

MPIBLAST 1.6.0 – Execution Guidelines for running applications in aggregated environment using ScaleMP’s vSMP Foundation

Overview

MPI BLAST (Basic Local Alignment Search Tool) is an open source parallel version of NCBI BLAST program. It improves the performance of basic biological sequence computation by utilizing distributed computational resources (memory, cpus) through database fragmentation, query segmentation, and parallel I/O. It supports multiple flavors of MPI (MPICH1/2, OpenMPI etc.).

This document describes the best execution practice for MPIBLAST using MPICH2 tuned for vSMP foundation. One of the prerequisite to run MPIBLAST is NCBI Toolbox which is an integral part of the 1.6.0 release. The latest source packages can be downloaded from <http://www.mpiblast.org/>.

Please see the following sections on details for building and running MPIBLAST to extract the maximum performance out of a vSMP Foundation system.

Building MPIBLAST with MPICH2

MPIBLAST 1.6.0 assumes an existing implementation of MPI on a user system. It is preferred to perform build and installation as a root.

As a part of building process, the configuration environment needs to be setup through following command.

```
./configure --prefix=`pwd`/mpiblast --with-mpi=/opt/ScaleMP/mpich2/1.3.1/bin
```

Once configuration is completed without any errors, proceed to build and install the MPIBLAST through following commands:

```
make ncbi  
make  
make install
```

Once MPIBLAST is installed properly, an NCBI environment file needs to be created at `~/ncbirc` to configure the shared and local storage paths that each node will use to access the database. Since the database files can be order of hundreds of megabytes and accessed frequently, it may be a good practice to set it up on RAM file system. Following are the sample contents of this file:

.ncbirc File

[NCBI]

Data=\${BASE}/mpiblast-1.6.0/ncbi/data

[BLAST]

BLASTDB=.

BLASTMAT=\${BASE}/mpiblast-1.6.0/ncbi/data

[mpiBLAST]

Shared=.

Local=.

Running MPIBLAST with MPICH2

This section explains how to execute MPIBLAST in parallel mode.

MPIBLAST requires minimum of 3 MPI processes to execute. It requires the input query file in a FASTA format. It writes the sequencing output in a file specified by `-o` option. It is crucial to use this option since the output can be of order of gigabytes depending on input file. Following environment variables need to be set to run MPIBLAST on a vSMP Foundation System.

Environment variables – MPICH2

While running MPIBLAST with MPICH2 tuned for vSMP Foundation, it is a standard practice to set up following environment variables in order to get maximum performance:

```
export PATH=/opt/ScaleMP/mpich2/1.3.1/bin:/opt/mpiblast/bin:$PATH
```

```
export LD_LIBRARY_PATH=/opt/ScaleMP/mpich2/1.3.1/lib:$LD_LIBRARY_PATH
```

```
export VSMP_PLACEMENT=PACKED
```

For MPICH2 1.3.1, the default is to use pinning, so the parameter below is redundant if using 1.3.1

```
export VSMP_MEM_PIN=yes
```

Following is a sample script to run MPIBLAST with 32 processes. It is recommended to format the database using *mpiformatdb* before invoking *mpiblast*.

Sample run script

```
export BASE=`pwd`
```

```
/usr/bin/time mpiformatdb -N $Num_workers -i uniprot_sprot.fasta -o T 2>&1 | tee log-format.out
```

```
/usr/bin/time mpiexec -n $Num_Processes mpiblast --partition-size=$Num_Partitions --copy-via=none -p blastx -d uniprot_sprot.fasta -i raimondii_contigs.fasta -F F -o uniprot_sprot_out.txt 2>&1 | tee log-blast.out
```

Description of variables used in the script

Num_workers=Number of cores per board

Num_Partitions=Num_workers + 1 (master)

Num_Processes=(Num_Partitions * Number of boards) +1 (master)

Examples

- 8 core / board system with 96 cores (12 Boards)

```
export BASE=`pwd`
```

```
/usr/bin/time mpiformatdb -N 8 -i uniprot_sprot.fasta -o T 2>&1 | tee log-format.out
```

```
/usr/bin/time mpiexec -n 109 mpiblast --partition-size=9 --copy-via=none -p blastx -d  
uniprot_sprot.fasta -i raimondii_contigs.fasta -F F -o uniprotsprot_out.txt 2>&1 | tee log-  
blast.out
```

- 12 core / board system with 96 cores (8 Boards)

```
export BASE=`pwd`
```

```
/usr/bin/time mpiformatdb -N 12 -i uniprot_sprot.fasta -o T 2>&1 | tee log-format.out
```

```
/usr/bin/time mpiexec -n 105 mpiblast --partition-size=13 --copy-via=none -p blastx -d  
uniprot_sprot.fasta -i raimondii_contigs.fasta -F F -o uniprotsprot_out.txt 2>&1 | tee log-  
blast.out
```