

# Hands-on Experience: “Execute your jobs in Ibex cluster”

Nagarajan KATHIRESAN, Ph.D.,  
Computational Scientist,  
KAUST Supercomputing lab, KAUST, KSA  
[nagarajan.kathiresan@kaust.edu.sa](mailto:nagarajan.kathiresan@kaust.edu.sa)

# C Code

```
#include<stdio.h>
int main()
{
    char hostname[1024];
    hostname[1023] = '\0';
    gethostname(hostname, 1023);
    printf("Hello World from Host: %s\n", hostname);
    return 0;
}
```

```
$ module load gcc/6.4.0          // Use appropriate compiler
```

```
$ gcc -o hello.exe hello.c      // Compile the C code
```

```
$ ./hello.exe                   // Execute your program
```

```
Hello World from Host: dbn503-33-r
```

# Understand your job requirements

## ❑ CPU requirement

- Single node (e.g. multi-threaded) or multi-node (e.g. MPI based)
  - ✓ `--nodes` (**Number of nodes required**)
  - ✓ `--cpus-per-task` (**Number of cpus requested per task**)

## ❑ Memory requirement

- ✓ `--mem` (**memory per node, by default 2GB**)
- ✓ `--mem-per-cpu` (**per CPU**)

## ❑ Expected time to complete you job

- ✓ `--time` (**Expected time to complete your job**)
- ✓ `--time=DD-HH:MM:SS` (**Format**)

## ❑ **Validate your SLURM job script before every job submission**

# SLURM Reservation

```
$ sinfo -T
```

RESV_NAME	STATE	START_TIME	END_TIME	DURATION	NODELIST
DMAP	ACTIVE	2018-11-08T11:28:32	2022-01-01T00:00:00	1149-12:31:28	db213-02-2
pathogen	ACTIVE	2018-11-08T11:39:50	2022-01-01T00:00:00	1149-12:20:10	besest514-03
BioCoreLab	ACTIVE	2018-11-12T08:00:52	2020-01-01T00:00:00	414-15:59:08	cn603-03-r, cn603-04-1, cn603-07-1, cn603-09-r, cn603-10-1, cn603-11-1
grimanre_15	ACTIVE	2018-11-15T10:00:00	2019-03-15T10:00:00	120-00:00:00	gpu510-[02,07,12,17]
VCC_WONKA	ACTIVE	2018-12-27T08:52:28	2019-01-17T00:00:00	20-15:07:32	gpu601-[06,08]
H2_test	ACTIVE	2019-01-07T14:52:28	2019-03-15T00:00:00	66-09:07:32	db201-02-[1-8], db201-12-[1-8], db201-22-[1-8], db202-02-[1-8], db202-12-[1-8], db202-22-[1-8], db203-02-[2-8], db203-12-[2-8], db208-02-[1-8], db208-12-[1-8], db208-22-[1-8], db210-02-[1-8], db210-12-[1-8], db210-22-[1-8], db211-02-[1-8], db211-12-[1-8], db211-22-[1-8], db212-02-[1-8], db212-12-[1-8], db212-22-[1-8], db213-02-[1,3-8], db213-12-[1-8], db213-22-[1-8], db214-02-[1-8], db214-12-[1-8], db214-22-[1-8]
IBEX_TRAINING	INACTIVE	2019-01-17T00:00:00	2019-01-17T20:00:00	20:00:00	cn603-14-r, dbn302-01-1, dbn404-32-r, ds506-21, gpu504-37

```
#SBATCH --res=IBEX_TRAINING
```

# Execute your program using SLURM/Scheduler

```
#!/bin/bash
#SBATCH --partition=batch
#SBATCH -J helloworld
#SBATCH -o %x.%J.out
#SBATCH -e %x.%J.err
#SBATCH --time=10:00
#SBATCH --nodes=1
echo "The job "${SLURM_JOB_ID}" is running on
"${SLURM_JOB_NODELIST}" and started at `date`"
echo
"#####"
./hello.exe
echo
"#####"

echo "Job "${SLURM_JOB_ID}" is finished at `date`"
```

# Submit your job

```
$ sbatch ./run_c.sh
Submitted batch job 988408
```

Scheduler will generate Error (\*.err) and Output files (\*.out)

```
-rw-r--r-- 1 kathirn g-kathirn      0 Jan 15 09:51 helloworld.988408.err
-rw-r--r-- 1 kathirn g-kathirn    319 Jan 15 09:51 helloworld.988408.out
```

```
$ cat helloworld.988408.out
```

```
The job 988408 is running on dbn404-08-r and started at Tue Jan 15
09:51:39 +03 2019
```

```
#####
```

```
Hello World from Host: dbn404-08-r
```

```
#####
```

```
Job 988408 is finished at Tue Jan 15 09:51:39 +03 2019
```

# Node topology

```
#SBATCH --nodes=1
```

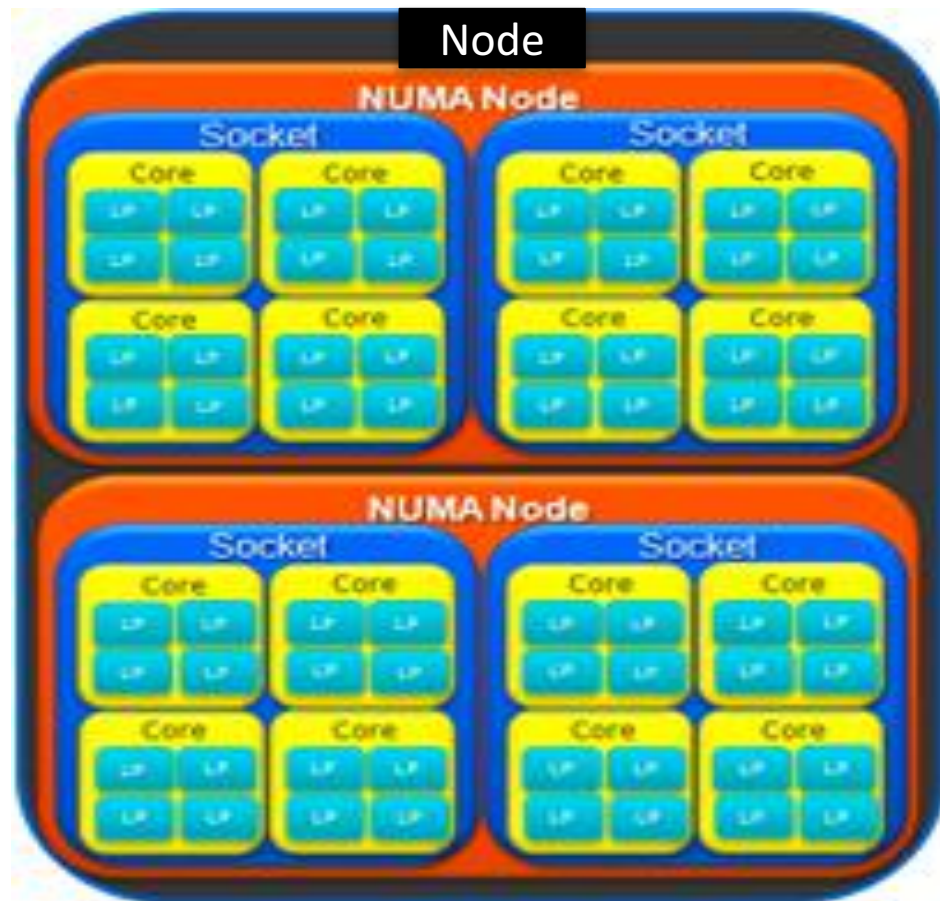
```
#SBATCH --exclusive
```

```
#SBATCH --cpus-per-task=1
```

```
#SBATCH --tasks=1
```

```
#SBATCH --ntasks-per-socket=1
```

```
#SBATCH --ntasks-per-node=1
```



# Output

```
#SBATCH --cpus-per-task=8
```

```
$ cat helloworld.989128.out
```

```
The job 989128 is running on dbn404-08-r and started at Tue Jan 15 10:25:47 +03 2019
```

```
#####
```

```
Hello World from Host: dbn404-08-r
```

```
#####
```

```
Job 989128 is finished at Tue Jan 15 10:25:47 +03 2019
```

```
#SBATCH --ntasks=8
```

```
$ cat helloworld.989130.out
```

```
The job 989130 is running on dbn404-08-r and started at Tue Jan 15 10:28:14 +03 2019
```

```
#####
```

```
Hello World from Host: dbn404-08-r
```

```
#####
```

```
Job 989130 is finished at Tue Jan 15 10:28:14 +03 2019
```

```
#SBATCH --nodes=2
```

```
#SBATCH --cpus-per-task=1
```

```
$ cat helloworld.989164.out
```

```
The job 989164 is running on cn603-22-1,cn603-22-r and started at Tue Jan 15 10:35:29 +03 2019
```

```
#####
```

```
Hello World from Host: cn603-22-1
```

```
#####
```

```
Job 989164 is finished at Tue Jan 15 10:35:29 +03 2019
```



# With srun

```
#SBATCH --tasks=8
```

```
srun ./hello.exe
```

```
$ cat helloworld.990311.out
```

```
The job 990311 is running on dbn404-08-r and started at Tue Jan 15 12:10:35  
+03 2019
```

```
#####
```

```
Hello World from Host: dbn404-08-r
```

```
Hello World from Host: dbn404-08-r
```

```
Hello World from Host: dbn404-08-r
```

```
Hello World from Host: dbn404-08-r
```

```
Hello World from Host: dbn404-08-r
```

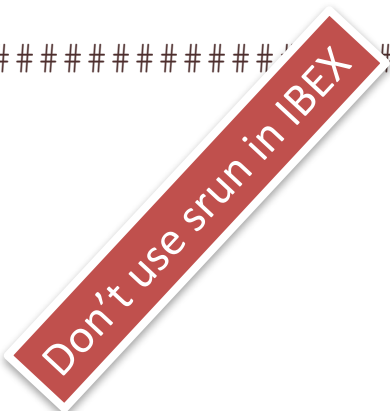
```
Hello World from Host: dbn404-08-r
```

```
Hello World from Host: dbn404-08-r
```

```
Hello World from Host: dbn404-08-r
```

```
#####
```

```
Job 990311 is finished at Tue Jan 15 12:10:35 +03 2019
```



# Hello, World! from R

```
#!/bin/bash
#SBATCH --partition=debug
#SBATCH -J R_World
#SBATCH -o %x.%J.out
#SBATCH -e %x.%J.err
#SBATCH --time=10:00
#SUBMIT --nodes=1
#SUBMIT --cpus-per-task=1

echo "The job "${SLURM_JOB_ID}" is running on "${SLURM_JOB_NODELIST}"
echo "and started at `date`"
echo "#####"

#load the module
module load R/3.5.0/gnu-6.4.0

#Run the R Command
R -e 'print("Hello, World!")'
echo "Job "${SLURM_JOB_ID}" is finished at `date`"
```

# Output

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

```
> print("Hello, World!")
```

```
[1] "Hello, World!"
```

```
>
```

```
>
```

```
Job 989297 is finished at Tue Jan 15 10:44:30 +03 2019
```

# Example: OpenMP

```
#include <stdio.h>
#include <sched.h>
#include <omp.h>
int main()
{
    #pragma omp parallel
    {
        int thread_num = omp_get_thread_num();
        int cpu_num = sched_getcpu();
        printf("Thread %3d is running on CPU %3d\n", thread_num, cpu_num);
    }
    return 0;
}
```

```
$ module load gcc/6.4.0           //Use the required version of compiler
$ gcc -fopenmp hello_omp.c       // Compile the code with OpenMP pragma enabled.
```

# SLURM job script

```
#!/bin/bash
#SBATCH --partition=debug
#SBATCH -J OpenMP
#SBATCH -o %x.%J.out
#SBATCH -e %x.%J.err
#SBATCH --time=10:00
#SBATCH --nodes=1
#SBATCH --cpus-per-task=1

echo "The job "${SLURM_JOB_ID}" is running on "${SLURM_JOB_NODELIST}"
and started at `date`"
echo
"#####"
./a.out
echo
"#####"
echo "Job "${SLURM_JOB_ID}" is finished at `date`"
```

# Output

```
$ sbatch ./run_openmp.sh
```

```
Submitted batch job 991789
```

```
$ cat OpenMP.991789.out
```

```
The job 991789 is running on dbn404-08-r and started at Tue Jan 15  
14:51:24 +03 2019
```

```
#####
```

```
Thread 0 is running on CPU 0
```

```
#####
```

```
Job 991789 is finished at Tue Jan 15 14:51:24 +03 2019
```

# No. of threads

...

...

```
#SBATCH --nodes=1
```

```
#SBATCH --cpus-per-task=1
```

```
export OMP_NUM_THREADS=8
```

```
echo "The job "${SLURM_JOB_ID}" is running on "${SLURM_JOB_NODELIST}"  
and started at `date`"
```

```
echo
```

```
"#####"
```

```
./a.out
```

```
echo
```

```
"#####"
```

```
echo "Job "${SLURM_JOB_ID}" is finished at `date`"
```

# Output

```
$ sbatch ./run_openmp.sh  
Submitted batch job 991828
```

```
The job 991828 is running on dbn404-08-r and started at Tue Jan 15 14:57:13 +03 2019  
#####  
Thread 0 is running on CPU 0  
Thread 7 is running on CPU 0  
Thread 6 is running on CPU 0  
Thread 5 is running on CPU 0  
Thread 4 is running on CPU 0  
Thread 3 is running on CPU 0  
Thread 2 is running on CPU 0  
Thread 1 is running on CPU 0  
#####  
Job 991828 is finished at Tue Jan 15 14:57:13 +03 2019
```

**All the 8 threads are executed in a single core (Core-0)**



# Use more cores!!

```
#SBATCH --nodes=1
```

```
#SBATCH --cpus-per-task=8
```

```
$ cat OpenMP.991939.out
```

```
The job 991939 is running on dbn404-08-r and started at Tue Jan 15 15:03:47 +03 2019
```

```
#####
```

```
Thread 0 is running on CPU 3
```

```
Thread 4 is running on CPU 7
```

```
Thread 2 is running on CPU 5
```

```
Thread 3 is running on CPU 6
```

```
Thread 5 is running on CPU 0
```

```
Thread 6 is running on CPU 1
```

```
Thread 7 is running on CPU 2
```

```
Thread 1 is running on CPU 4
```

```
#####
```

```
Job 991939 is finished at Tue Jan 15 15:03:47 +03 2019
```

**All the 8 threads are parallelly executed across 8 cores (Core:0-7)**

# Hello World from MPI

```
#include <mpi.h>
#include <stdio.h>
int main(int argc, char** argv)
{
    int world_size;
    int world_rank;
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    int core_id;
    MPI_Init(NULL, NULL);
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    MPI_Get_processor_name(processor_name, &name_len);
    core_id = sched_getcpu();
    printf("Hello world from Rank: %d running at Hostname: %s, Core: %d
out of %d processors\n", world_rank, processor_name, core_id, world_size);
    MPI_Finalize();
}
```

# Compile the MPI Code

```
$ module load openmpi/3.1.3/gnu-8.1.0
```

```
$ mpicc -o hello_mpi.exec hello_mpi.c
```

```
#!/bin/bash
```

```
#SBATCH --partition=batch
```

```
#SBATCH -J OpenMPI
```

```
#SBATCH -o %x.%J.out
```

```
#SBATCH -e %x.%J.err
```

```
#SBATCH --time=10:00
```

```
#SBATCH --nodes=1
```

```
#SBATCH --tasks=4
```

```
echo "The job "${SLURM_JOB_ID}" is running on "${SLURM_JOB_NODELIST}"
```

```
echo "and started at `date`"
```

```
echo "#####"
```

```
#load the module
```

```
module load openmpi/3.1.3/gnu-8.1.0
```

```
#then run one of the following:
```

```
mpirun -np 4 ./hello_mpi.exec
```

```
echo "#####"
```

```
echo "Job "${SLURM_JOB_ID}" is finished at `date`"
```

# Output

```
$ sbatch ./run_openMPI.sh  
Submitted batch job 993090
```

```
$ cat OpenMPI.993090.out  
The job 993090 is running on dbn404-08-r  
and started at Tue Jan 15 16:50:41 +03 2019  
#####  
Hello world from Rank: 0 running at Hostname: dbn404-08-r, Core: 1 out of 4 processors  
Hello world from Rank: 2 running at Hostname: dbn404-08-r, Core: 2 out of 4 processors  
Hello world from Rank: 3 running at Hostname: dbn404-08-r, Core: 3 out of 4 processors  
Hello world from Rank: 1 running at Hostname: dbn404-08-r, Core: 0 out of 4 processors  
#####  
Job 993090 is finished at Tue Jan 15 16:50:41 +03 2019
```

# MPI Process distribution across the sockets

```
#SBATCH --tasks=4
```

```
#SBATCH --ntasks-per-core=1
```

```
#SBATCH --ntasks-per-socket=2
```

```
The job 993124 is running on dbn404-08-r  
and started at Tue Jan 15 16:56:19 +03 2019
```

```
#####
```

```
Hello world from Rank: 2 running at Hostname: dbn404-08-r, Core: 1 out of 4 processors
```

```
Hello world from Rank: 1 running at Hostname: dbn404-08-r, Core: 8 out of 4 processors
```

```
Hello world from Rank: 0 running at Hostname: dbn404-08-r, Core: 0 out of 4 processors
```

```
Hello world from Rank: 3 running at Hostname: dbn404-08-r, Core: 9 out of 4 processors
```

```
#####
```

```
Job 993124 is finished at Tue Jan 15 16:56:19 +03 2019
```

# MPI Process distribution across the nodes

```
#SBATCH --tasks=4
#SBATCH --ntasks-per-core=1
#SBATCH --ntasks-per-socket=1
#SBATCH --ntasks-per-node=2
```

```
The job 993127 is running on cn603-15-l,cn603-15-r
and started at Tue Jan 15 17:00:51 +03 2019
```

```
#####
```

```
Hello world from Rank: 0 running at Hostname: cn603-15-l, Core: 0 out of 4 processors
```

```
Hello world from Rank: 1 running at Hostname: cn603-15-l, Core: 20 out of 4 processors
```

```
Hello world from Rank: 2 running at Hostname: cn603-15-r, Core: 16 out of 4 processors
```

```
Hello world from Rank: 3 running at Hostname: cn603-15-r, Core: 32 out of 4 processors
```

```
#####
```

```
Job 993127 is finished at Tue Jan 15 17:00:53 +03 2019
```

# GPU

```
#include<stdio.h>
__global__ void mykernel(void){
}
int main(void) {
    mykernel<<<1,1>>>();
    printf("Hello World!\n");
    return 0;
}
```

```
$ module load cuda/9.0.176
```

```
$ nvcc -o hello.nvidia.exe hello.cu
```

```
$ ./a.out
```

```
Hello World!
```

# Offload the compute to GPUs

```
#include "stdio.h"
__global__ void add(int a, int b, int *c)
{
*c = a + b;
}
int main()
{
    int a,b,c;
    int *dev_c;
    a=3;
    b=4;
    cudaMalloc((void*)&dev_c, sizeof(int));
    add<<<1,1>>>(a,b,dev_c);
    cudaMemcpy(&c, dev_c, sizeof(int), cudaMemcpyDeviceToHost);
    printf("%d + %d is %d\n", a, b, c);
    cudaFree(dev_c);
    return 0;
}
```



# Output

```
$ nvcc add.cu
```

```
$ ./a.out  
3 + 4 is 7
```

# NVIDIA – Profile

Wget [https://cuda-tutorial.readthedocs.io/en/latest/tutorials/tutorial01/solutions/vector\\_add.cu](https://cuda-tutorial.readthedocs.io/en/latest/tutorials/tutorial01/solutions/vector_add.cu)

```
$ nvcc vector.cu
[dgpu105-13 Intel]:/home/kathirn/examples
$ ./a.out
out[0] = 3.000000
PASSED
```

# NVIDIA – ProfileOutput

```
[dgpu105-13 Intel]:/home/kathirn/examples
```

```
$ nvprof ./a.out
```

```
==15348== NVPROF is profiling process 15348, command: ./a.out
```

```
out[0] = 3.000000
```

```
PASSED
```

```
==15348== Profiling application: ./a.out
```

```
==15348== Profiling result:
```

	Type	Time(%)	Time	Calls	Avg	Min	Max	Name
GPU activities:	96.12%	2.42309s	1	2.42309s	2.42309s	2.42309s	2.42309s	vector_add(float*, float*, float*, int)
	1.95%	49.225ms	2	24.613ms	24.224ms	25.001ms		[CUDA memcpy HtoD]
	1.92%	48.500ms	1	48.500ms	48.500ms	48.500ms		[CUDA memcpy DtoH]
API calls:	86.03%	2.52362s	3	841.21ms	24.858ms	2.47334s		cudaMemcpy
	13.62%	399.41ms	3	133.14ms	341.03us	398.71ms		cudaMalloc
	0.23%	6.6237ms	3	2.2079ms	792.05us	2.9346ms		cudaFree
	0.09%	2.5029ms	188	13.313us	453ns	505.70us		cuDeviceGetAttribut
e	0.02%	691.77us	2	345.89us	333.32us	358.45us		cuDeviceTotalMem
	0.02%	446.24us	1	446.24us	446.24us	446.24us		cudaLaunch
	0.01%	221.72us	2	110.86us	108.81us	112.91us		cuDeviceGetName
	0.00%	40.434us	4	10.108us	270ns	29.650us		cudaSetupArgument
	0.00%	9.2390us	3	3.0790us	583ns	7.0260us		cuDeviceGetCount
	0.00%	6.1150us	1	6.1150us	6.1150us	6.1150us		cudaConfigureCall
	0.00%	4.4810us	4	1.1200us	582ns	1.8430us		cuDeviceGet

```
[dgpu105-13 Intel]:/home/kathirn/examples
```



جامعة الملك عبدالله  
للعلوم والتقنية  
King Abdullah University of  
Science and Technology

IBEX



Thank you!

[nagarajan.kathiresan@kaust.edu.sa](mailto:nagarajan.kathiresan@kaust.edu.sa)