SOFTWARE REQUIREMENTS DESCRIPTION FOR THE CODE ASHPLUME-P

Version 1.0

Prepared by

Charles B. Connor
Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

September 2000

Approved by:

H. Lawrence McKague, Element Manager
Geology and Geophysics

Date 10/4/00

CONTENTS

Section Page

1 INTRODUCTION 1

2 SOFTWARE REQUIREMENT DESCRIPTION FOR PARALLEL ASHPLUME 2
2.1 SOFTWARE FUNCTION 2
2.2 THEORETICAL BASIS 2
2.3 COMPUTATIONAL APPROACH 3
2.4 DATA FLOW 5
2.4.1 I/O 5
2.5 HARDWARE AND SOFTWARE REQUIREMENTS 6
2.6 GRAPHICS 6
2.7 PROCESSORS 6
2.7.1 Pre-processor 6
2.7.2 Post-processor 6

3 REFERENCES 7

1 INTRODUCTION

This Software Requirements Description (SRD) document describes the requirements for the parallelized version of an existing Ashplume computer code (Jarzemba et al., 1997). The new codes will be called Ashplume-P. Just like the Ashplume, it will allow the user to calculate expected volcanic tephra (ash) accumulation as a conditional probability, but it will use parallel computer processing. This software will be used in the high-level waste (HLW) repository license application review, as described in the Yucca Mountain Review Plan (YMRP).

Igneous activity (IA) has been identified as a Key Technical Issue (KTI) by the U.S. Nuclear Regulatory Commission. A fundamental part of the IA KTI is estimation of how volcanic tephra is dispersed in the environment. See Connor et al. (2000), Jarzemba (1997), and Hill et al. (1998) for reviews of this topic. Dispersion of HLW contaminated volcanic tephra in the environment is currently abstracted in performance assessments (PA) using the Ashplume code. Ashplume-P will not became a part of the TPA code. Rather, Ashplume-P will be used in the IA KTI in order to more fully explore parameter variations, such as eruption conditions, than is practical in a full PA analysis. Parallelization of Ashplume will allow users to obtain many more realizations of potential eruptions during Monte Carlo simulations. Ashplume-P will also incorporate more recent developments in eruption modeling than covered in Ashplume and will use a stratified wind field model. Thus, Ashplume-P will provide a means of reviewing estimates of volcanic eruptions resulting in airborne radionuclide transport, as described in the YMRP.

2 SOFTWARE REQUIREMENT DESCRIPTION FOR PARALLEL ASHPLUME

This SRD briefly outlines the software function; theoretical basis (physical and mathematical model); computational approach; data flow, including I/O; hardware and software requirements; and pre- and post-processors.

2.1 SOFTWARE FUNCTION

The purpose of this software is to provide the user with a comparatively rapid and efficient means of calculating the dispersion of tephra given input on the parameters of volcanic eruptions. Output from the code includes an estimate of the mass of tephra accumulated at a point location due to a volcanic eruption of given magnitude. Output is extended to include multiple points on a grid, which may be contoured by the user to form an isopach map, or simulations involving multiple eruptions, used to analyze probable tephra accumulations stochasitcally.

2.2 THEORETICAL BASIS

Modeling ash dispersion is part of the IA activities in the Airborne Transport ISI and is used in PA to calculate risk for the critical group or over a map area. Ashplume-P will use Suzuki's (1983) model to mimic the thermo-fluid-dynamics of ash dispersion in the atmosphere using the following:

$$X(x,y) = \int_{\rho_{min}}^{\rho_{max}} \int_{0}^{H} \frac{5P(z)f(\rho)Q}{8\pi C t^{5/2}} \exp\left[-\frac{5(x-ut)^2+y^2)}{8Ct^{5/2}}\right] dz d\rho$$

where:

X is the mass of ash and radionuclides accumulated at geographic location x, y, relative to the position of the volcanic vent,

$P(z)$ is a probability density function for diffusion of ash out of the eruption column, treated as a line source extending vertically from the vent to total column height, H,

$f(\rho)$ is a probability density function for grain size, ρ,

Q is the total mass of material erupted,

u is wind speed in the x-direction,

t is diffusion time of ash,

and C is eddy diffusivity.

In the Suzuki model, the erupting column is treated as a line source reaching some maximum height governed by the energy and mass of the eruption. A linear decrease in the upward velocity of particles is assumed, resulting in segregation of ash or ash and waste particles in the ascending column by settling velocity, which is a function of grain size, shape, and density. Tephra particles are removed from the column based on their settling velocity, the decrease in upward velocity of the column as a function of height, and a probability density function which attempts to capture particle diffusion out of the column. Diffusion of the tephra out of the column is modeled for a uniform wind field and is governed by the diffusion-advection equation with vertical settling. Thus, results derived using this model depend heavily on assumptions about the shapes of the distributions $P(z)$ and $f(\rho)$. These distribution functions are empirically derived from analogous eruptions.

The intent of Ashplume-P is to test parameter variations and their impact on patterns of ash dispersion. The code Ashplume-P will vary from Ashplume (Jarzemba et al., 1997) in the following ways:

- A stratified wind field model will be developed to directly account for wind patterns in the YMR.
- Event power will not be used as an input parameter; rather, mass flow and vent velocity will be used.
- Particle size distribution will be characterized by normal distribution in phi units.
- Alternative integration schemes will be used.
- The code will be structured to run in parallel.

Essentially, these modifications either bring the model closer to volcanological literature on this topic (e.g., Sparks et al., 1997), or speed the computation to make it practical to explore the effects of parameter variation more fully. It is anticipated that changes can be introduced into Ashplume after thorough investigation based on the results of modeling with Ashplume-P, should this be necessary.

The introduction of a stratified wind model may be the most important difference between Ashplume and Ashplume-P. A stratified wind field will be incorporated into the model by specifying variation in the wind field as a function of height. A starting height, z, and wind speed and direction, u, are associated with each wind field in the calculation. With a wind field that varies as a function of height, the tephra particles' site of deposition is controlled by the winds that exist between their departure from the column and impact on the ground. Ashplume-P will deal with these multiple wind fields by calculating the average wind vector the tephra particles experience during their fallout:

$$u_{avg} = \frac{\Delta z}{Z} \sum_{k=0}^{N_k} u_k$$

where Z is the height from which the tephra particle is released, N_k is the number of wind strata used in the model, Δz is the thickness of the stratum in which the wind field is assumed to be uniform, u_k is the wind vector in stratum k, and u_{avg} is the average resulting wind vector for particles released at height Z.

2.3 COMPUTATIONAL APPROACH

Successful use of the model for hazard assessment relies on estimation of the input parameters that describe the volcanic site. In practice, the model is calculated numerous times using a range of input parameters that reflect the historical range of similar volcanic activity. Statistical evaluation of the results can then be used for the hazard assessment. This probabilistic approach to compute the likelihood and nature

of future events requires the understanding of previous activity which is quantified by the set of input parameters. Development of these input parameters will not be part of Ashplume-P.

The Message Passing Interface (MPI) and Parallel Virtual Machine (PVM) libraries are free and available tools for creating and running C, C++ or Fortran code in parallel on a cluster of machines. They facilitate the message passing between nodes of the computers and the ability to start and stop processes on every node. Our parallel code will use the PVM software.

The master-slave relationship model will be used in the parallel version of the Suzuki code. In this model, the master process does not do any of the actual computation. The sole job of the master process is to orchestrate the computing that will be done by slave processes. When started on the server node, the master process reads in the variables and the grid dimensions from input files. The master process starts the slave processes on all of the nodes of the cluster. The master decomposes the problem and distributes even portions to the slaves for computation. When the slaves are finished, they send their results to the waiting master and quit. The master then finishes by outputting the returned results on the cluster's server. This is a commonly used static scheduling method for parallel computing. The computational job is divided only once at startup and then distributed for computation. There is only need for communication between the nodes and the server at startup and then again when the computations are complete.

In our parallel implementation, the decomposition of the computational domain will be achieved in two possible ways. The program will either divide the domain of the grid or the sets of input variables. If the grid decomposition is used, the master process divides the grid points where tephra deposition is to be calculated and distributes each slave its portion of the grid. Each slave computes the accumulation on its assigned portion and returns the results to the master process. The other decomposition option is to distribute portions of the input variable file. The variable file is composed of sets of variables. Each variable set is a pair of parameters that describe one eruption for the model. These eruptions, represented by the sets of variables, can be distributed equally to the slaves for computation. Each slave would receive the variables for a number of eruptions. The slaves then simultaneously compute their assigned eruptions on the points of the grid and communicate the results back to the master process.

On a homogenous cluster, the domain is divided and distributed to the identical nodes of the cluster. Each node finishes their computations at the same time. The problem was efficiently computed by the members of the cluster. Our implemented static decomposition does have a drawback. If the cluster contains heterogeneity, our static decomposition allows the faster machines to complete their assigned portion before the others are finished. In this situation, the total time it takes to compute the problem is limited to the slowest time of one of the nodes.

Our cluster is inhomogeneous. There are some minor differences between the nodes of the cluster, but there is a strong difference between the nodes and the server. Even though the server is executing the master and a slave, the slave process still finishes more than twice as fast as slaves running on the nodes. To address this, previous static decomposition and distribution can be altered in two possible ways to more effectively use the cluster. One approach is to divide the domain so that running times are more nearly equal on all slaves. Small initial divisions could be distributed to the slaves. When a slave is finished computing its division, it could return the result and request another chunk of the problem to compute. This dynamic technique allows faster nodes to compute more divisions and so a larger portion of the problem. With this method, there is a price to be paid with the additional increase in inter-node communication. The other alteration utilizes specific PVM functions that allow the nodes to be ranked. The faster nodes, in our case

the server node, could be given a higher rank and then distributed a larger portion of the problem. This technique addresses the speed differences between our server and nodes while avoiding the problem of additional host-slave communication.

2.4 DATA FLOW

Data flow for this code consists of three factors: the basic computational algorithm, the Master node algorithm, and the Slave node algorithm. For each eruption, the deposition at each of the grid points is calculated and the results are returned to the user.

The following basic algorithm will be used:

- Get parameter file
- Get grid dimension file
- AshDeposited = 0.0
- for each x grid point
- for each y grid point
- for each eruption
    - Integrate through the particle sizes using particle step
    - Integrate over the column using the column step
    - AshDeposited = AshDeposited + formula( )

The following describes the data flow in the Master algorithm:

- if Input decomposition
    - Send each Slave the grid and portion of input variables.
- if grid decomposition
    - Send each Slave the portion of grid and the input variables.
- for each slave
    - wait for results
    - output results

The following describes the data flow in the Slave algorithm:

- Receive input data from master
- Compute tephra accumulation
- Return results to master

2.4.1 I/O

The implemented program of the ash deposition model begins by reading two input files. One file contains the parameters used to represent different eruptions. The other file contains a description of the grid that the ash calculation is to be calculated on.

2.5 HARDWARE AND SOFTWARE REQUIREMENTS

Ashplume-P will run on a Beowulf cluster, a networked set of computers dedicated to parallel computations. The Beowulf cluster usually consists of "off the shelf" personal computers connected by Ethernet. It is anticipated that the cluster will consist of 9-12 machines, each containing a 200 mhz Pentium II processor or greater. The machine's operating system will be the SuSE flavor of Linux. Each machine is equipped with an Ethernet card and connected together by Fast Ethernet and a 16 port hub. In addition to the Ethernet hub, there will be a server node which has a > 450 mhz Pentium III processor. The server will contain two Ethernet cards which allow it to communicate with the private Beowulf network and the outside network. The server node (master) functions as both a member of the cluster and as a cluster interface. To avoid the problems associated with transferring files and programs to each of the nodes, the cluster's server will use the Network File System (NFS) to share files between the server and nodes.

The code will be written in C and Parallel Virtual Machine (PVM) language will be resident on the cluster.

2.6 GRAPHICS

No graphics are used in this code.

2.7 PROCESSORS

The calculational code is straightforward, well-defined, and easily programmed. The parallelization of this code will require successful implementation in PVM, but should be easily accomplished.

2.7.1 Pre-processor

In the pre-processor the user will identify the file containing input parameters for the code, and determine whether grid decomposition or input decomposition will be used. The user will define the locations, relative to the volcano, where ash mass will be calculated.

2.7.2 Post-processor

Output from the Ashplume-P code will be in the form of a data file, defined using the file extension ".out". This file will be suitable for plotting using graphical tools such as Generic Mapping Tools and standard scientific graphing packages.

3 REFERENCES

Connor, C.B., B.E. Hill, B. Winfrey, N.M. Franklin, and P.C. La Femina, *Estimation of volcanic hazards related to tephra fallout*. Submitted to Natural Hazards Review, Available HTTP: http://colima.geophysics.swri.edu/cconnor/ash. 2000.

Hill, B.E., C.B. Connor, M.S. Jarzemba, P.C. La Femina, M. Navarro, and W. Strauch. 1995 eruptions of Cerro Negro volcano, Nicaragua, and risk assessment for future eruptions. *Geological Society of America* 110: 1,231–1,241. 1998.

Jarzemba, M.S. Stochastic radionuclide distributions after a basaltic eruption for performance assessments of Yucca Mountain. *Nuclear Technology* 118: 132–141. 1997.

Jarzemba, M.S., P.A. LaPlante, and K.J. Poor. Ashplume Version 1.0—A code for contaminated ash dispersal and deposition. Technical description and user's guide. *CNWRA 97–004 (Revision 1)*. 1997.

Sparks, R.S.J., M.I. Bursik, S.N. Carey, J.G. Gilbert, L.S. Glaze, H. Sigurdsson, and A.W. Woods. *Volcanic Plumes*. John Wiley and Sons. New York: 594pp.

Suzuki, T. A theoretical model for the dispersion of tephra, in Shimozuru, ed., Arc Volcanism. *Terra Scientific*: 95–113. 1983.

Subject: **Re: Introduction of Linux**
Date: Mon, 02 Oct 2000 17:18:34 -0500
From: Budhi Sagar <bsagar@gargol.cnwra.swri.edu>
Organization: CNWRA
To: lmckague@gargol.cnwra.swri.edu
CC: DIRS-MGRS_GROUP <DIRS-MGRS_GROUP@gargol.cnwra.swri.edu>

Larry,

Parallelizing codes like ASHPLUME is very much what other staff at the Center do and that is not unique. What is unique is that it is being parallelized at the Beowulf cluster which does run with linux. My question arose from two concerns (i) this is a code that we shall use in licensing review (per the SRD) and therefore will likely need to provide to the client who neither have the Beowulf cluster nor the linux; neither did we seek the NRC consent for this task; (ii) it does introduce a new OS to the CNWRA, even though from a back door. Besides, "Chuck's students" are CNWRA employees and being Chuck's students doesn't exempt them from the normal way we do business.

My reason for writing the e-mail was to develop some consensus on whether or not to admit linux as one of the OSs for the CNWRA and not have a single staff member spending his and our student employees' time on it.

Larry, to what charge account(s) is the time of these student employees being charged to anyway?

Budhi

Larry McKague wrote:

> Budhi,
>
> You are right the initial Use of linux was on a lap top to be used in the
> field for data reduction and interpretation. In the proposal for the
> Beowulf cluster ((Connor, De La Espriella, Farrell Painter and Green, Nov
> 1999)It was pointed out that "The PC's in a Beowulf cluster run Linux. ...
> Linux provides standard parallel programming tools ,..." Recently, last
> week, Multiflow was put on the Beowulf cluster for Ron Green's use.
> Although not parallized Ron is going to take advantage of the mulitiple
> servers.
>
> Also this is from a recent email from Chuck
>
> "All I can say is that I do not advocate introducing an new
> operating system into the CNWRA. If people need help
> with their computers - it makes sense to maintain a
> single operating system - like NT. My students and I
> are using Linux for very specific tasks - parallelizing the
> ash dispersion code, running geophysical instruments, not
> the same stuff a lot of people are doing. I am sure most people
> don't want the hassle of learning LINUX - it is like

> trying to learn NT if you are used to a MAC."
>
> Larry
>
> Budhi Sagar wrote:
>
> > All,
> >
> > Chuck Connor has brought Linux into the mix of our operating systems.
> > The best I remember is that we started out with a single exception
> > (about two years ago) for LINUX to be used on a portable machine for
> > real time processing of geophysical data. Now I see that it is on the
> > Beowulf cluster. Chuck is specially developing a version of ASHPLUME
> > for the Beouwulf cluster (I just finished reading his SRD). There may
> > be other codes that may be developed for this OS. It seems to me that
> > we need to make a decision whether or not we want to spend our limited
> > resources dealing with this operating system (other than UNIX and NT).
> > The Linux apparently has many 'flavors'; the current one used by Chuck
> > is called SuSE. Your thoughts?
> >
> > Budhi