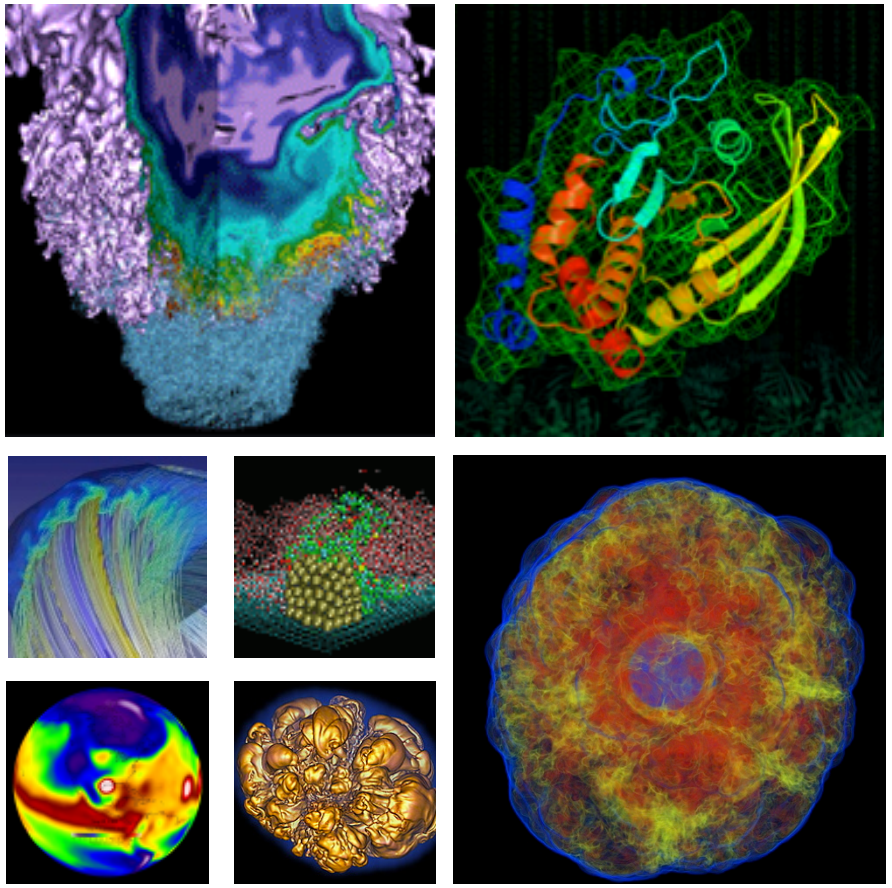
Submitting and Running Jobs



Scott French
NERSC User Services Group

New User Training
February 23, 2015

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

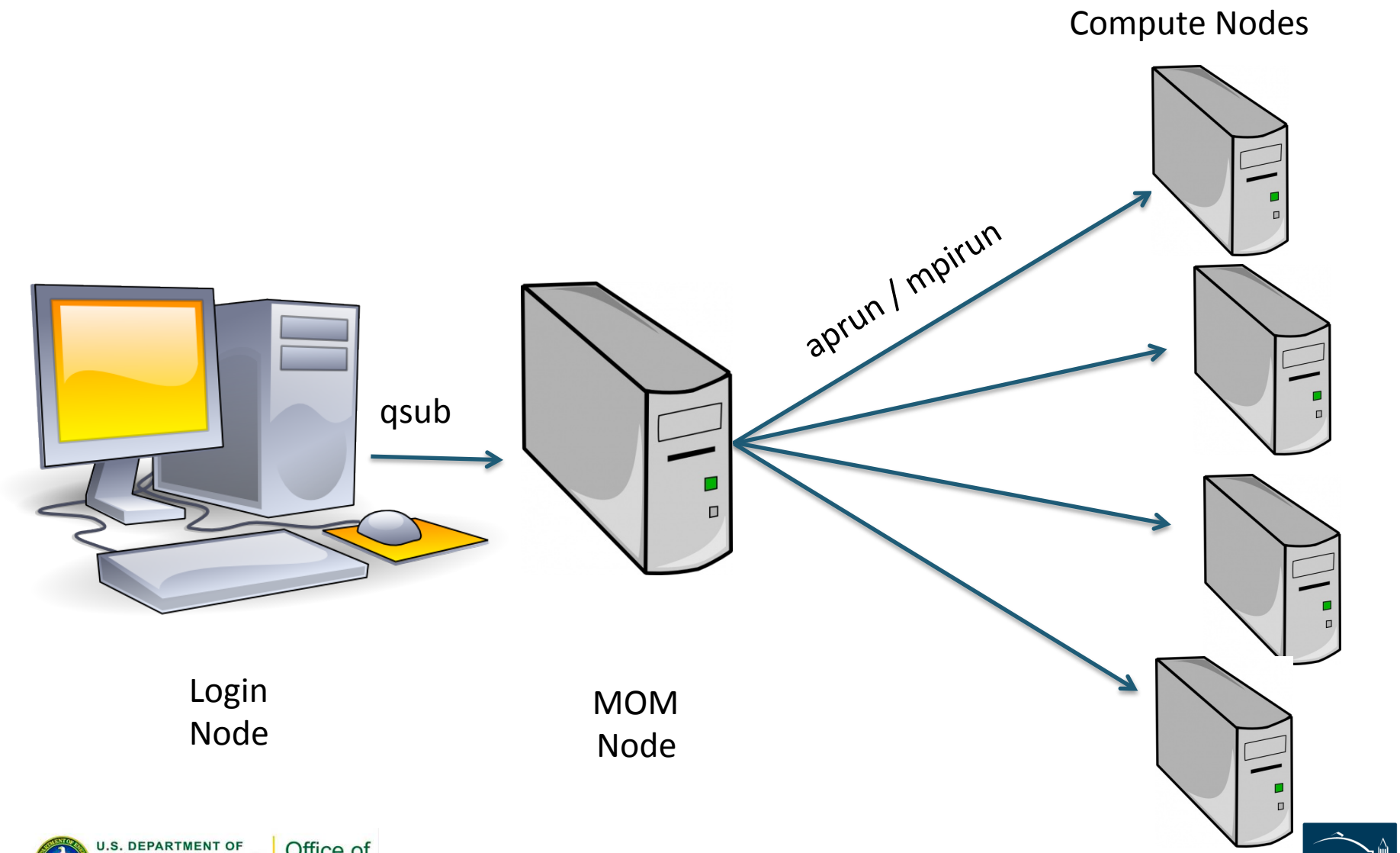
Jobs at NERSC

- **Most jobs are parallel, using 10s to 100,000+ cores**
- **Production runs execute in batch mode**
- **Interactive and debug jobs are supported for up to 30 minutes**
- **Typically run times are a few to 10s of hours.**
  - Each machine has different limits.
  - Limits are necessary because of MTBF and the need to accommodate 5,500 users' jobs
- **Also a number of 'serial' jobs**
  - Typically some kind of pleasantly parallel simulation

Login Nodes and Compute Nodes

- Each supercomputer has 3 types of nodes visible to users
  - Login nodes
  - Compute nodes
  - Application launcher or “MOM” nodes
- Login nodes
  - Edit files, compile codes, run UNIX commands
  - Submit batch jobs
  - Run short, small utilities and applications
- Compute nodes
  - Execute your application; dedicated to your job
  - No direct login access
- Application launcher or “MOM” nodes
  - Execute your batch script commands
  - Carver: “head” compute node
  - Edison / Hopper: shared “service” node (not a compute node)

Launching Parallel Jobs



Launching Parallel Applications

- An “application launcher” executes your code
  - Distributes your executables to all your nodes
  - Starts concurrent execution of N instances of your program
  - Manages execution of your application
  - On Edison / Hopper: the launcher is called “aprun”
  - On Carver: “mpirun”
- Only the application launcher can start your application on compute nodes
- You can’t run the application launcher from login nodes

Submitting Batch Jobs

- To run a job on the compute nodes you must write a "batch script" that contains
  - Batch directives to allow the system to schedule your job
  - An `aprun` or `mpirun` command that launches your parallel executable
- Submit the job to the queuing system with the `qsub` command
  - `% qsub my_batch_script`

Edison - Cray XC30

- 133,824 cores, 5,576 nodes
- “Aries” interconnect
- 2 x 12-core Intel ’Ivy Bridge' 2.4 GHz processors per node
- 24 processor cores per node, 48 with hyperthreading
- 64 GB of memory per node
- 357 TB of aggregate memory
- 2.7 GB memory / core for applications
- /scratch disk quota of 10 TB
- 7.6 PB of /scratch disk
- Choice of full Linux operating system or optimized Linux OS (Cray Linux)
- Intel, Cray, and GNU compilers

Sample Edison Batch Script - MPI

```
#PBS -q debug
#PBS -l mppwidth=96
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
aprun -n 96 ./my_executable
```

Sample Edison Batch Script - MPI

```
#PBS -q debug
#PBS -l mppwidth=96
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
aprun -n 96 ./my_executable
```

Job directives: instructions for the batch system

- Submission queue
- How many compute cores to reserve for your job (/ 24 = # nodes)
- How long to reserve those nodes
- Optional: what to name STDOUT files, what account to charge, whether to notify you by email when your job finishes, etc.

Sample Edison Batch Script - MPI

```
#PBS -q debug
#PBS -l mppwidth=96
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
aprun -n 96 ./my_executable
```

Change from home directory to job submission directory

- Script is initially run from your home directory, **which is not advisable** (as we mention in the filesystem intro)
- You will see much better performance if your job reads / writes from one of the high-performance scratch filesystems

Sample Edison Batch Script - MPI

```
#PBS -q debug
#PBS -l mppwidth=96
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
aprun -n 96 ./my_executable
```

Launches parallel executable on the compute nodes

- Carries over (partial) login environment
- Controls how your executable:
  - maps to processors on the compute nodes (e.g. how many tasks?)
  - accesses the memory on each processor

Sample Edison Batch Script - MPI

```
#PBS -q debug
#PBS -l mppwidth=96
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
aprun -n 96 ./my_executable
```

`mppwidth` is number of compute cores requested for your job

- `mppwidth` = 24 x # of nodes on Edison (and Hopper)
- must be **greater than or equal to** the number of tasks requested (`-n`)

Sample Edison Batch Script - MPI

```
#PBS -q debug
#PBS -l mppwidth=192
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
aprun -n 96 -N 12 ./my_executable
```

`-N` = number of tasks per node

Might do this to get more memory / task

Note that `mppwidth` has changed accordingly

Sample Edison Batch Script - MPI

```
#PBS -q debug
#PBS -l mppwidth=48
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
aprun -n 96 -j 2 ./my_executable
```

`-j` = Turn on hyperthreading

Hybrid OpenMP/MPI

```
#PBS -q regular
#PBS -l mppwidth=96
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
export OMP_NUM_THREADS=6
aprun -n 16 -d 6 -N 4 -S 2 ./hybrid.x
```

A more complex example for mixing MPI and OpenMP:

- 16 tasks (**-n**), 4 on each node (**-N**), 6 OpenMP threads per task (**-d**), assign 2 tasks to each NUMA node (**-S**)

Many more examples on www.nersc.gov

Carver - IBM iDataPlex

- 9,984 compute cores
- 1,202 compute nodes
- 2 quad-core Intel Nehalem 2.67 GHz processors per node
- 8 processor cores per node
- 24 GB of memory per node (48 GB on 80 "fat" nodes)
- 2.5 GB / core for applications (5.5 GB / core on "fat" nodes)
- Interconnect: InfiniBand 4X QDR

- NERSC global /scratch directory quota of 20 TB
- Full Linux operating system
- PGI, GNU, Intel compilers

Sample Carver Batch Script - MPI

```
#PBS -q debug
#PBS –l nodes=16:ppn=8
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
mpirun –n 128 ./myexecutable
```

Number of MPI tasks (similar to `aprun`'s `-n` argument)
Distribution across compute cores primarily controlled via the `ppn` value on the `PBS -l` directive (other ways possible)

Interactive Parallel Jobs

- You can run small parallel jobs interactively for up to 30 minutes (ex. is for Edison)

```
login% qsub -I -l mppwidth=48
[wait for job to start]
mom% cd $PBS_O_WORKDIR
mom% aprun -n 48 ./mycode.x
```

Serial Jobs

- Both Carver and the Crays have a special queue for running serial jobs
  - A single process running on a single core
  - Each serial node can run up to 12 jobs from different users on Carver and 24 jobs on Hopper or Edison

```
#PBS -q serial
#PBS -l walltime=00:10:00
#PBS -N my_job

cd $PBS_O_WORKDIR
./myexecutable
```

Monitoring Your Job

- Once your job is submitted, it will start when resources are available
- You can monitor it with:
  - `qstat –a`
  - `qstat –u username`
  - `showq`
  - `qs`
  - NERSC web site

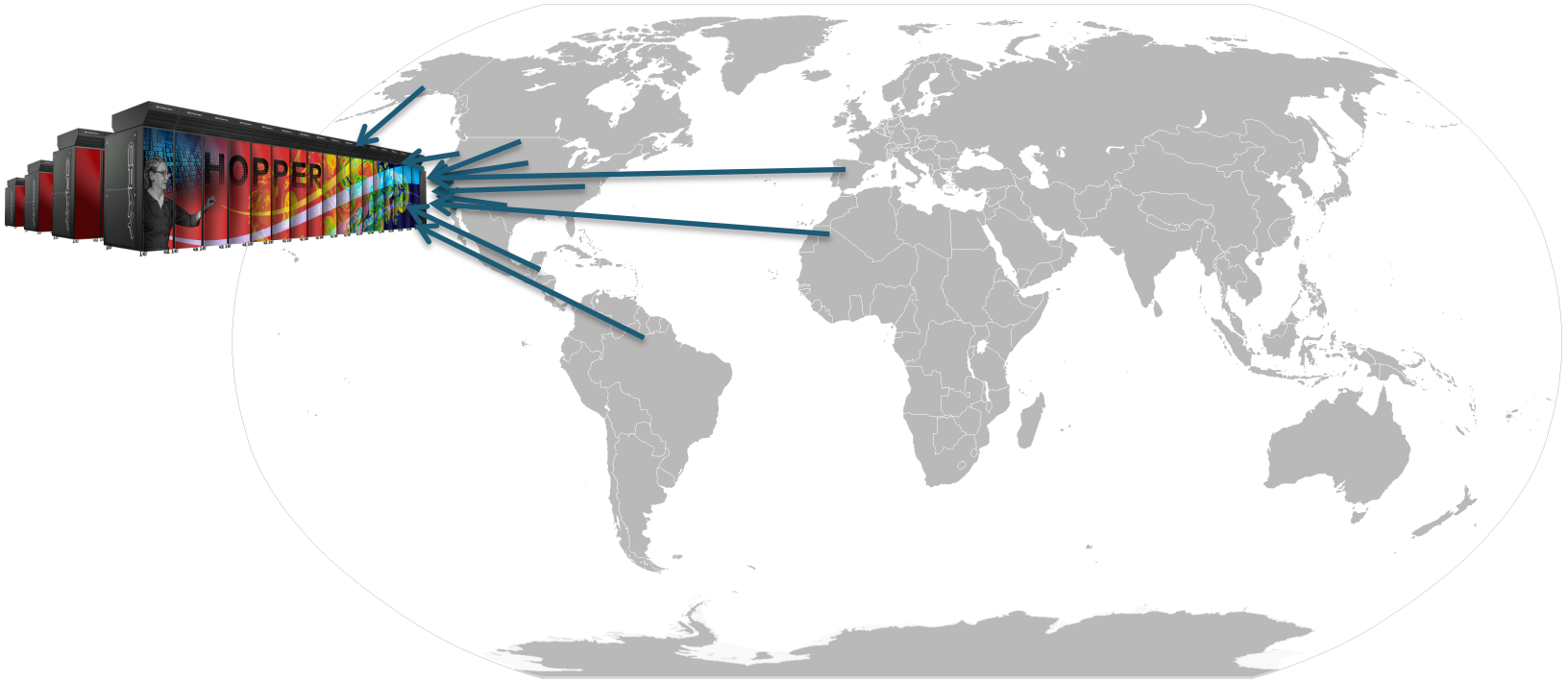
    https://www.nersc.gov/users/live-status/global-queue-look/

    https://www.nersc.gov/users/job-logs-and-analytics/completed-jobs/
  - MyNERSC

    https://my.nersc.gov

I qsub'd my job, but it's not running!

- **You are not alone!**



How do we handle the demand?

That's what the batch queue is for!

source: itu.dk

- **Your jobs will wait until the resources are available for them to run.**
- **Your job's place in the queue is a mix of time and priority, so line jumping is allowed, but it may cost more.**

Job Limits

There are per user, per machine job limits. Here are the limits on Edison as of October 16, 2014.

Specify these queues with
`#PBS -q queue_name`

Not these!

Submit Queue	Execution Queue	Nodes	Physical Cores	Max Wallclock (hours)	Relative Priority	Run Limit	Queued Limit	Queue Charge Factor
debug	debug	1-512	1-12,288	30 mins	2	2	2	1
ccm_int[1]	ccm_int	1-512	1-12,288	30 mins	2	2	2	1
regular	reg_small	1-682	1-16,368	48 hrs	3	24	24	1
	reg_med	683-2048	16,369-49,152	36 hrs	3	8	8	0.6
	reg_big	2049-4096	49,153-98,304	36 hrs	2	2	2	0.6
	reg_xbig	4097-5462	98,305-131,088	12 hrs	2	2	2	0.6
ccm_queue	ccm_queue	1-682	1-16,368	48 hrs	3	16	16	1
premium	premium	1-2048	1-49,152	36	1	1	1	2
low	low	1-682	1-16,368	24	4	8	8	0.5
killable[2]	killable	1-682	1-16,368	48 hrs	3	8	8	1
serial[3]	serial	1	1	48 hrs	-	50	50	1/24

Tips for jobs

- Submit shorter jobs, they are easier to schedule
  - Checkpoint if possible to break up long jobs
  - Short jobs can take advantage of 'backfill' opportunities
  - Run short jobs just before maintenance
- Very important: make sure the wall clock time you request is accurate
  - As noted above, shorter jobs are easier to schedule
  - Many users unnecessarily enter the largest wall clock time possible as a default

How Your Jobs Are Charged

- **Your repository is charged for each node your job was allocated for the entire duration of your job.**
  - The minimum allocatable unit is a node (*except for the serial queues*). Hopper and Edison have 24 cores / node, so your minimum charge is 24*walltime.

MPP hours = (# nodes) * (# cores / node) * (walltime) * (QCF) * (MCF)

  - Example: 96 Edison cores for 1 hour in regular queue
    MPP hours = (4) * (24) * (1 hour) * (1) * (2) = 192 MPP hours
  - Serial jobs are charged with: (walltime) * (MCF)
- **If you have access to multiple repos, pick which one to charge in your batch script**

```
#PBS -A repo_name
```

Charge Factors & Discounts

- **Each machine has a "machine charge factor" (MCF) that multiplies the "raw hours" used**
  - Edison MCF = 2.0
  - Hopper MCF = 1.0
  - Carver MCF = 1.5
- **Each queue has a "queue charge factor" (QCF) and corresponding relative scheduling priorities**
  - Premium QCF = 2.0
  - Low QCF = 0.5
  - Regular (and everything else) QCF = 1.0 (Hopper: 0.8)
- **On Edison:**
  - Jobs requesting more than 682 nodes (reg_med, reg_big, reg_xbig queues) get a 40% discount (QCF = 0.6)

More Information

NERSC Web pages

Hopper:

http://www.nersc.gov/users/computational-systems/hopper/running-jobs/

Edison:

http://www.nersc.gov/users/computational-systems/edison/running-jobs/

Carver:

http://www.nersc.gov/users/computational-systems/carver/running-jobs/

Contact NERSC Consulting:

- Toll-free 800-666-3772
- 510-486-8611, #3
- Email *consult@nersc.gov*.

Thank You

Average Queue Wait Time

	Hours Requested																																																	
Nodes	<1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	48+
1	1.6	5.3	4.1	5.6	5.6	8.5	13	38	58	24	9.8	17	18	110	67	15	31	16	42	6.2	17	10	26	26	29	11	19	20	57	26	42	13	12	27	34	21	46	0.0	18	26	26	17	20	0.0	28	20	21	24	43	29
2	1.7	7.5	8.0	9.6	22	15	25	8.6	21	14	15	13	23	18	22	23	15	14	30	34	34	31	25	30	41	21	35	33	20	19	25	0.0	41	0.0	36	29	47	40	0.1	33	45	0.0	35	17	58	31	33	33	58	76
3	1.4	17	4.8	5.1	4.4	8.0	48	22	11	-2,590.3	15	5.0	19	28	29	19	26	14	23	23	51	0.0	35	37	45	66	89	0.0	32	40	51	0.0	42	0.0	0.0	50	112	44	31	0.0	35	0.0	0.0	0.0	8.3	22	39	37	93	24
4	3.6	11	6.3	14	10.0	12	14	99	16	27	27	29	38	16	24	26	39	20	71	28	48	0.0	14	29	52	60	44	0.0	35	0.0	112	0.0	54	0.0	0.0	69	54	0.0	25	65	148	0.0	0.0	0.0	0.0	0.0	0.0	93	65	44
5	2.2	1.1	7.2	6.0	5.9	9.7	20	13	16	6.5	49	8.6	19	13	38	25	12	28	35	33	58	43	25	50	76	61	0.0	0.0	0.0	23	43	0.0	47	0.0	0.0	43	49	0.0	44	0.0	44	0.0	45	0.0	0.0	0.0	0.0	25	74	32
6	1.1	6.0	3.5	10	6.4	13	11	13	41	11	40	21	19	24	21	23	34	445	32	0.0	43	0.0	35	84	80	0.0	27	0.0	78	0.0	58	34	56	49	28	53	59	0.0	42	0.0	53	0.0	0.0	0.0	0.0	0.0	56	93	102	34
7	0.4	5.4	2.5	7.5	9.9	6.5	14	3.2	15	7.8	38	31	24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	47	0.0	83	16	57	0.0	0.0	0.0	0.0	0.0	37	0.0	47	0.0	0.0	55	19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	104	0.0	0.0	0.0	64	35
8	1.8	16	8.4	36	9.8	16	19	17	16	16	59	41	22	4.7	27	36	37	0.0	33	57	64	0.0	0.0	73	68	18	18	0.0	56	0.0	25	0.0	75	44	28	0.0	56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	86	17	0.0	77	67	35
9	0.4	2.1	2.5	5.8	8.7	15	11	12	14	0.0	32	18	18	0.0	0.0	36	23	0.0	0.0	0.0	130	0.0	0.0	0.0	45	0.0	0.0	0.0	0.0	0.0	236	0.0	0.0	0.0	0.0	0.0	63	0.0	0.0	0.0	51	0.0	0.0	0.0	90	49	72	59	104	16
10	1.8	7.2	8.6	4.8	13	11	12	17	21	18	28	26	23	50	81	72	40	0.0	13	25	80	0.0	72	50	110	60	0.0	0.0	0.0	0.0	62	0.0	0.0	0.0	0.0	36	47	0.0	0.0	0.0	55	0.0	0.0	0.0	113	91	0.0	57	87	76
11	10	26	7.0	6.1	16	13	30	15	15	11	22	27	31	38	27	31	25	39	45	61	75	0.0	0.0	125	56	0.0	41	0.0	0.0	0.0	59	0.0	0.0	0.0	0.0	47	77	41	0.0	0.0	52	0.0	0.0	0.0	0.0	48	58	81	102	45
12	2.4	9.6	10	5.3	23	47	96	16	36	31	254	18	32	9.5	0.0	45	40	0.0	22	53	60	0.0	0.0	113	57	61	0.0	0.0	34	0.0	36	0.0	0.0	0.0	0.0	0.0	138	0.0	0.0	0.0	38	63	0.0	0.0	68	37	63	47	59	42
13	0.4	1.4	7.7	5.0	10	19	13	6.4	8.8	10	80	0.0	18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	48	0.0	0.0	0.0	57	0.0	0.0	0.0	0.0	0.0	50	0.0	0.0	0.0	0.0	0.0	15	0.0	0.0	0.0	64	0.0	0.0	0.0	0.0	0.0	70	0.0	66	0.0
14	1.6	1.1	18	12	8.7	22	17	14	27	27	27	0.0	29	0.0	0.0	0.0	0.0	0.0	105	0.0	97	0.0	0.0	37	54	0.0	0.0	0.0	0.0	0.0	38	0.0	0.0	0.0	0.0	0.0	37	0.0	0.0	0.0	0.0	0.0	53	0.0	151	0.0	0.0	0.0	109	120
15	12	6.7	15	16	7.7	9.9	62	81	14	47	53	37	25	47	86	38	17	38	0.0	0.0	72	0.0	0.0	65	204	54	0.0	48	0.0	0.0	44	0.0	0.0	0.0	0.0	11	93	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	51	0.0	0.0	77	87
16	1.6	1.9	10	27	11	22	39	18	18	30	27	28	22	0.0	124	42	40	43	29	66	58	77	0.0	65	65	79	33	45	0.0	96	68	140	0.0	53	0.0	120	53	0.0	0.0	0.0	105	0.0	0.0	0.0	0.0	0.0	0.0	80	55	32
17-19	1.3	26	19	5.3	10	8.0	20	13	46	21	30	11	43	0.0	104	27	30	0.0	75	0.0	76	0.0	0.0	144	112	132	0.0	0.0	152	63	36	0.0	116	0.0	0.0	64	98	0.0	49	0.0	69	0.0	0.0	0.0	67	0.0	0.0	0.0	77	53
20-23	8.9	8.3	9.3	19	20	19	23	24	29	46	32	25	29	68	53	57	154	0.0	0.0	67	57	76	41	61	66	59	0.0	0.0	0.0	39	75	0.0	202	0.0	0.0	59	61	0.0	0.0	0.0	65	0.0	27	0.0	0.0	0.0	44	41	75	36
24-31	2.6	2.6	4.8	11	15	24	48	15	46	30	19	11	25	166	47	43	47	69	57	67	53	0.0	28	66	125	78	0.0	0.0	86	39	66	0.0	132	41	0.0	102	75	0.0	0.0	0.0	94	0.0	0.0	0.0	0.0	100	40	40	87	104
32-47	1.5	9.2	9.5	26	18	27	58	18	36	51	45	28	37	82	51	54	49	65	86	44	60	128	47	66	101	46	0.0	0.0	0.0	21	89	0.0	134	43	0.0	67	67	0.0	84	49	88	47	0.0	0.0	0.0	62	65	79	88	55
48-63	6.0	12	9.3	15	18	18	32	54	83	43	31	65	40	0.0	47	64	32	0.0	74	74	52	0.0	0.0	71	114	62	0.0	0.0	86	0.0	64	0.0	69	0.0	0.0	108	72	0.0	34	0.0	58	0.0	0.0	0.0	0.0	106	95	77	80	98
64-127	1.6	23	20	21	31	29	36	45	53	33	39	43	52	77	67	59	73	60	86	61	72	45	86	94	78	63	61	0.0	105	119	59	0.0	64	0.0	0.0	127	64	0.0	0.0	42	65	0.0	0.0	0.0	0.0	59	58	0.0	70	101
128-255	4.8	-178.1	21	38	40	50	54	45	77	51	33	149	48	66	130	58	73	73	78	0.0	63	0.0	55	79	101	0.0	0.0	0.0	0.0	41	73	0.0	34	0.0	0.0	175	64	0.0	0.0	0.0	45	50	0.0	0.0	98	0.0	109	102	91	63
256-511	7.8	41	47	38	83	47	131	30	67	51	56	32	44	43	71	51	79	2.5	105	58	95	0.0	0.0	98	86	349	0.0	0.0	0.0	77	95	0.0	0.0	0.0	0.0	0.0	54	0.0	0.0	0.0	68	0.0	0.0	0.0	0.0	0.0	0.0	112	105	0.0
512-1023	12	53	100	91	48	48	65	30	50	94	44	60	56	62	28	66	54	44	36	41	53	0.0	82	32	42	16	0.0	32	42	0.0	87	0.0	144	0.0	36	82	76	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	128	0.0
1024-1535	21	31	32	26	301	29	77	35	56	55	90	39	89	39	41	0.0	33	65	93	0.0	45	0.0	0.0	0.0	45	0.0	0.0	0.0	0.0	0.0	72	0.0	0.0	0.0	0.0	139	208	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1536-2047	24	43	21	11	35	32	66	0.0	48	0.0	47	0.0	35	0.0	0.0	0.0	39	0.0	31	0.0	142	0.0	19	0.0	64	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	110	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2048-3071	30	32	59	41	32	45	50	49	45	0.0	106	0.0	21	46	0.0	70	203	78	144	101	33	0.0	0.0	67	53	31	0.0	0.0	56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3072-4095	25	57	102	0.0	0.0	0.0	0.0	0.0	0.0	130	0.0	0.0	193	0.0	0.0	0.0	0.0	0.0	72	0.0	0.0	0.0	0.0	0.0	31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4096-6143	133	75	75	50	50	0.0	129	60	0.0	14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6144-9531	0.0	55	0.0	29	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0