Argonne NATIONAL LABORATORY

Parallel I/O on Mira

Venkat Vishwanath

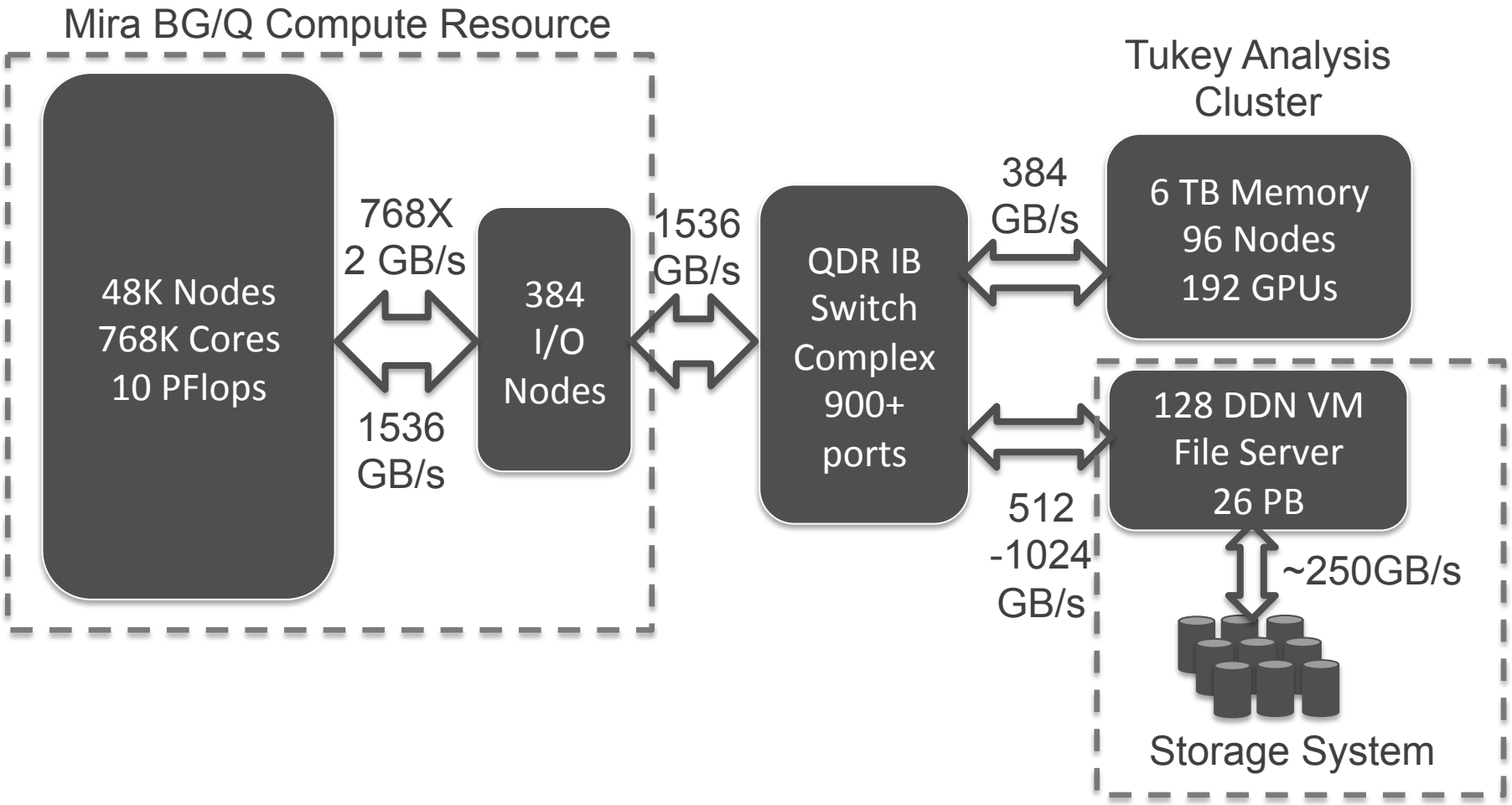
Steve Crusan and Kevin Harms

Argonne National Laboratory

venkat@anl.gov

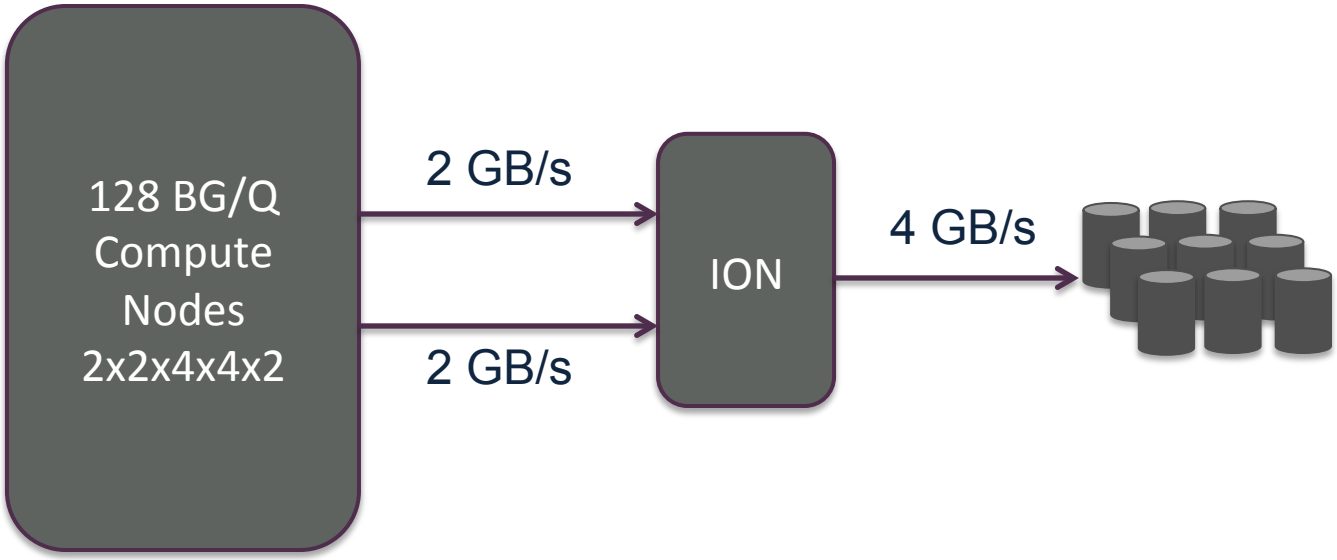
Disclaimer: Mira I/O subsystem and filesystem is still work in progress. The configuration and performance will change, and we will keep users informed of these.

ALCF-2 I/O Infrastructure



Storage System consists of 16 DDN SFA12KE “couplets” with each couplet consisting of 560 x 3TB HDD, 32 x 200GB SSD, 8 Virtual Machines (VM) with two QDR IB ports per VM

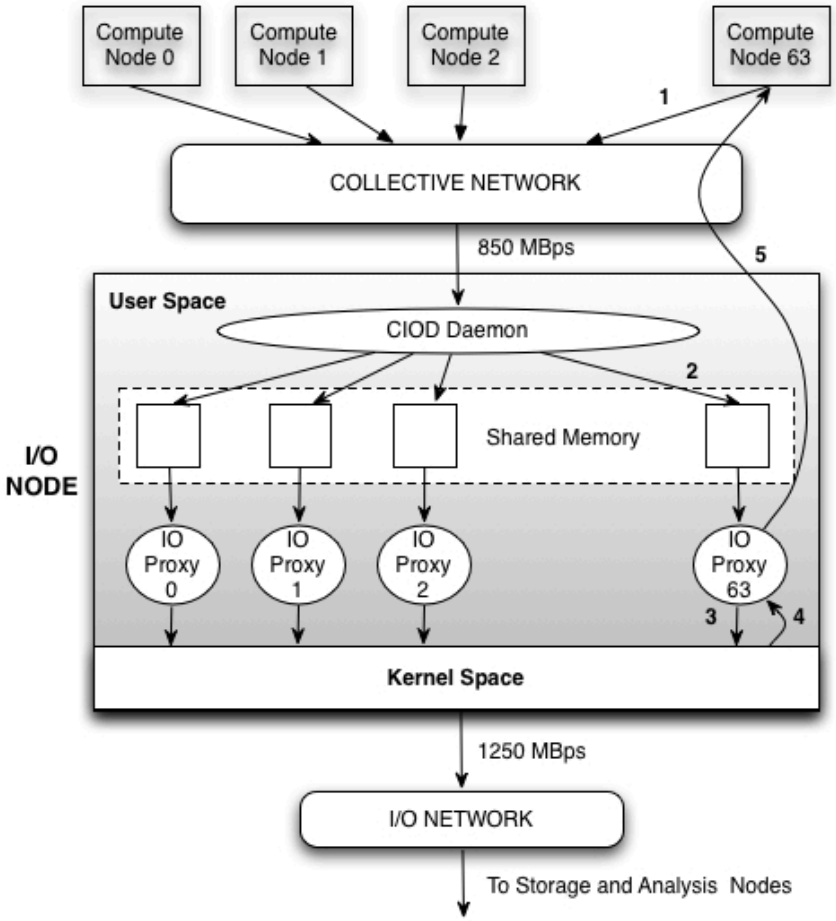
I/O Subsystem of BG/Q- Simplified View



- For every 128 compute nodes, we have two nodes designated as **Bridge** nodes. Each Bridge node has a link to the I/O Node.

- On the ION, we have a QDR Infiniband connected on a PCI-e Gen2 Slot

Control and I/O Daemon (CIOD) - I/O Forwarding mechanism for BG/P and BG/Q



- CIOD is the I/O forwarding infrastructure provided by IBM for the Blue Gene / P
- For each compute node, we have a dedicated I/O proxy process to handle the associated I/O operations
- On BG/Q, we have a **thread based** implementation instead of process-based

Intrepid vs Mira : I/O Comparison

	Intrepid BG/P	Mira BG/Q	Increase
Flops	0.557 PF	10 PF	20X
Cores	160K	768K	~5X
ION / CN ratio	1 / 64	1 / 128	0.5X
IONs per Rack	16	8	0.5X
Total Storage	6 PB	35 PB	6X
I/O Throughput	62GB/s	240 GB/s Peak	4X
User Quota?	None	Quotas enforced!!	-
I/O per Flops (%)	0.01	0.0024	**0.24X**

I/O interfaces and Libraries on Mira

- Interfaces
  - POSIX
  - MPI-IO
- Higher-level Libraries
  - HDF5
  - Netcdf4
  - PnetCDF
- Early I/O Performance on Mira
  - Shared File Write ~70 GB/s
  - Shared File Read ~180 GB/s

I/O Recommendations

- Single shared file vs file per process (MPI rank)
  - Ideally, you should write a few large files
- Use MPI-IO Interfaces
  - Use MPI Collective I/O (eg. MPI_File_write_at_all()) when you are writing small data sizes per rank
  - qsub --env BGLOCKLESSMPIO_F_TYPE=0x47504653
- For non-overlapping writes
  - pass the environment variable at job submission BGLOCKLESSMPIO_F_TYPE=0x47504653 or prefix file with bglockless:/ to disable locking in the ADIO layer
  - Factor of **two** improvement in performance
- For POSIX, instead of lseek and write, use pwrite
  - pwrite(int fd, const void *buf, size_t count, off_t offset);

Mo

- F
- F
- H
- A
- D

```
//   Create and Preallocate the File On Master
if ( 0 == m_rank)
    {
       retval = MPI_File_open(MPI_COMM_SELF, (char *)m_partFileName,
                              MPI_MODE_WRONLY | MPI_MODE_CREATE,
                              MPI_INFO_NULL,&m_fileHandle);
       assert(retval == MPI_SUCCESS);

       // Preallocate File
       MPI_File_set_size(m_fileHandle, m_partFileSize);

       MPI_File_close (&m_fileHandle);
    }

    MPI_Barrier (MPI_COMM_WORLD);

    //   Open the File for Writing
    retval = MPI_File_open(MPI_COMM_WORLD, (char *)m_partFileName,
                           MPI_MODE_WRONLY, MPI_INFO_NULL, &m_fileHandle);

    assert(retval == MPI_SUCCESS);

    MPI_Barrier (MPI_COMM_WORLD);
```

or

or

Tools for I/O Performance Profiling on BG/Q

- HPM
  - Bob Walkup
  - https://www.alcf.anl.gov/user-guides/hpctw
- Darshan
  - Kevin Harms
  - https://www.alcf.anl.gov/user-guides/darshan
- TAU
  - Sameer Shende
  - https://www.alcf.anl.gov/user-guides/tuning-and-analysis-utilities-tau

Pro

- Li
- Jo

Data for MPI rank 0 of 65536
Times and statistics from MPI_Init() to MPI_Finalize().

MPI Routine	#calls	avg. bytes	time(sec)
MPI_Comm_size	63	0.0	0.000
MPI_Comm_rank	2	0.0	0.000
MPI_Isend	4636	262086.0	0.031
MPI_Recv	4666	260650.4	0.215
MPI_Probe	4636	0.0	0.041
MPI_Waitall	1161	0.0	1.591
MPI_Barrier	1277	0.0	99.924
MPI_Gather	28	80.0	0.942
MPI_Allgather	146	14.1	7.821
MPI_Allgatherv	62	32.6	8.719
MPI_Reduce	6	16.7	0.001
MPI_Allreduce	950	23.6	55.296
MPI_File_close	11	0.0	0.231
MPI_File_open	11	0.0	16.976
MPI_File_read_at	1	7559376.0	37.601
MPI_File_write_at_all	20	2426940.0	92.211

total communication time = 174.581 seconds.
total elapsed time = 589.773 seconds.
total MPI-IO time = 147.019 seconds.

Profiling I/O using HPM

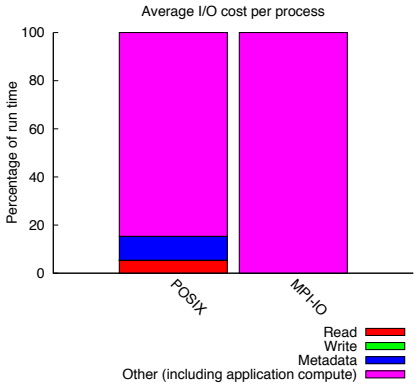
MPI_File_read_at	#calls	avg. bytes	time(sec)
	1	7559376.0	37.601
MPI_File_write_at_all	15	0	10.052
	5	9707760.0	82.159

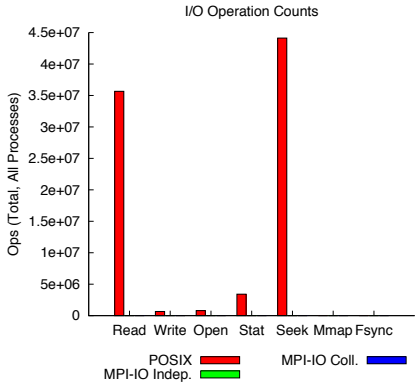
Performance Issue: 15 Collective calls writing **0 bytes** taking 10 secs.

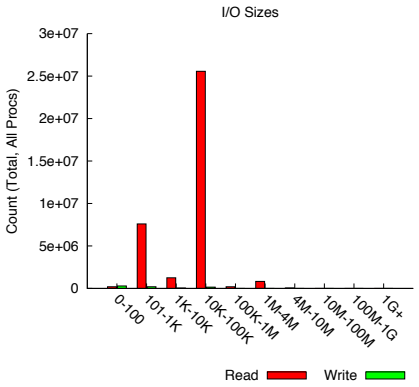
Darshan - An I/O Characterization Tool

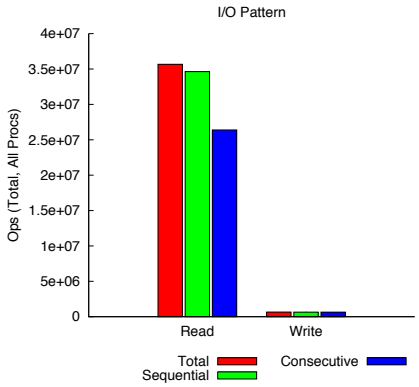
- An open-source tool developed for statistical profiling of I/O
- Designed to be lightweight and low overhead
  - Finite memory allocation for statistics (about 2MB) done during MPI_Init
  - Overhead of 1-2% total to record I/O calls
  - Variation of I/O is typically around 4-5%
  - Darshan does not create detailed function call traces
- No source modifications
  - Uses PMPI interfaces to intercept MPI calls
  - Use ld wrapping to intercept POSIX calls
  - Can use dynamic linking with LD_PRELOAD instead
- Stores results in single compressed log file
- http://www.mcs.anl.gov/darshan

Darshan on BG/Q









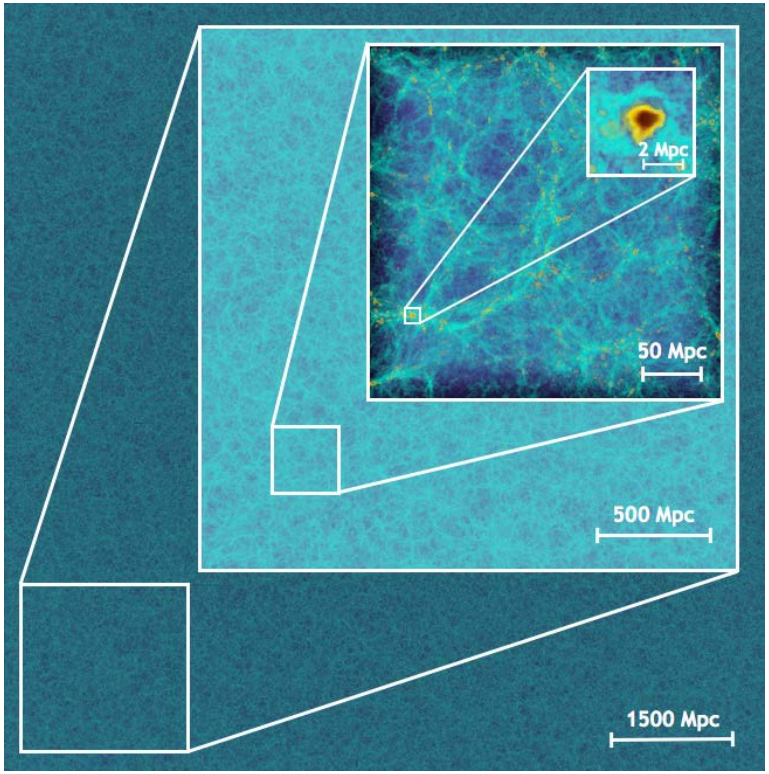
Most Common Access Sizes

access size	count
65536	23066526
800	2378602
304	2172322
400	2029354

- logs:
/gpfs/vesta-fs0/logs/darshan/vesta
/gpfs/mira-fs0/logs/darshan/mira
- bins:
/soft/perftools/darshan/darshan/bin
- wrappers:
/soft/perftools/darshan/darshan/wrappers/<wrapper>
- export PATH=/soft/perftools/darshan/darshan/wrappers/xl:${PATH}

Early Experience Scaling I/O for HACC

- Hybrid/Hardware Accelerated Computational Cosmology code is lead by Salman Habib at Argonne
- Members include Katrin Heitmann, Hal Finkel, Adrian Pope, Vitali Morozov
- Particle-in-Cell code
- Achieved ~14 PF on Sequoia and 7 PF on Mira
- On Mira, HACC uses 1-10 Trillion particles



Zoom-in visualization of the density field illustrating the global spatial dynamic range of the simulation -- approximately a million-to-one

Early Experience Scaling I/O for HACC

- Typical Checkpoint/Restart dataset is **~100-400 TB**
- Analysis output, written more frequently, is **~1TB-10TB**
- Step 1:
  - Default I/O mechanism using single shared file with MPI-IO
  - Achieved ~15 GB/s
- Step 2:
  - File per rank
  - Too many files at 32K Nodes (262K files with 8RPN)
- Step 3:
  - An I/O mechanism writing 1 File per ION
  - Topology-aware I/O to reduce network contention
  - Achieves up to **160 GB/s** (~10 X improvement)
- Written and read **~10 PB** of data on Mira (and counting)

Questions