Tuning Parallel I/O on Blue Waters for Writing 10 Trillion Particles

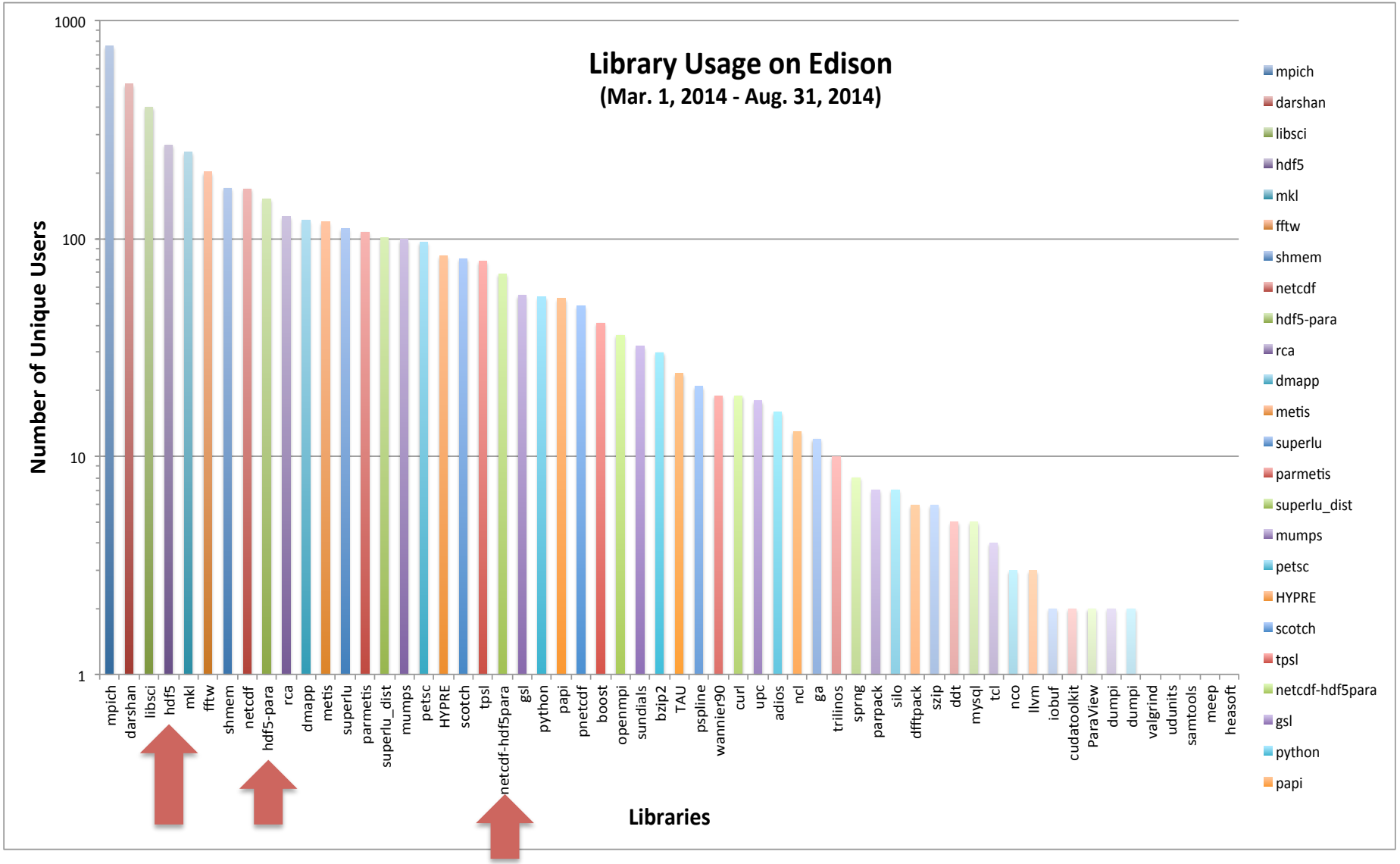
Kalyana Chadalavada

Robert Sisneros, Suren Byna, Quincey Koziol

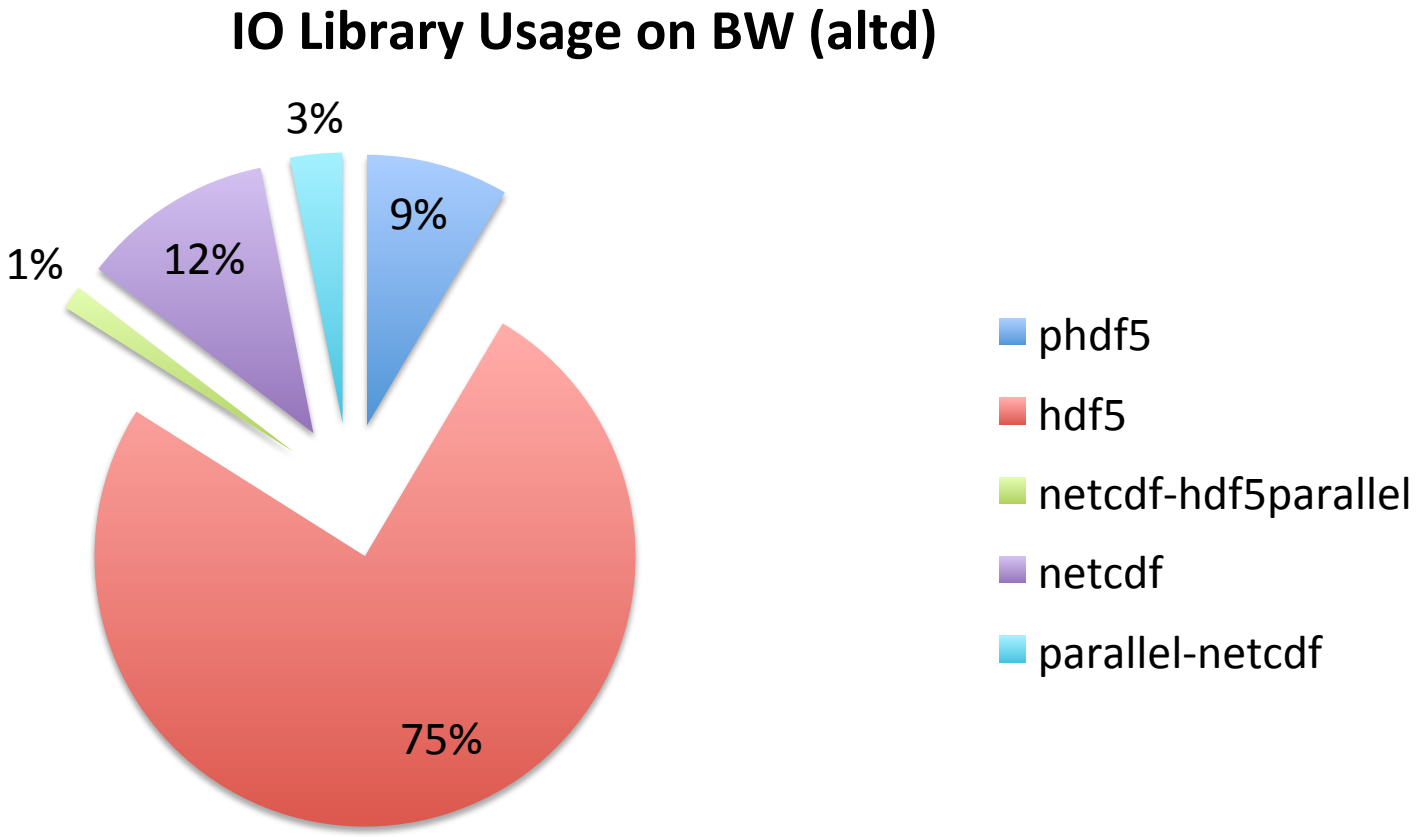
HDF5

- Hierarchical data format version 5
  - Portable, high-level I/O library
  - Self-describing
  - Used by NASA, Boeing, GE, CGNS, Mathworks
  - 100+ scientific libraries and applications
    - Silo (LLNL), FLASH (ANL), NIF (LLNL), Exodus (SNL), Chombo (LBL),…
  - Parallel HDF5
    - Makes use of MPI-IO optimizations
    - Chunking, metadata caching
    - Lustre (& other file systems) aware

Heavily used at Supercomputer Centers

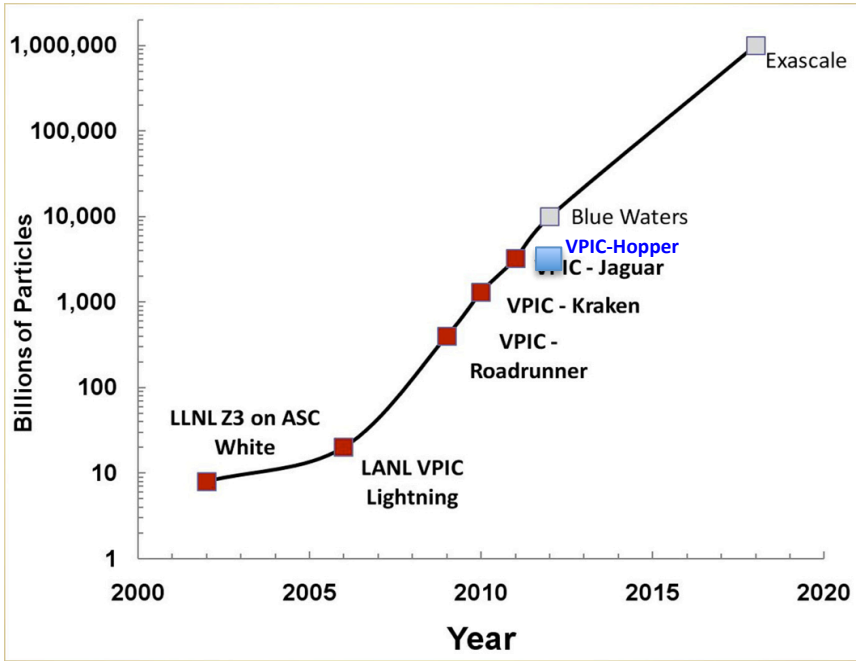


Heavily used at Supercomputer Centers

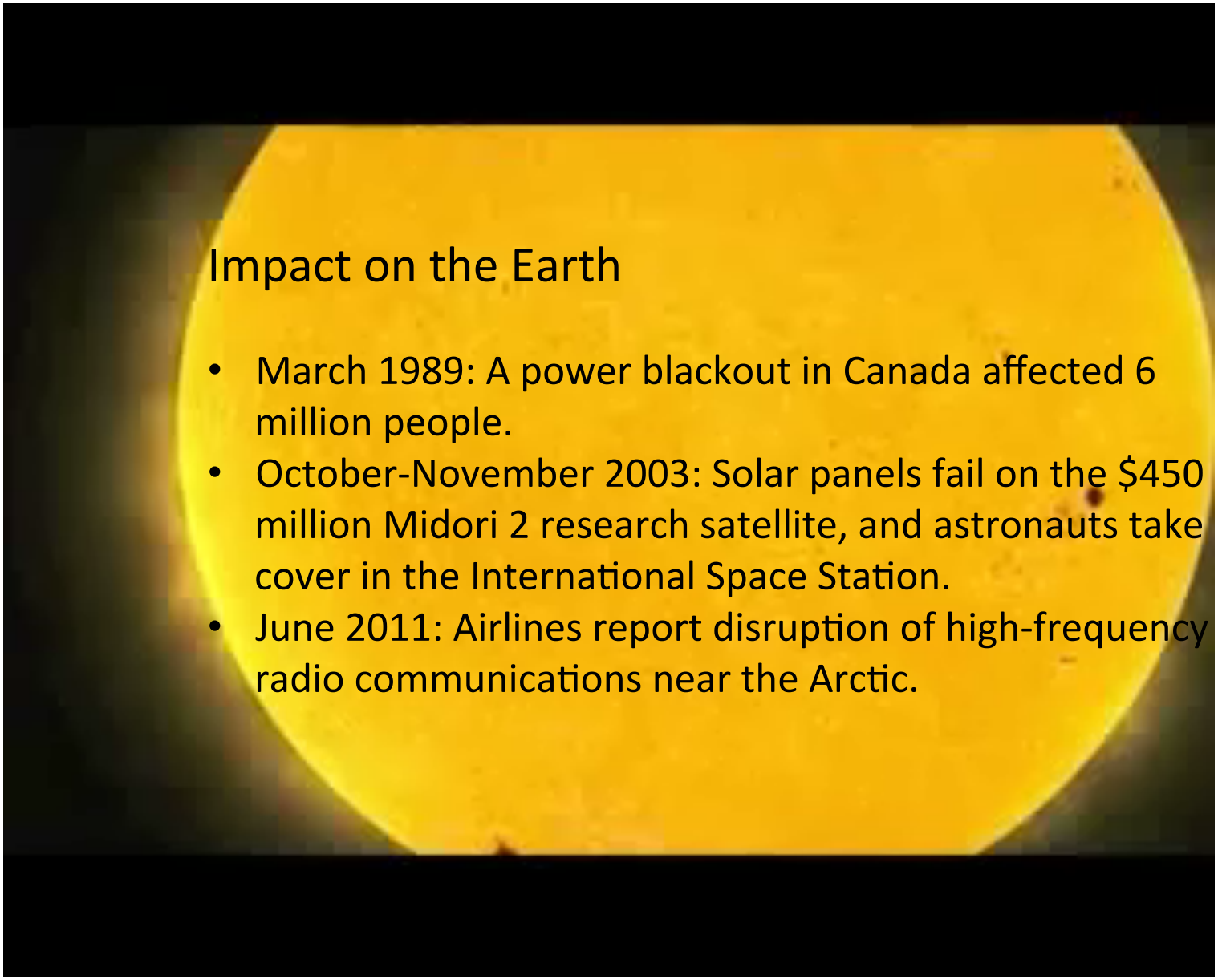


Vector Particle-in-Cell (VPIC) Code

- A state-of-the-art 3D electromagnetic relativistic PIC plasma physics simulation
- It is an exascale problem and scales well on large systems
- An open boundary VPIC simulation of magnetic reconnection (Space weather)



The application: Space Weather Simulation



Impact on the Earth

- March 1989: A power blackout in Canada affected 6 million people.
- October-November 2003: Solar panels fail on the $450 million Midori 2 research satellite, and astronauts take cover in the International Space Station.
- June 2011: Airlines report disruption of high-frequency radio communications near the Arctic.

Plasma Physics Simulation

- Magnetic Reconnection simulation
  - Simulates trillions of particles for tens of thousands of time steps
  - Particles include electrons and ions
  - Studies behavior of particles at magnetic reconnection points and in magnetic turbulence
  - Writes data pertaining to electrons at periodic intervals (every ~2000 time steps)
  - Also writes a smaller set of magnetic and electric field data

Blue Waters (IO) Configuration

- Sustained Petascale performance
  - XE6+XK7 (22,640 + 4,228) nodes
  - 13PF peak, 24X24X24 Torus
  - Several apps (incl. VPIC) > 1PF sustained performance
- Three distinct file systems > 1 TB/s aggregate
  - home, project: 98 GB/s, 2.2 PB each
  - Scratch: 980 GB/s, 22 PB usable
  - Three distinct metadata servers
  - Jobs & user interactivity don't interfere

VPIC – IO on Blue Waters

- IO kernel of a VPIC simulation's particle data write phase
- Each MPI process produces millions of particles, where each particle has 8 properties
  - 6 floats, 2 ints
- 10 trillion particles → ~300 TB per time step
  - 10x problem size compared to Hopper (CUG 2013)
- Attempt to maximize throughput to single shared file

VPIC-IO on Blue Waters

- Experimental setup
  - Build Configuration
    - Cray Compilation Environment (CCE, v.8.2.2)
    - Cray MPI(v6.2.0)
    - Parallel HDF5 1.8.11
      - Experimental multidataset feature
  - Lustre client, v.1.8.6 + patches (2.5.1 current)
    - Stripe count limited to 160 OSTs per file
  - Application Configuration
    - 5,120 to 298,048 ranks
    - 32 million+ particles per MPI rank
    - 1 GB data per write, 5 TB total file size
  - All experiments in non-dedicated mode
    - Multiple runs for each experiment

Tuning VPIC-IO on Blue Waters – Lustre & HD5 Close

- Baseline performance: 25 GB/s
- Lustre striping optimizations
  - Single file striping limited to 160 OSTs (~107 GB/s)
  - 128 MB/dataset, 8 datasets, 1 GB per rank
  - Match stripe size to data per write (128MB)
- HDF5 File close optimizations to avoid file size truncate verification
  - File size verification causing significant metadata overhead
  - Patch H5Fclose

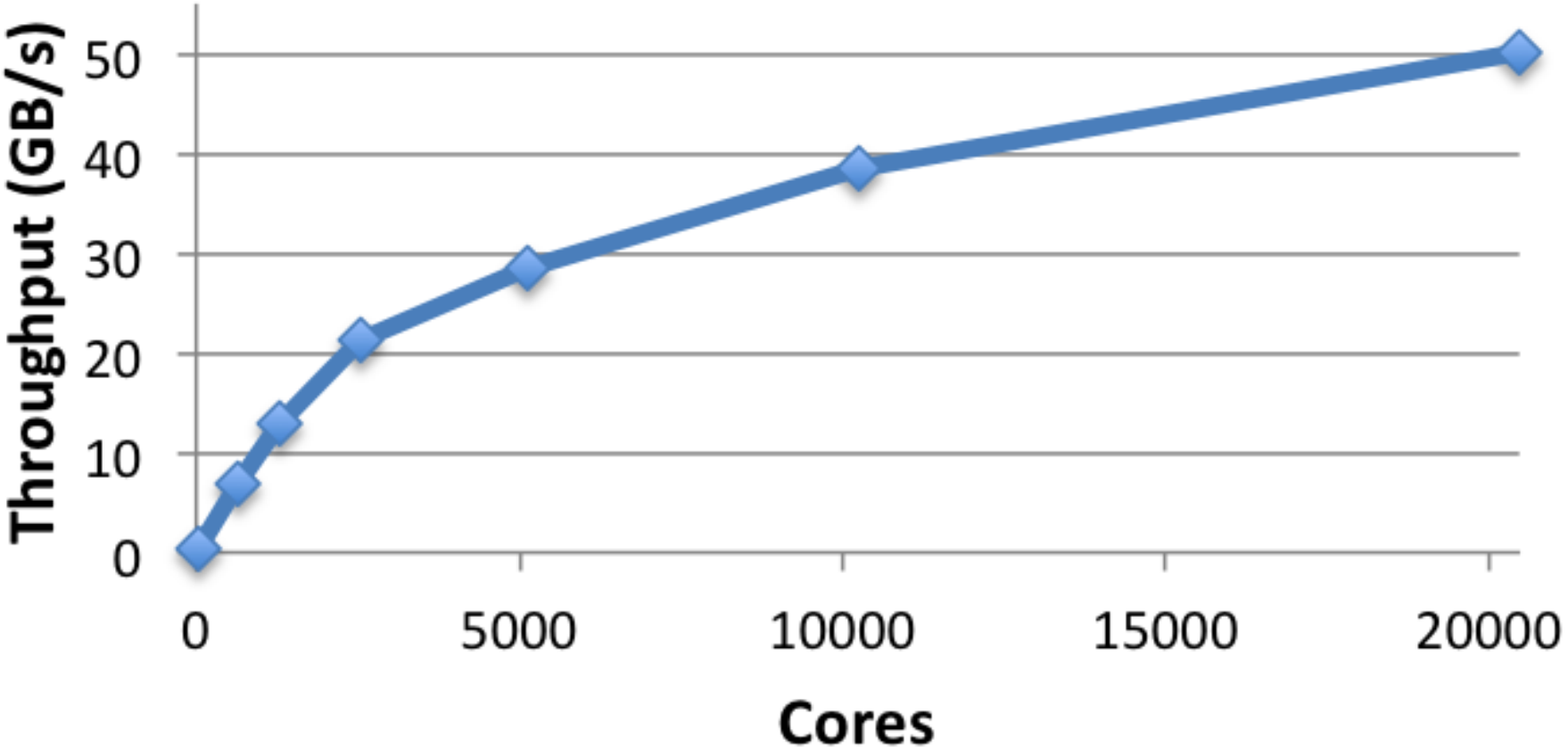
Tuning VPIC-IO on Blue Waters – Collective Buffering

- Collective Buffering
  - Enabled by default
  - Helps large unaligned & small writes to same OSTs
  - Is moving 640 GB per write efficient?
- 43.78 GB/s (5120 rank, 5 TB file) ***75% increase***
- 10 trillion particles, 300 TB file -- ?

Tuning VPIC-IO on Blue Waters – Reduce OST Oversubscription

- 10 Trillion particle run setup
  - 298,048 ranks, 32m particles/rank, 291 TB
  - Nodes to OST ratio:
    - 18628:160 vs 320:160 for baseline (116:1 vs 2:1)
    - 9318:160 using 32 ranks per node (58:1)
  - Increase stripe size to 1 GB (data written per rank)
- 51.81 GB/s ***100% increase***

Hero run results

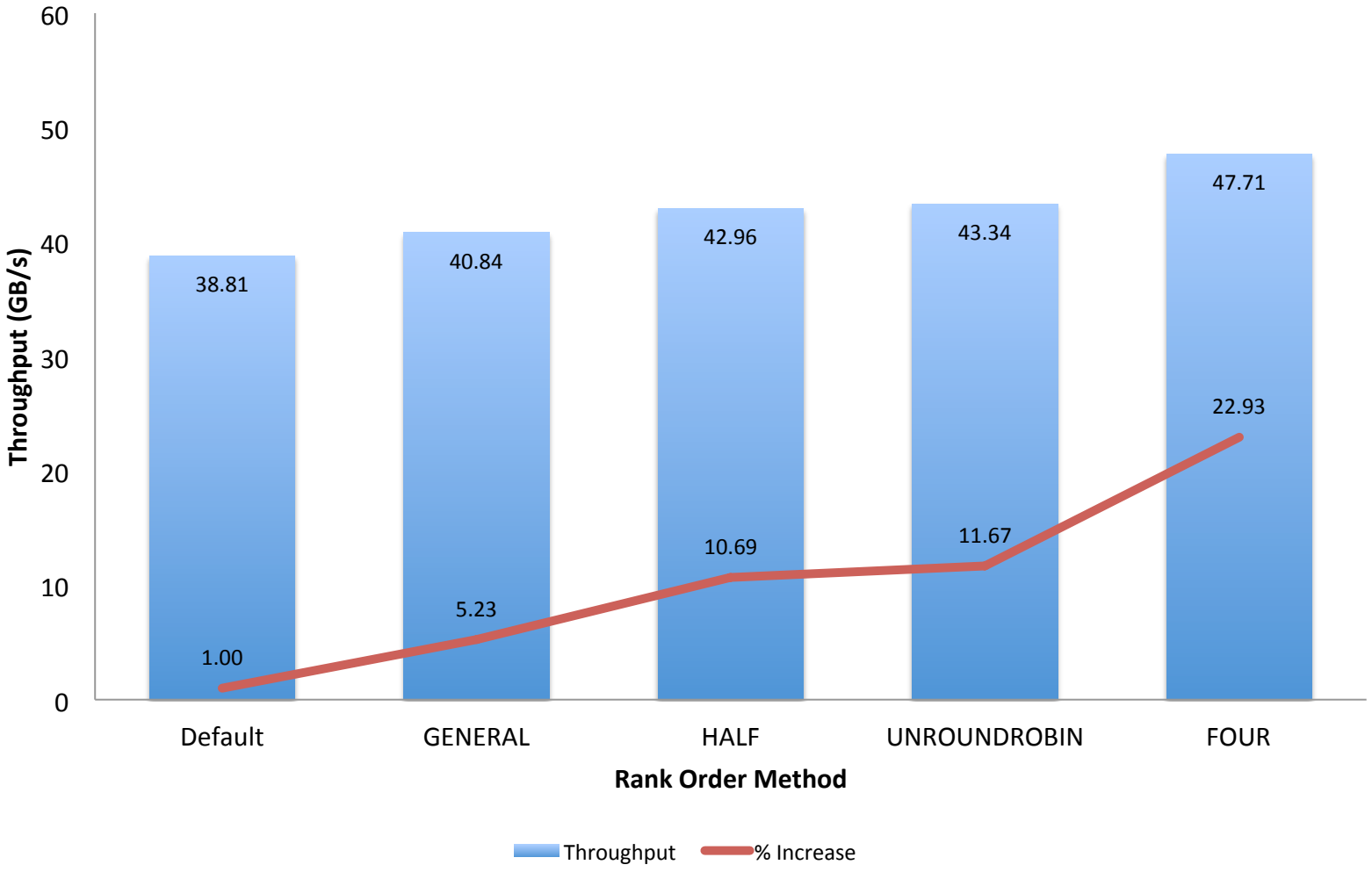


* Reduced contention from other jobs

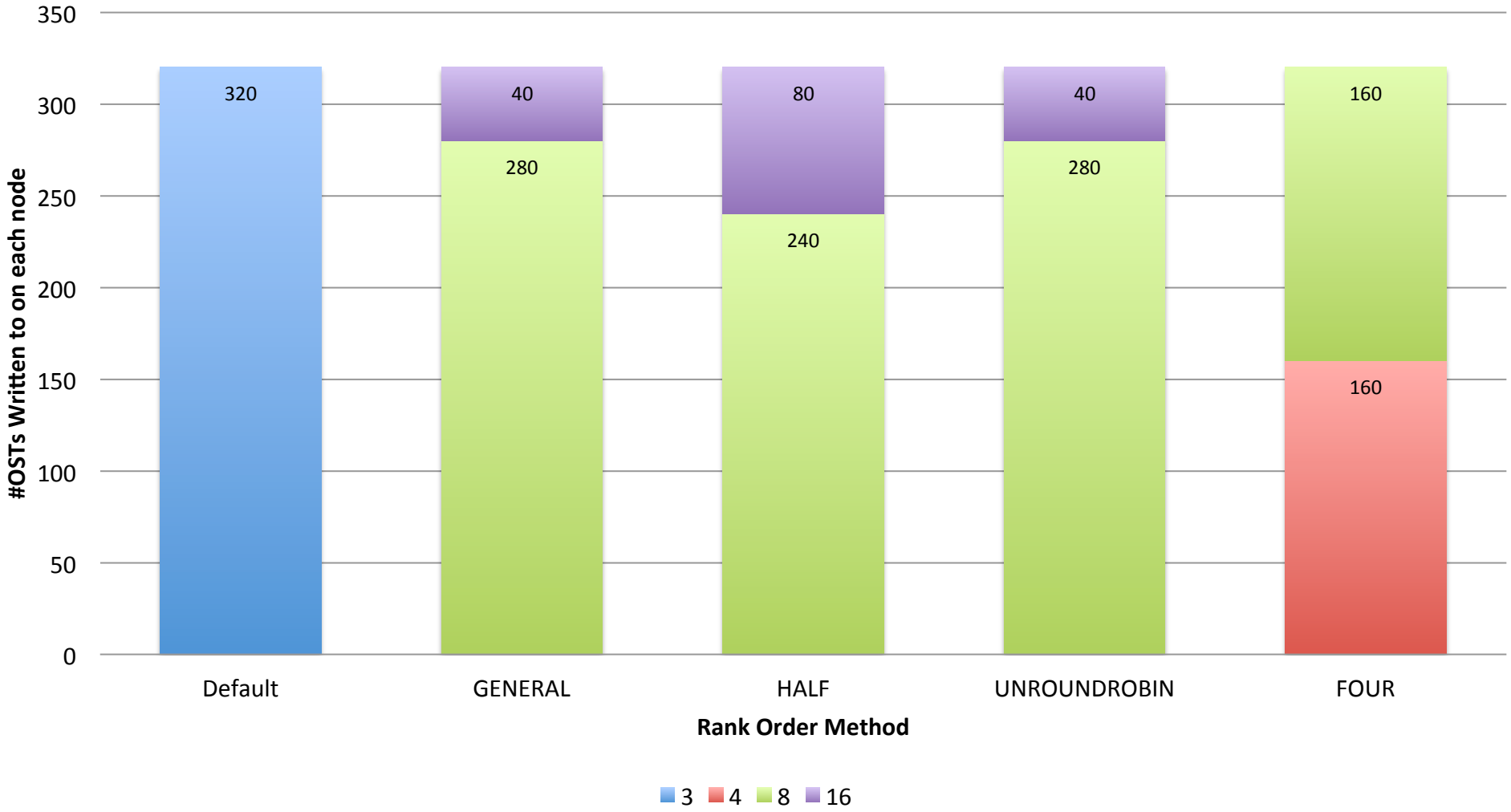
Tuning VPIC-IO on Blue Waters – Multi-dataset write

- Multi-dataset writes
  - Write multiple datasets with a single HDF5 write call
- Default Rank Order: SMP style
  - 1 node (16 ranks) -> 16 OSTs
  - 1 OST <- 32 ranks on 32 nodes
  - reorder to group ranks that write to same OST
    - Minimize #nodes writing to an OST
    - Minimize #OSTs written to from a node
- 5120 ranks, 5 TB file, 1 GB stripe
- 56.21 GB/s ***125% increase***

Rank Order Experiments - h5perf



Rank Order Experiments – #OSTs written to



Overall Performance of VPIC-IO

- 56 GB/s I/O rate in writing 5TB data using 5K cores with multi-dataset write optimization
- VPIC-IO kernel running on **298,048 cores**
  - ~10 Trillion particles
  - **291 TB, single file**
  - 1 GB stripe size and 160 Lustre OSTs
  - ~~52 GB/s~~ 60.51 GB/s
  - 56% of the peak performance

Conclusions and Future Work

- Sub-filing
  - Current implementation writes all the data into a single shared file
  - With sub-filing, data can be written to multiple smaller shared files (instead of having ~300TB file)
- On Blue Waters
  - Testing with new Lustre configuration with >160 OSTs (peak I/O rate: ~1 TB/s, 1440 OSTs)
  - Experiments w/Declustered RAID (Summer 2015?)

Questions?

THANK YOU