's authorship of the subject matter.