

# Development and launch of parallel C-DVMH and Fortran-DVMH programs on hybrid supercomputer K-100

## DVM System

December 17, 2016

To compile and launch parallel DVMH-programs on supercomputer K-100 it is necessary:

1. To initiate DVM system version installed on supercomputer K-100. To perform the command:

```
dvminit
```

As a result of given command performing, `dvm` and `usr.par` files necessary for DVMH-program compilation and execution will be copied to the current directory of a user, and also the symbolic reference `~dvm_current` to the installed DVM system version will be created. Demonstrational DVMH-programs are available in the directory `~dvm_current/dvm.sys/demo` (files with `*.cdv` and `*.fdv` extensions).

2. Compilation of DVMH-programs.

- (a) Compilation of C-DVMH program is performed by the command:

```
./dvm c [-noH] [-noCuda] <program name>.<cdvm-ext> ,
```

where `<cdvm-ext>` = `cdv|c|h`

For example,

```
./dvm c jac3d_perf.cdv
```

The result of the command is ready-for-run parallel program in MPI/OpenMP/CUDA model with the name `jac3d_perf`.

```
./dvm c -noCuda jac3d_perf.cdv
```

The result of the command is ready-for-run parallel program in MPI/OpenMP model with the name `jac3d_perf`.

```
./dvm c -noH jac3d_perf.cdv
```

The result of the command is ready-for-run parallel program in MPI model with the name `jac3d_perf`.

- (b) Compilation of Fortran-DVMH program is performed by the command:

```
./dvm f [-noH] [-noCuda] <program name>.<fdvm-ext> ,
```

where `<fdvm-ext>` = `fdv|f|ftn|for|f90|f95|f03`

For example,

```
./dvm f jac3d_perf.fdv
```

The result of the command is ready-for-run parallel program in MPI/OpenMP/CUDA model with the name `jac3d_perf`.

```
./dvm f -noCuda jac3d_perf.fdv
```

The result of the command is ready-for-run parallel program in MPI/OpenMP model with the name `jac3d_perf`.

```
./dvm f -noH jac3d_perf.fdv
```

The result of the command is ready-for-run parallel program in MPI model with the name `jac3d_perf`.

*Notes.*

- i. DVMH-program can have standard extension (for example, `*.f90` or `*.c`).
- ii. For Fortran-programs the record format (free or fixed) is determined automatically by the program extension. For the programs having extensions `*.f90`, `*.f95`, `*.f03`, the free format is used, and for all remaining – the fixed one. It is possible to specify a format using compiler options: `-f90` (free) and `-FI` (fixed). For example,

```
./dvm f -f90 jac.fdv
```

### 3. Launching DVMH-program.

Before DVMH program start it is necessary to set the following environment variables in dvm script (details see in [documentation](#)):

- (a) Maximum runtime of the program. `maxtime` variable:

```
export maxtime=30
```

Default value is 30 minutes. For more long calculations (for example, 1 hour) it is necessary to set:

```
export maxtime=60
```

- (b) A number of MPI processes launched on a node. `DVMH_PPN` variable:

```
export DVMH_PPN='2'
```

- (c) A number of the threads used by each MPI process. `DVMH_NUM_THREADS` variable:

```
export DVMH_NUM_THREADS='4'
```

*Note.* Value of this variable is ignored for the programs compiled with the option `-noH`.

- (d) For each MPI process it is necessary to specify a number of used graphic accelerators. `DVMH_NUM_CUDAS` variable:

```
export DVMH_NUM_CUDAS='3'
```

*Note.* Value of this variable is ignored for the programs compiled with option `-noH` or `-noCuda`.

One node of supercomputer K-100 contains two 6-core processors Intel Xeon X5670 and three graphic accelerators NVidia Fermi C2050.

#### **Recommended configurations:**

When specifying the configuration:

```
export DVMH_PPN='3'  
export DVMH_NUM_THREADS='0'  
export DVMH_NUM_CUDAS='1'
```

three MPI processes will be launched on each compute node of the computer K-100, and each of them will use its own graphic accelerator for computations.

When specifying the configuration:

```
export DVMH_PPN='2'  
export DVMH_NUM_THREADS='6'  
export DVMH_NUM_CUDAS='0'
```

two MPI processes with 6 threads per process will be launched on each compute node of the computer K-100. Graphic accelerators won't be used.

When specifying the configuration:

```
export DVMH_PPN='12'  
export DVMH_NUM_THREADS='1'  
export DVMH_NUM_CUDAS='0'
```

12 MPI processes will be launched on each compute node of the computer K-100. There won't be any job distribution between threads and graphic accelerators.

When specifying the configuration:

```
export DVMH_PPN='1'  
export DVMH_NUM_THREADS='12'  
export DVMH_NUM_CUDAS='3'
```

all computing resources of a node will be used. One MPI process will be launched on each node. This process will use 3 graphic accelerators and 12 threads.

When specifying the configuration:

```
export DVMH_PPN='2'  
export DVMH_NUM_THREADS='6'  
export DVMH_NUM_CUDAS='1'
```

two MPI processes with 6 threads per process will be launched on each compute node of the computer K-100, and each of them will use its own graphic accelerator for computations.

*This configuration is most preferred due to the architecture of supercomputer K-100.*

After setting of environment variables listed above, the DVMH-program can be launched by the command:

```
./dvm run <processor matrix> <program name> [<task parameters>]
```

The parameter <processor matrix> sets rank of a processor grid on which the task will be run (<n1> <n2>... <nK>). During the task execution <n1>\*<n2>\*...\*<nK> MPI processes will be created. For example,

```
./dvm run 2 1 2 jac3d_perf
```

As a result of this command the task which uses four MPI processes will be created. For the configuration:

```
export DVMH_PPN='2'  
export DVMH_NUM_THREADS='0'  
export DVMH_NUM_CUDAS='1'
```

two compute nodes will be provided for the task execution. Two MPI processes will be launched on each of them. Each from the processes will use one graphic accelerator. In total 4 graphic accelerators will be used for computations.

If there are not enough free resources and the program was queued (Task "<task name>.<task number>" queued successfully), it is possible to delete it from queue (mqdel <task name>.<task number>), to get information about a number of free processors (mfree) and to restart the task on smaller number of processors.

Outputs of the task are redirected to the files <task name>.<task number>.1/output and <task name>.<task number>.1/error.

*Note.* <task name> is <program name>.<number of processors>.<task number>. The field <task number> is a number of the program starts on given number of processes in the current directory.

4. Obtaining efficiency characteristics of DVMH-program execution. Command:

```
./dvm pa <task name>.sts.gz+ <name of output file with characteristics>
```

5. The help information about other DVM commands is available as:

```
./dvm help
```

The detailed information about DVM system is on [DVM](#) site.

Questions and notes should be sent to the address: [dvm@keldysh.ru](mailto:dvm@keldysh.ru).

The instruction how to work on supercomputer K-100 computer is available on the [site](#).