Mai 1, 2021

README for using the parallel-version (MPI) of PHT3D v2.17 (mpi_fett)

The parallelization in PHT3D v2.17 has now been improved and extended compared to the earlier version by **Aaron McDonough** developed 10 Years ago at CSIRO, Perth. The new version has a better load balancing due to distributing

(1) the cells for the reactions randomly over the model domain
(2) the transported components
(3) the initialization of point sources for every stress-period

over the MPI processes. The speed up with 4 processors is about 2 to 3, with 8 processors 4 to 6 (see Figures at the end of this document). But it depends on the problem. However, we don't have much experience with very large monster models, yet.

On Windows (Win7 and Win10) :

For the windows version it is required to install another software package from Microsoft, i.e., MSMPI, so to say the communicator between the used MPI processes. The software is contained in the folder MSMPI_v8.1 or you can download the software at the official Microsoft download center:

https://www.microsoft.com/en-us/download/details.aspx?id=55494

You need only to download and install MSMpiSetup.exe.

Next, download the pht3dv217_mpi_fett.exe, and for simplicity put that in your model folder.

Start a DOS/CMD window and navigate to that folder.

Run PHT3D with:

mpiexec -n N pht3dv217_mpi_fett.exe Pht3d.nam

where N is the number of processors you want to use on your machine.

During the installation of MSMPI or the first start of the mpiexec, Windoof asks you to grant the program access to your private network. You need to allow that, because MPI is designed to make the communication via the network interface. Otherwise it will not work.

On Linux

The compiled executable pht3dv217_mpi_fett in the source folder may be working out of the box. You need to have OpenMPI installed on your system before you can run it. If it does not run and whinges about missing libraries, you need to re-compile it with the GNU compilers and MPI-wrappers as detailed below. Note that compiling the present source with MPI does not work yet with the INTEL compilers (will be provided later).

For compiling in MPI mode you need to have installed gfortran and gcc, and the OpenMPI developing libraries before can you simply run USE_MPI= yes USE_MPI_WRAPPERS=yes make in the source-folder. It will create the executable pht3dv217_mpi_fett.

For compiling in single, i.e., non-MPI mode you just need to run just make (i.e. without USE_MPI= yes USE_MPI_WRAPPERS=yes) in the source-folder. It will create the executable pht3dv217.

Start the program with mpirun -n N pht3dv217_mpi_fett Pht3d.nam

where N is the number of processors you want to use on your machine.

On a HPC Facility

On a HPC it is very similar to Linux. Compiling is the same. You only need to be sure that the gfortran/gcc and OpenMPI modules are loaded. Job execution is different and it depends on particular system used.

The HPC staff will provide you with the information. Running the model, however, needs a batch-script to be started. Typically with job-submission commands, e.g. 'qsub' or 'sbatch'. This means your job goes in a queue and sits there until computation time is available on the system. In this script, the compiler and OpenMPI modules have to be loaded + information about the computation time, memory requirements, nodes and cores to be used and the model execution command have to be specified. Here also the HPC staff need to provide an example script, since, as I said before, the file will be different from system to system.

Further important notes

(I) If you compare the simulation results of the single and the parallel version you may experience little differences. Normally they are not large and within the machine accuracy. However, under certain circumstances the differences can be severe. However, they are typically not caused by the parallelization. Instead it is caused by

(a) a wrong use of PUTS & GETS in the RATE expressions. PUTS & GETS have to be carefully placed, and not surrounded or skipped by if-statements in the RATE. Because if they get not always updated, the value stays in the buffer (see the PHREEQC manual), is reused and spread over all cells, which then can create wrong results. Sometimes this effect is hidden in the single processor version but becomes visible when running the MPI version. This means if you note differences between the MPI version and the single processor version, most likely its a result of puts-statements in the wrong position inside the BASIC script. It is advised that put statements should be place after SAVE MOLES.

(b) the order of the rates under KINETICS. Under the KINETICS Block, due to the same reasons as in (a), i.e., a cell reads a in value from a different cell. Thus a rate that is providing values with PUT should be located before the rate that receives it with GET. One can alleviate this problem to some degree with small reaction time steps, but not always. Also, because of the pht3d_ph.dat pre-defined required order of kinetic mobile and immobile species and minerals it is not always possible.

Therefore, a different PUT assignment scheme is required so that each cell has its own unique set of storage numbers that is used in the PUT & GET:

PUT (variable_name, n*N+CELL_NO)

GET (n*N+CELL_NO)

where n is the storage number, N ≥ total number of grid cells (nlay*nrow*ncol), CELL_NO is a intrinsic PHREEQC basic variable holding the actual cell/grid number.

N can be very high, I have tested it with N = 100.000.000, so don't worry if you have a large 3D model with 100 million grid cells :)

This approach works !

(c) the advection-schemes that use Methods of Characteristics, as often a random placement of particles is used in every iteration step. The differences between single and MPI version due to that are typically very low and of no relevance. If needed, switching off the parallelization with respect to the transported components will make this effect disappear. To do so, you need to pass the environment variable COMP_DEC set to FALSE during execution:

Windows: mpiexec -n N -env COMP_DEC FALSE pht3dv217_mpi_fett.exe Pht3d.nam

Linux: mpirun -n N -x COMP_DEC=FALSE pht3dv217_mpi_fett Pht3d.nam

(II) If you run the MPI version, please check and compare from time to time with the single processor version, to be on the save side.

(III) On Linux I strongly recommend to run MODFLOW prior PHT3D that a new mt3d link file is produced on that machine. DO NOT transfer the mt3d link file from a windows machine to the linux box as it may cause problems.

To put a run in background and redirect the standard output to a file, e.g., 'out' under Linux, you can run the command as follows:

mpirun -n N pht3dv217_mpi_fett Pht3d.nam > out 2>&1 &

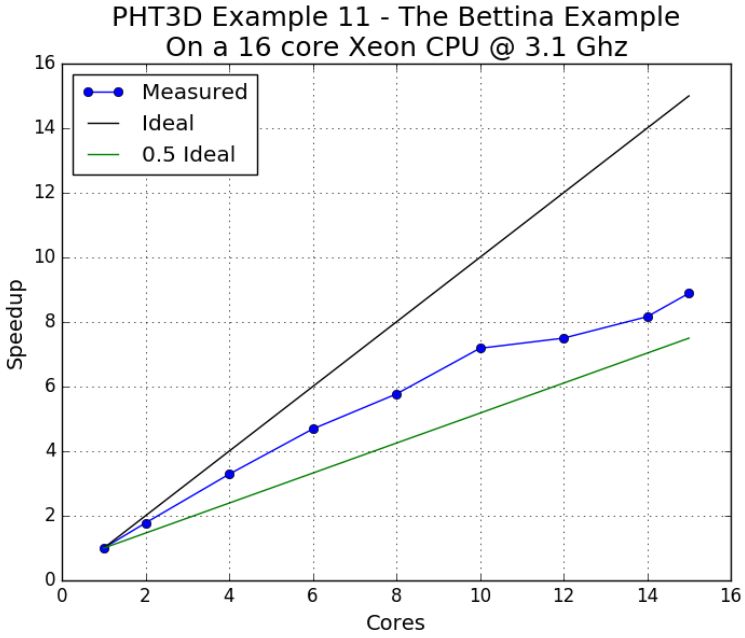
Then you can log out without interrupting the simulation.

(IV) The original MPI implementation, i.e. layer, row, or column domain decomposition by **Aaron McDonough** can be still used via the environment variable PHT3D_DECOMP (= layer or row or column).

Happy computing !
-Janek

Example 11 from PHT3Dv2 Manual

* 99 columns
* 68 layers
* 42 compounds
* 68 WEL-boundaries and 1 stress period @ 120 reaction steps



Dynamod - Reactive model of subterranean estuary

(http://www.staff.uni-oldenburg.de/janek.greskowiak/Dynamod_redox.html)

* 350 columns
* 78 layers
* 7 components
* 300 GHB-boundaries and 3650 stress periods @ 1 reaction step