

Science Driven System Architecture

John Shalf

SDSA Mission

- **Analyze Office of Science HPC Workload**
 - Define subset of applications to represent workload requirements (SSP)
 - Assess emerging workload requirements in response to tech trends
- **Use analysis of full applications to**
 - Compare HPC systems based on “delivered” performance on target workload
 - Profile applications to quantify sources of performance degradation
 - Create proxy applications to represent full-application requirements
- **Assess technology alternatives to make better supercomputers for science requirements**
 - Work together with vendors on technology alternatives (Blue Planet)
 - Perform our own experiments/testbeds (Green Flash)

Overlap with SCG and FTG

- **Many members of SDSA are matrixed from LBNL Computing Research Division (SCG, FTG, and ANAG)**
 - *Expect some overlap with FTG and SCG presentation*
- **SDSA focus on applying knowledge gained from CRD collaboration to NERSC operations and planning**

Overview of Activities

- **NERSC Workload Analysis**
 - “Benchmarks are only useful insofar as they model the intended workload”
- **Benchmark selection/packaging/analysis for NERSC procurements**
 - NERSC Sustained System Performance composite benchmark
 - AMR and I/O microbenchmarks
- **I/O benchmarking and tuning**
 - Predicting full application performance with a synthetic proxy
 - HDF5 Library Tuning
- **Programming model/language survey**
 - What is the practical programming model/language for expressing fine-grained parallelism
 - How can we get strong-scaling from explicit on-chip parallelism

Workload Analysis

Workload Analysis

- **Understand Office of Science Computational Requirements**
 - Augment with anticipated algorithm/science/technology trends
- **Identify key performance features and minimum requirements for RFP**
- **Inform NERSC Sustained System Performance (SSP) Benchmark Selection**
 - Effective performance on SSP to reflect effective performance on NERSC workload

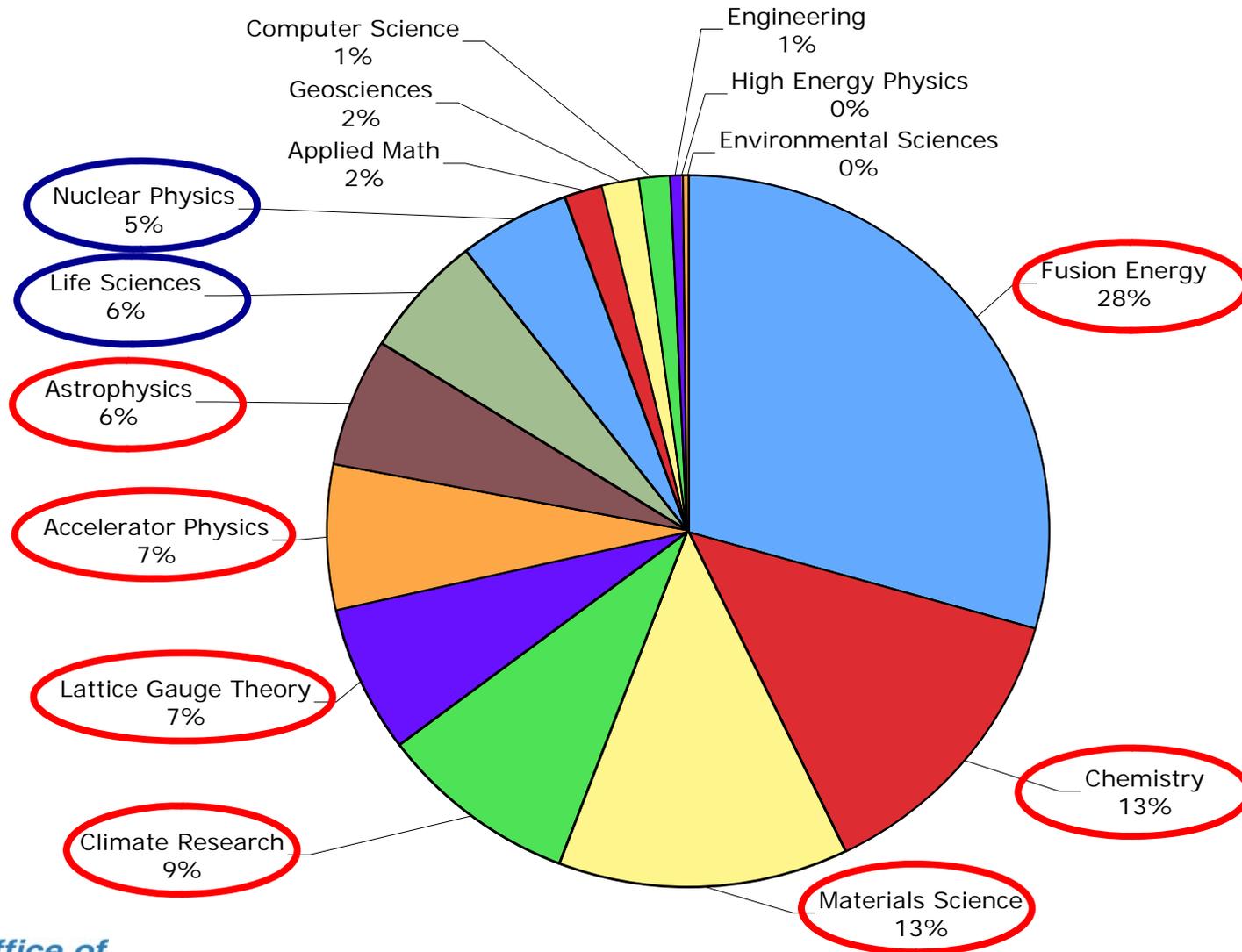
“Benchmarks are only useful insofar as they model the intended computational workload.”

Ingrid Bucher & Joanne Martin, LANL, 1982

Balancing Requirements

- **NERSC Workload overview**
 - ~3000 users
 - ~300-400 projects representing a broad range of science
 - ~500-700 codes (~2 codes per project on average!)
 - 15 science areas for 6 Office of Science divisions
- **Select a subset (<10) codes to represent the requirements of the workload**
 - Contribution workload (workload coverage)
 - Contribution to each area of science (algorithm/science-area coverage)
- **Must cover algorithm usage across science areas**
 - Assumes evolving workload (don't alienate science areas)
 - Search for islands of coherence in the codes or algorithm selection by different scientific disciplines
 - *Still daunting*

Focus on Science Areas

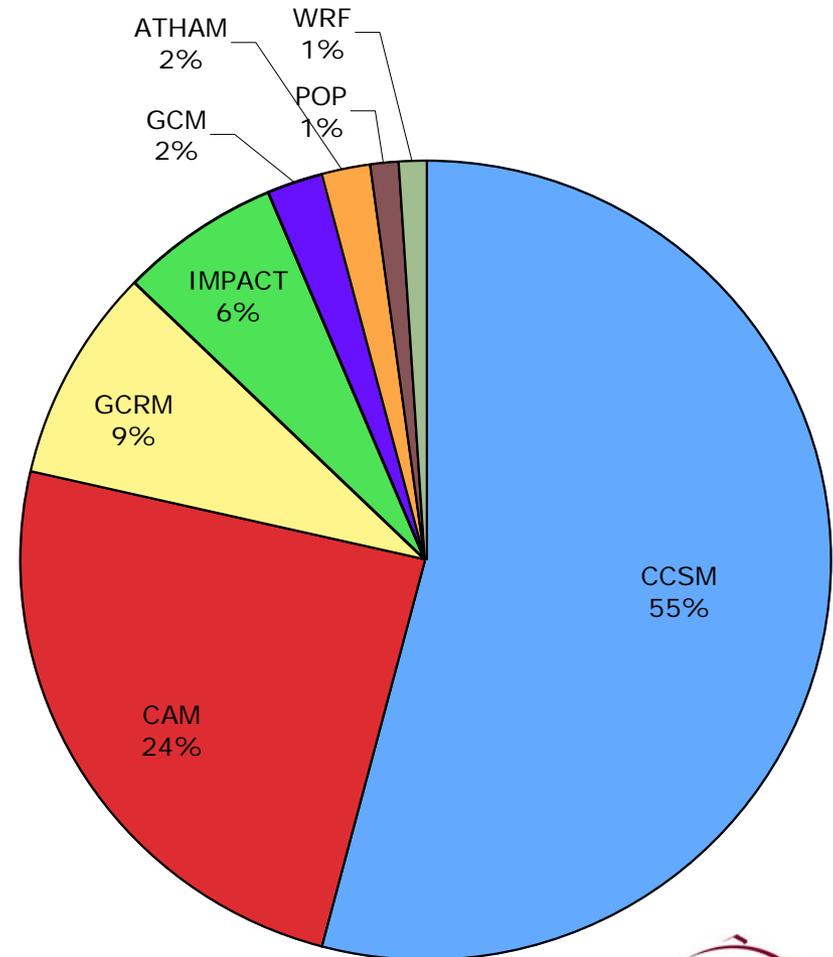


Example: Climate Modeling (BER)

	Code	MPP Award	Percent	Cumulative%
1	CCSM	2,342,000	51%	51%
2	CAM	2,000,000	23%	74%
3	GCRM	2,000,000	8%	82%
4	IMPACT	1,085,000	6%	88%
5	GCM	375,000	2%	90%
6	ATHAM	280,000	2%	92%
7	POP	100,000	1%	93%
8	WRF	80,000	1%	94%

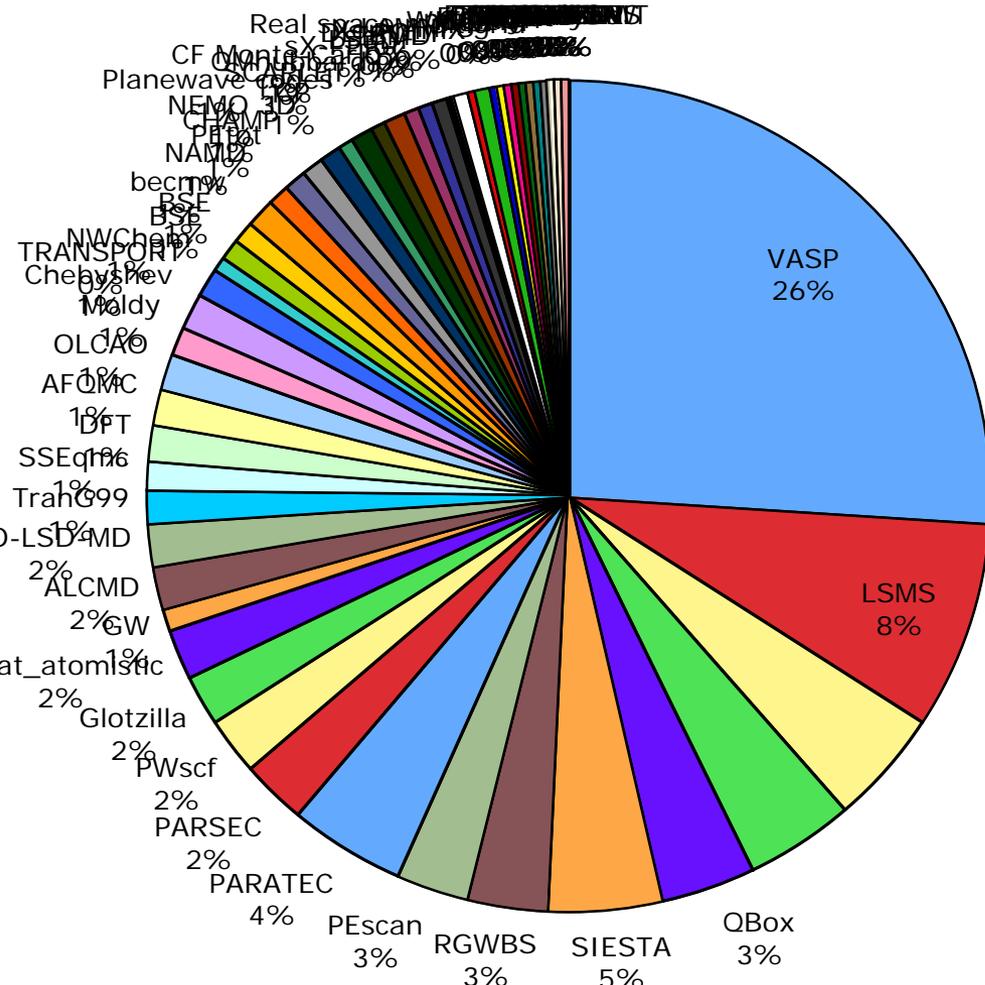
- CAM and POP dominate CCSM computational requirements
- FV-CAM increasingly replacing Spectral-CAM in future CCSM calculations
- FV-CAM with D-Mesh selected (coordinate w/NCAR procurement)

Climate without INCITE



Example: Material Science

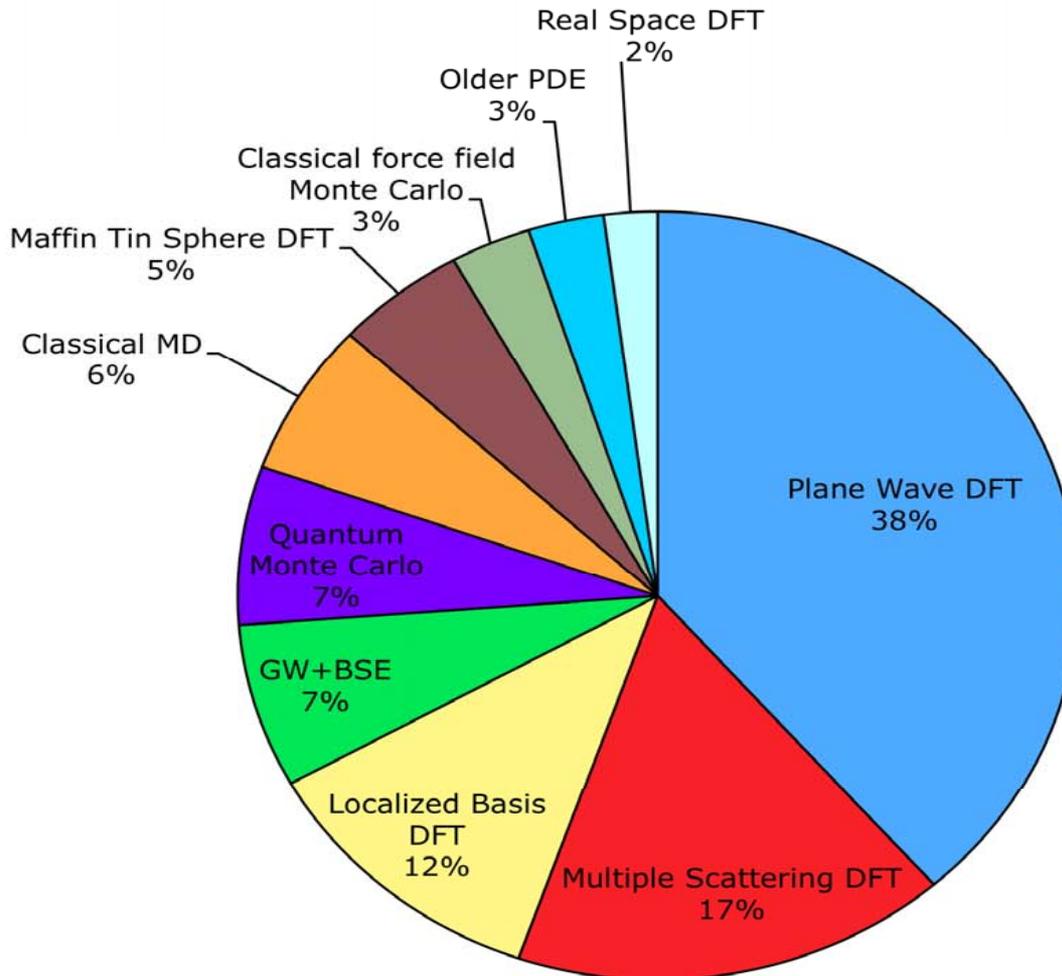
- 7,385,000 MPP hours awarded
- 62 codes, 65 users
- Typical code used in 2.15 allocation requests



	Code	MPP Hours	Percent	Cumulative%
1	VASP	1,992,110	26%	26%
2	LSMS	600,000	8%	34%
3	FLAPW, DMol3	350,000	5%	39%
4	CASINO	312,500	4%	43%
5	QBox	262,500	3%	46%
6	SIESTA	346,500	5%	51%
7	RGWBS	232,500	3%	54%
8	PEscan	220,000	3%	57%
9	PARATEC	337,500	4%	61%
10	PARSEC	182,500	2%	64%
	Other	167,300	34%	66%

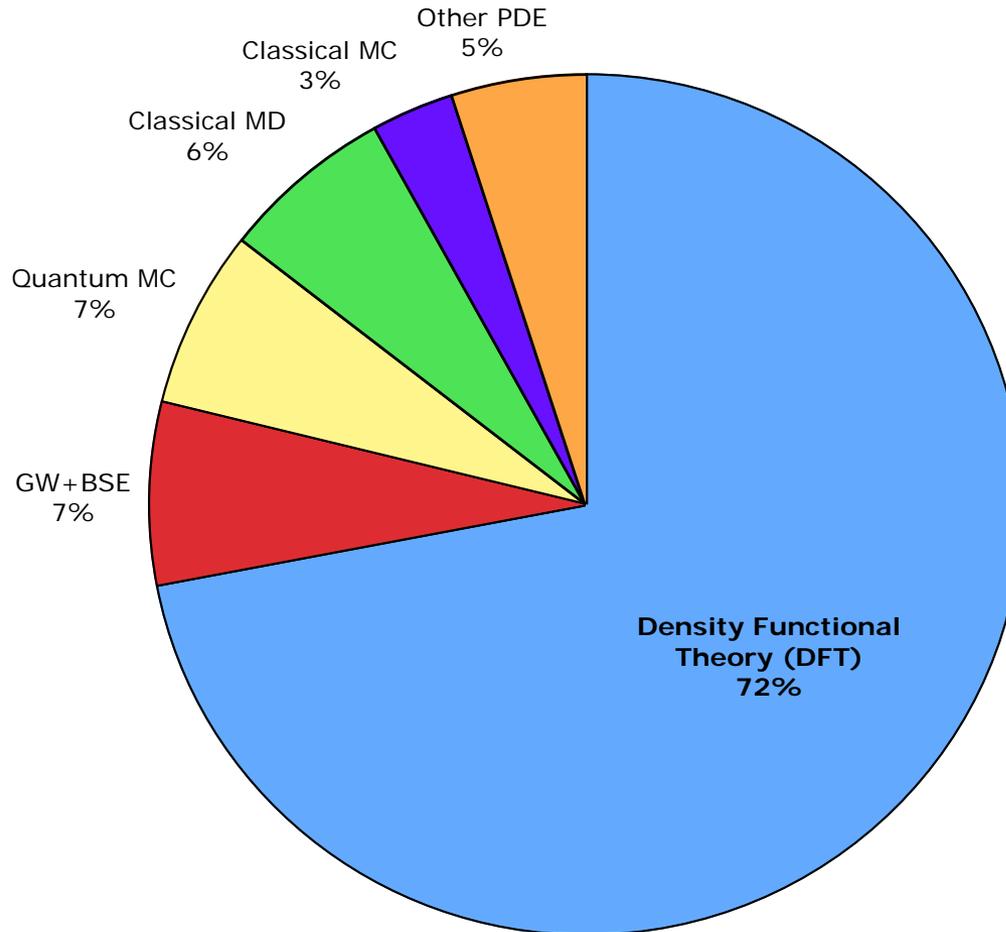
Example: Materials Science *(by algorithm)*

Analysis by Lin-Wang Wang

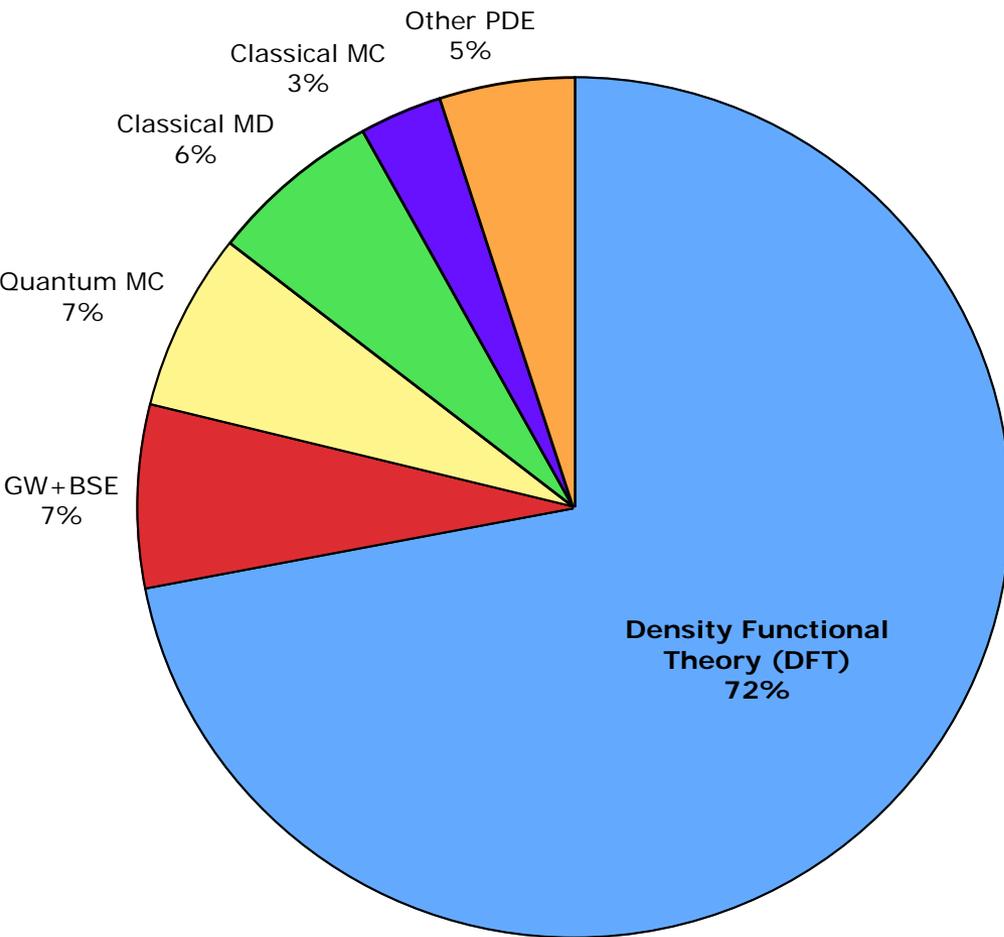


Example: Materials Science (by algorithm category)

Analysis by Lin-Wang Wang



Materials Science (by algorithm category)



- Density Functional Theory codes
 - >70% of the workload!
 - Majority are planewave DFT!
- Common requirements for DFT
 - 3D global FFT
 - Dense Linear Algebra for orthogonalization of wave basis functions
 - Dense Linear Algebra calculating pseudopotential
- Dominant Code: VASP
- Similar Codes (planewave DFT)
 - QBox
 - PARATEC
 - PETOT/PESCAN

Other Application Areas

- **Fusion: 76 codes**

- 5 codes account for >50% of workload: OSIRIS, GEM, NIMROD, M3D, GTC
- Further subdivide to PIC (OSIRIS, GEM, GTC) and MHD (NIMROD, M3D) code categories

- **Chemistry: 56 codes for 48 allocations**

- Planewave DFT: VASP, CPMD, DACAPO
- Quantum Monte Carlo: ZORI
- Ab-initio Quantum Chemistry: Molpro, Gaussian, GAMESS
- Planewave DFT dominates (but already covered in MatSci workload)
- Small allocations Q-Chem category add up to dominant workload component

- **Accelerator Modeling**

- 50% of workload consumed by 3 codes VORPAL, OSIRIS, QuickPIC
- Dominated by PIC codes

code	MPP Award	Percent	Cumulative%
OSIRIS	2,112,500	11%	11%
GEM	2,058,333	11%	22%
NIMROD	2,229,167	12%	34%
M3D	1,921,667	10%	45%
GTC	1,783,333	10%	54%

Code	Award	Percent	Cumulative%
ZORI	695,000	12%	12%
MOLPRO	519,024	9%	21%
DACAPO	500,000	9%	29%
GAUSSIAN	408,701	7%	36%
CPMD	396,607	7%	43%
VASP	371,667	6%	49%
GAMESS	364,048	6%	56%

Code	MPP Award	Percent	Cumulative%
VORPAL	1,529,786	33%	33%
OSIRIS	784,286	16%	49%
QuickPIC	610,000	13%	62%
Omega3p	210,536	4%	66%
Track3p	210,536	4%	70%

Benchmark Selection Criteria

- **Coverage**
 - Cover science areas
 - Cover algorithm space
- **Portability**
 - Robust ‘build’ systems
 - Not architecture specific implementation
- **Scalability**
 - Do not want to emphasize applications that do not justify scalable HPC resources
- **Open Distribution**
 - No proprietary or export-controlled code
- **Availability of Developer for Assistance/Support**

NERSC-6 Application Benchmarks

<i>Benchmark</i>	<i>Science Area</i>	<i>Algorithm Space</i>	<i>Base Case Concurrency</i>	<i>Problem Description</i>	<i>Lang</i>	<i>Libraries</i>
CAM	Climate (BER)	Navier Stokes CFD	56, 240 Strong scaling	D Grid, (~.5° resolution); 240 timesteps	F90	netCDF
GAMESS	Quantum Chem (BES)	Dense linear algebra	384, 1024 (Same as Ti-09)	DFT gradient, MP2 gradient	F77	DDI, BLAS
GTC	Fusion (FES)	PIC, finite difference	512, 2048 Weak scaling	100 particles per cell	F90	
IMPACT-T	Accelerator Physics (HEP)	PIC, FFT	256,1024 Strong scaling	50 particles per cell	F90	
MAESTRO	Astrophysics (HEP)	Low Mach Hydro; block structured-grid multiphysics	512, 2048 Weak scaling	16 32³ boxes per proc; 10 timesteps	F90	Boxlib
MILC	Lattice Gauge Physics (NP)	Conjugate gradient, sparse matrix; FFT	256, 1024, 8192 Weak scaling	8x8x8x9 Local Grid, ~70,000 iters	C, assem.	
PARATEC	Material Science (BES)	DFT; FFT, BLAS3	256, 1024 Strong scaling	686 Atoms, 1372 bands, 20 iters	F90	Scalapack, FFTW

Algorithm Diversity

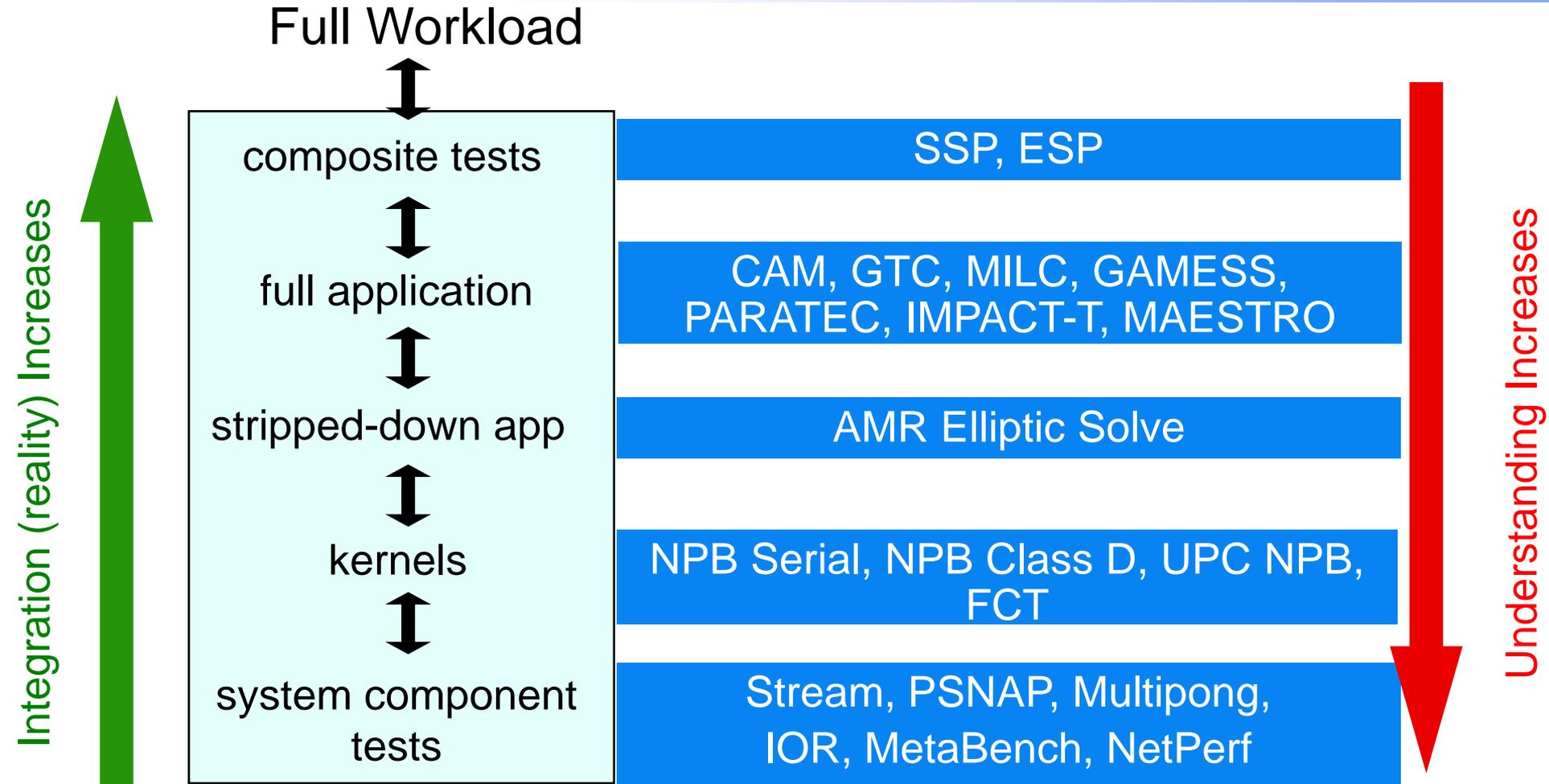
<i>Science areas</i>	<i>Dense linear algebra</i>	<i>Sparse linear algebra</i>	<i>Spectral Methods (FFT)s</i>	<i>Particle Methods</i>	<i>Structured Grids</i>	<i>Unstructured or AMR Grids</i>
Accelerator Science		X	X	X	X	X
Astrophysics	X	X	X	X	X	X
Chemistry	X	X	X	X		
Climate			X		X	X
Combustion					X	X
Fusion	X	X		X	X	X
Lattice Gauge		X	X	X	X	
Material Science	X		X	X	X	

NERSC users require a system which performs adequately in all areas

N6 Benchmarks Coverage

Science areas	Dense linear algebra	Sparse linear algebra	Spectral Methods (FFT)s	Particle Methods	Structured Grids	Unstructured or AMR Grids
Accelerator Science		X	X IMPACT-T	X IMPACT-T	X IMPACT-T	X
Astrophysics	X	X MAESTRO	X	X	X MAESTRO	X MAESTRO
Chemistry	X GAMESS	X	X	X		
Climate			X CAM		X CAM	X
Combustion					X MAESTRO	X AMR Elliptic
Fusion	X	X		X GTC	X GTC	X
Lattice Gauge		X MILC	X MILC	X MILC	X MILC	
Material Science	X PARATEC		X PARATEC	X	X PARATEC	

Benchmark Hierarchy



Consistency is measured over all benchmarks as Coefficient of variation from 5 consecutive runs.

NERSC-6 Composite SSP Metric

The largest concurrency run of each full application benchmark is used to calculate the composite SSP metric

NERSC-6 SSP

CAM
240p

GAMESS
1024p

GTC
2048p

IMPACT-T
1024p

MAESTRO
2048p

MILC
8192p

PARATEC
1024p

For each benchmark measure

- FLOP counts on a reference system*
- Wall clock run time on various systems*

Example of N6 SSP on Hypothetical System

Hypothetical N6 System	Results			
	Tasks	System Gflopcnt	Time	Rate per Core
CAM	240	57,669	408	0.589
GAMESS	1024	1,655,871	2811	0.575
GTC	2048	3,639,479	1493	1.190
IMPACT-T	1024	416,200	652	0.623
MAESTRO	2048	1,122,394	2570	0.213
MILC	8192	7,337,756	1269	0.706
PARATEC	1024	1,206,376	540	2.182
GEOMETRIC MEAN				0.7

Rate Per Core =
Ref. Gflop count /
(Tasks*Time)

Flop count
measured on
reference
system

Measured wall
clock time on
hypothetical
system

Geometric
mean of
'Rates per
Core'

SSP (TF) = Geo mean of rates per core * # cores in system / 1000

N6 SSP of 100,000 core system = $0.7 * 100,000 / 1000 = 70$

N6 SSP of 200,000 core system = $0.7 * 200,000 / 1000 = 140$

Allows vendors to size systems based on benchmark performance

Benchmarks Must Evolve in Response to Technology Trends

- **Parallel computing has thrived on weak-scaling for past 15 years**
- **Flat CPU performance increases emphasis on strong-scaling**
- **Benchmarks changed accordingly**
 - **Concurrency:** *Increased 4x over NERSC-5 benchmarks*
 - **Strong Scaling:** *Input decks emphasize strong-scaled problems*
 - **Implicit Methods:** *Added MAESTRO application benchmark*
 - **Multiscale:** *Added AMR Poisson benchmark*
 - **Lightweight Messaging:** *Added UPC FT benchmark*

I/O Microbenchmarks

(Honzhang Shan)

Can a synthetic benchmark be used to predict the I/O performance of full scientific applications?

Use Case: Use a single proxy benchmark to replace running full application benchmarks on a number of systems (e.g. in a procurement).

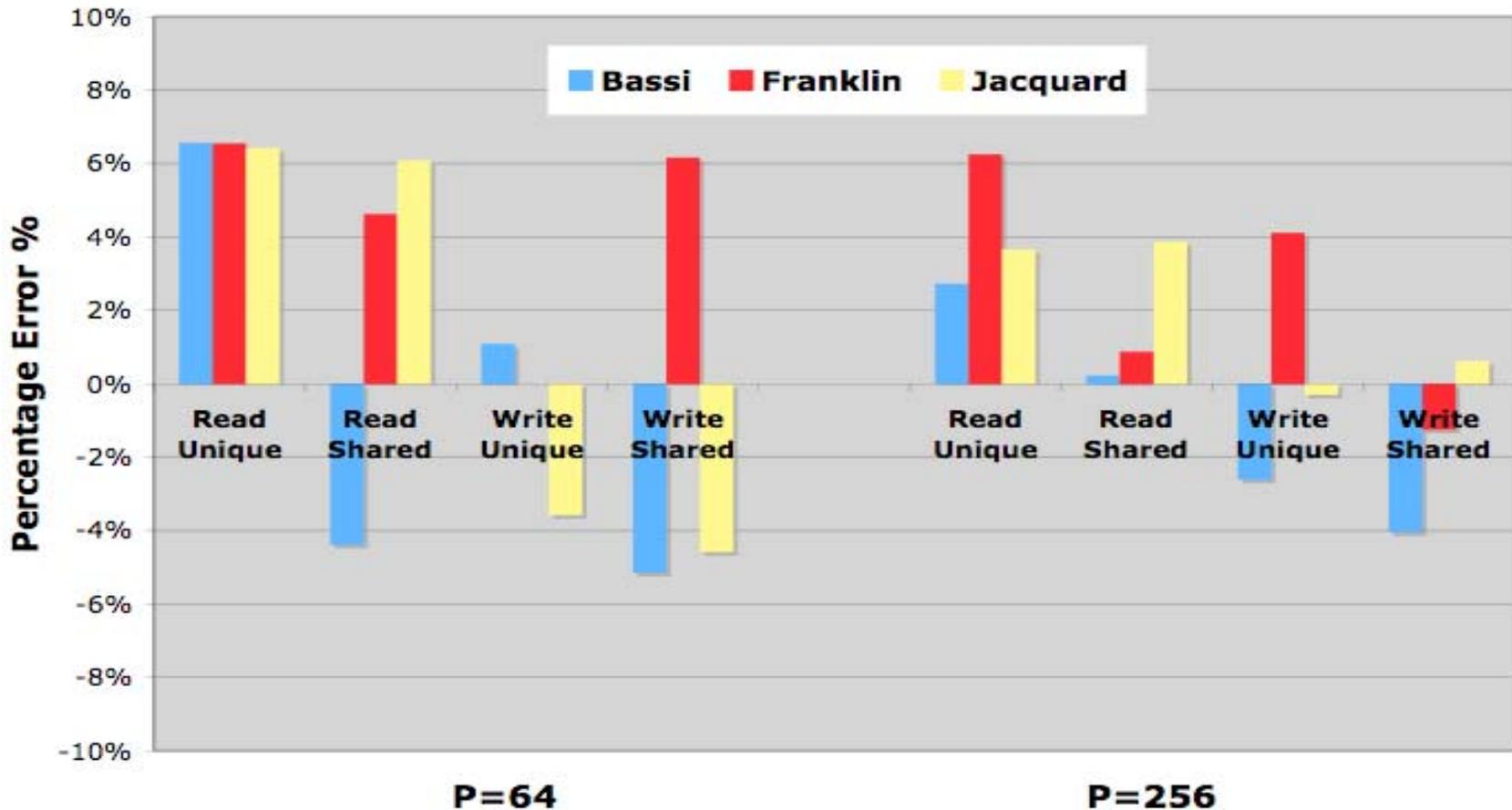
- **Accurately reflect the requirements of the intended workloads**
- **Accurately predict the performance for applications**

More rigorous than simply mimicking application I/O patterns

Approach

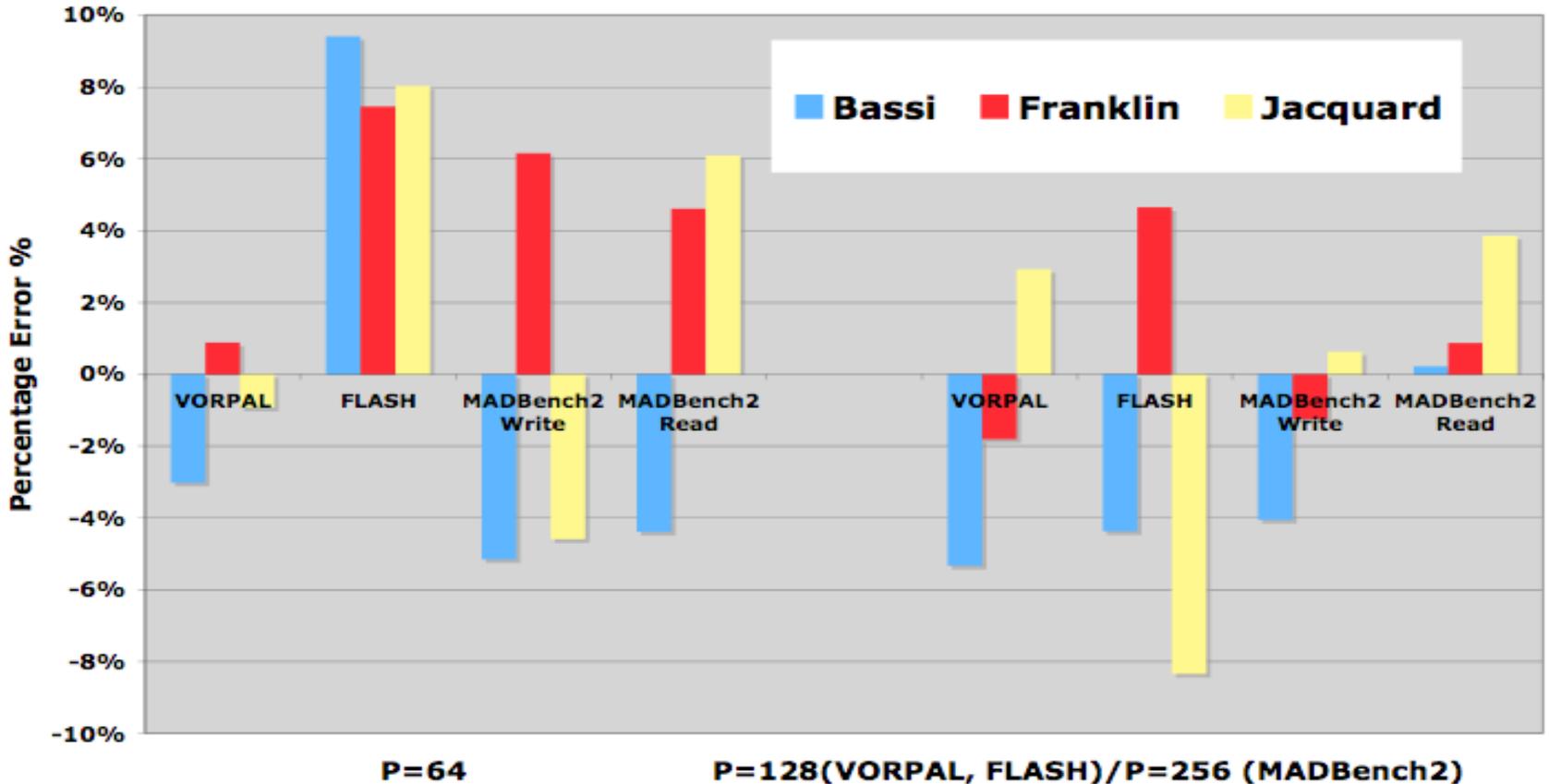
- **Survey the I/O requirements of DOE Office of Science applications**
- **Investigate the use of a synthetic benchmark as an application proxy to study I/O performance on DOE platforms**
- **Determine effectiveness of this synthetic benchmark in predicting I/O performance across a range of applications**

I/O Performance Prediction for MADBench2



IOR captures essential features of madbench2 IO behavior, both access patterns and performance.

I/O Performance Prediction



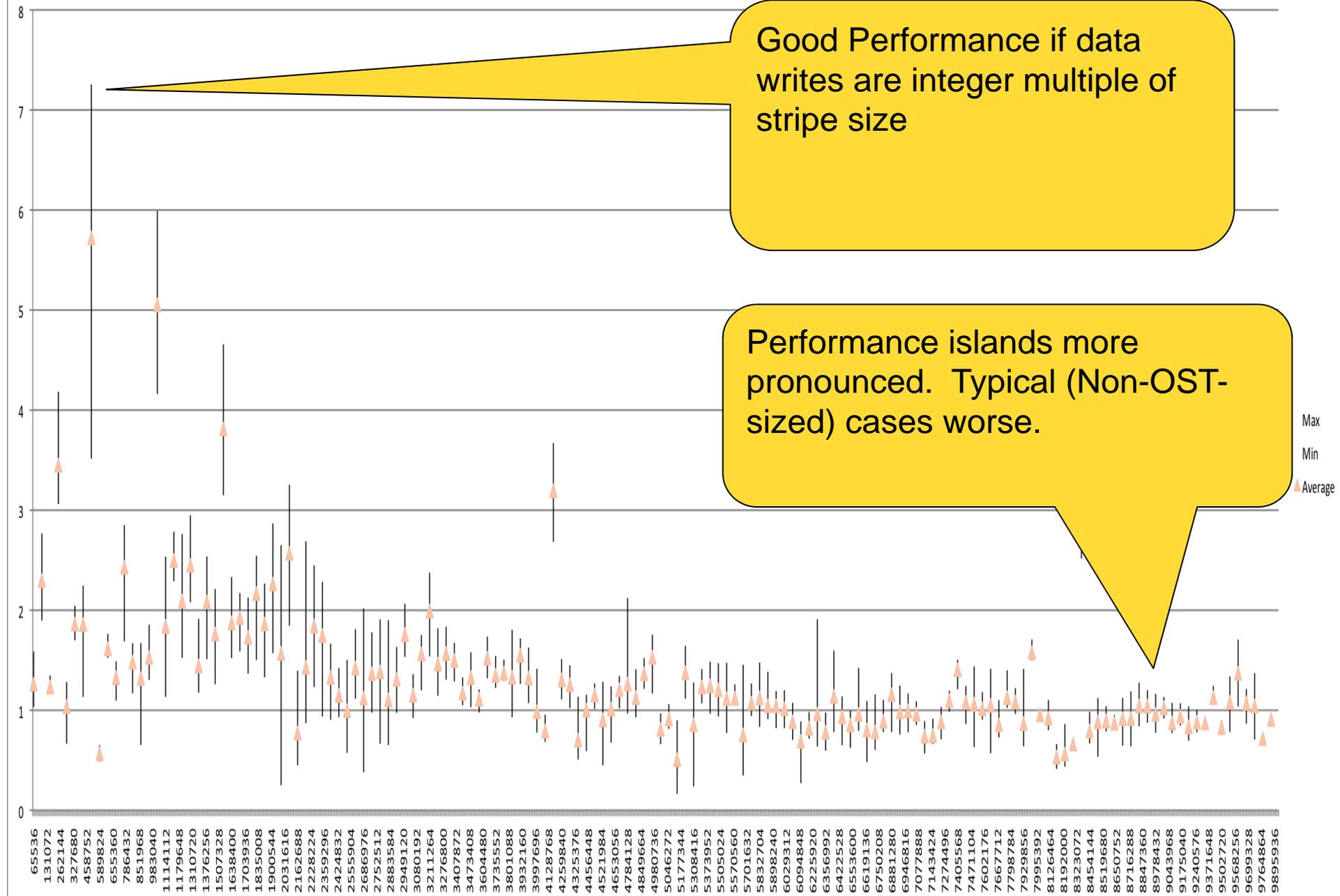
- Prediction error is within 10% in the worst case

HDF5 Performance Tuning

What is HDF5

- **Self-describing portable file format**
- **Implements Object Database Data Model**
 - Abstracts specifics of file layout (like conventional database)
 - Enables focus on naming of objects and high-level data relationships
- **Popular among DOE and NSF user community**
 - 3rd most popular lib in NERSC workload survey
- **Performing poorly on Cray/Lustre Filesystem**
 - Due to lack of investment in maintenance

80 processors with 80 OSTs (Shane case)

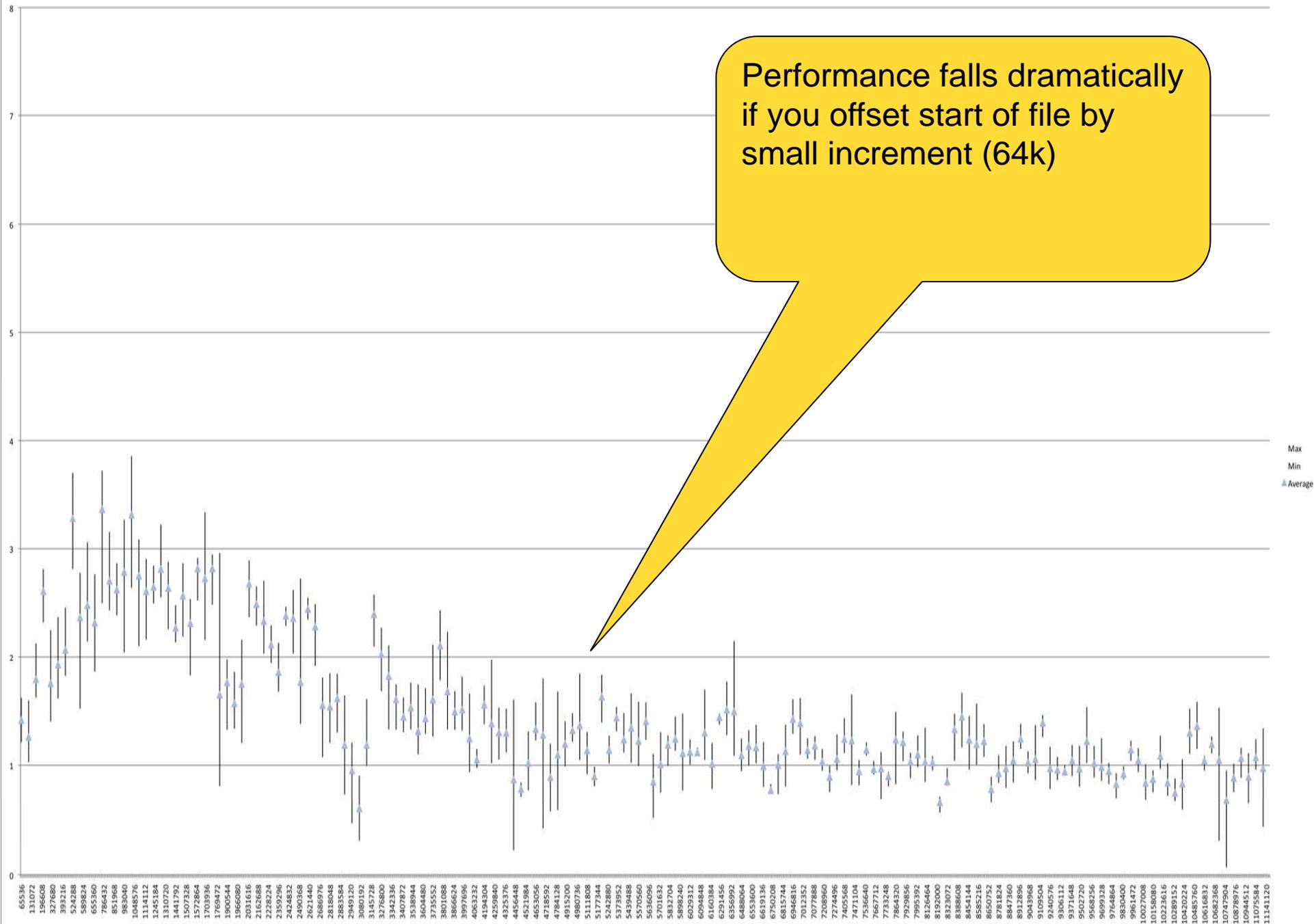


Good Performance if data writes are integer multiple of stripe size

Performance islands more pronounced. Typical (Non-OST-sized) cases worse.

80 processors with 40 OST : offset file start by 64k

Performance falls dramatically if you offset start of file by small increment (64k)



Impractical to aim for such small “performance islands” with conventional I/O

- **Transfer size for interleaved I/O must always match OST stripe width (1 Megabyte)**
 - Difficult to constrain domain-decomposition to granularity of I/O
 - Compromises load balancing for particle codes
 - Not practical for AMR codes (load-balanced, but not practical to have exactly identical domain sizes)
- **Every compute node must write exactly aligned to OST boundary (1 Megabyte alignment)**
 - How is this feasible if users write metadata or headers to their files?
 - Not practical when domain-sizes are slightly non-uniform (such as AMR, particle load balancing, outer-boundary conditions for 3D grids)

Apply Tuning Principles to HDF5

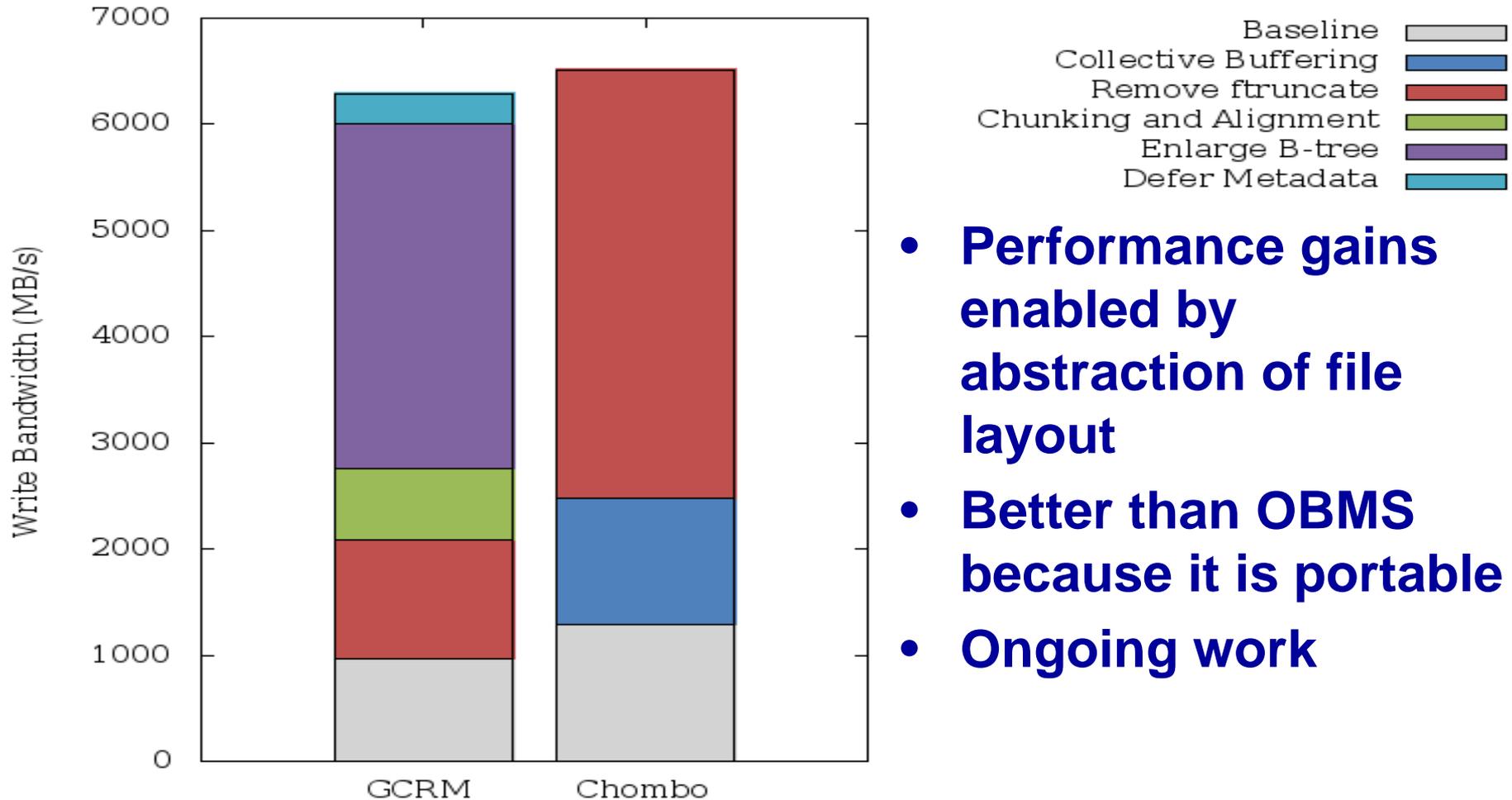
(make optimizations transparent to users)

- **Small writes are bad (aggregate to >1MB operations)**
- **Use wide striping on Lustre for parallel I/O**
- **Choose #stripes to be multiple of #clients**
 - Best to set striping before writing to file
- **Use transaction sizes equal to stripe size**
- **Defer Metadata writes**
- **Align writes to stripe boundaries**
 - Even if writes to file are sparse
- **2-phase I/O to fix alignment issues**
 - # I/O clients equal to #OSTs assigned
 - Reorganize I/O so that it is always aligned to OSTs (e.g. Data organized so Client #1 always handles transactions for same OST)

Performance Tuning Results

(first two applications)

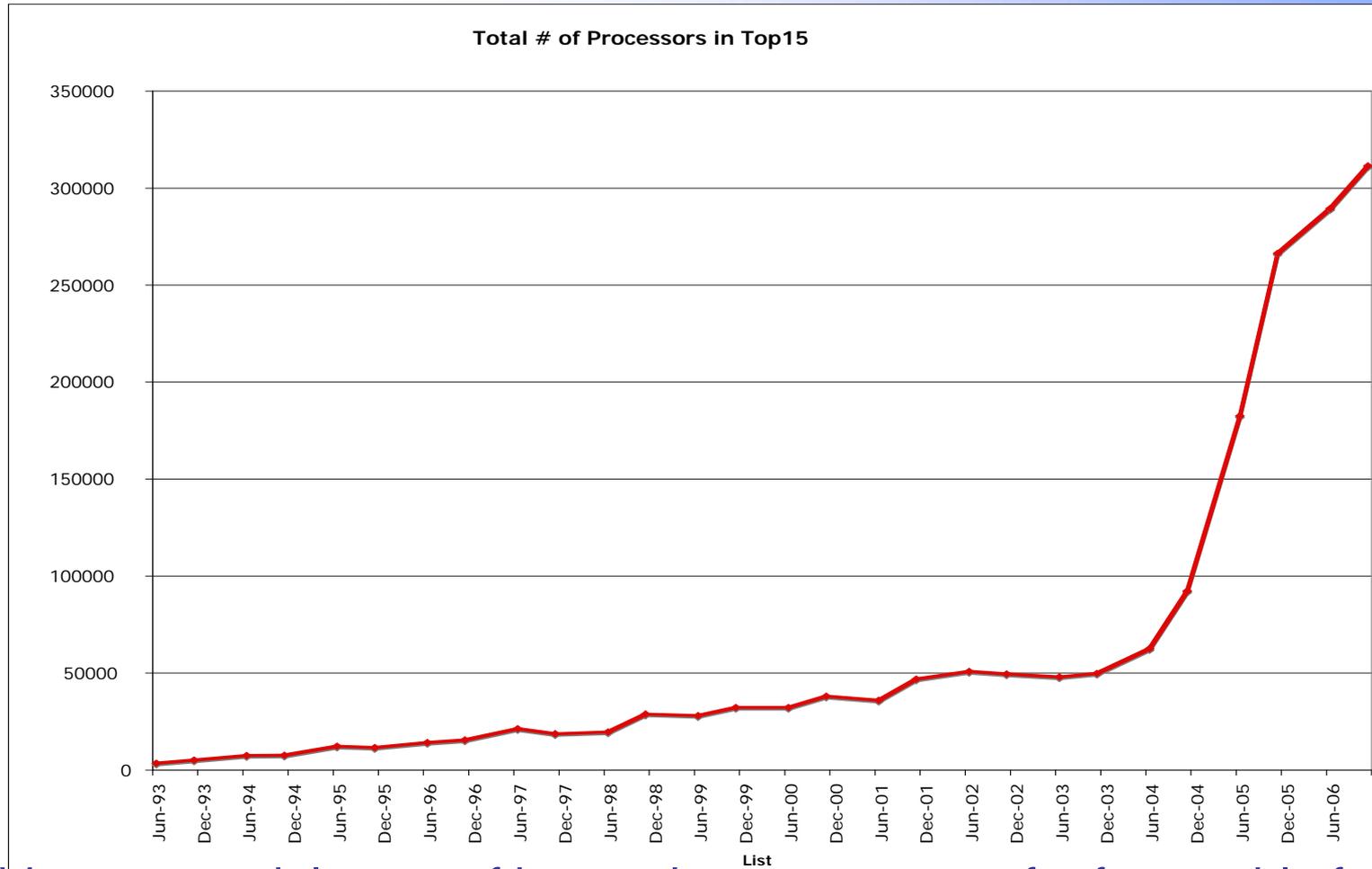
HDF I/O Tuning



- **Performance gains enabled by abstraction of file layout**
- **Better than OBMS because it is portable**
- **Ongoing work**

New Programming Model for expressing Fine-Grained Intranode Parallelism

The Future of HPC System Concurrency



Must ride exponential wave of increasing concurrency for foreseeable future!

Fortunately, most of the concurrency growth is within a single socket

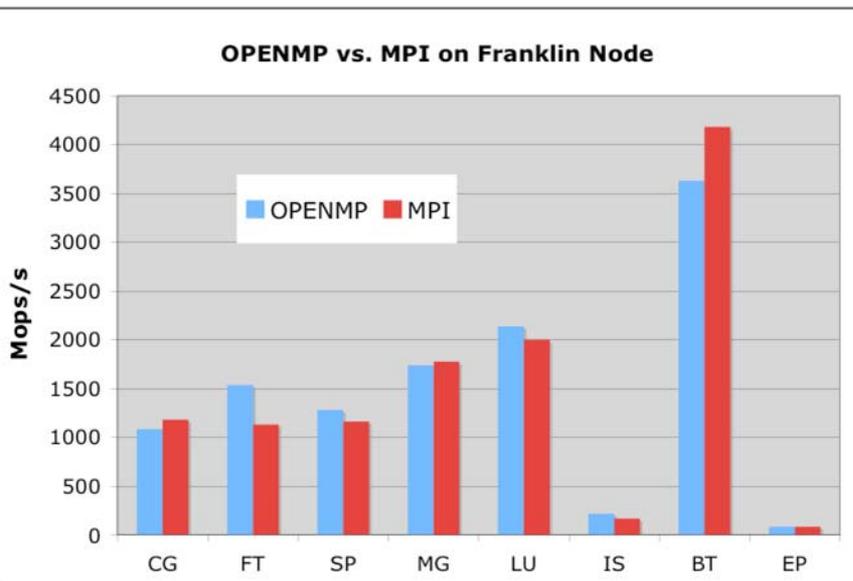
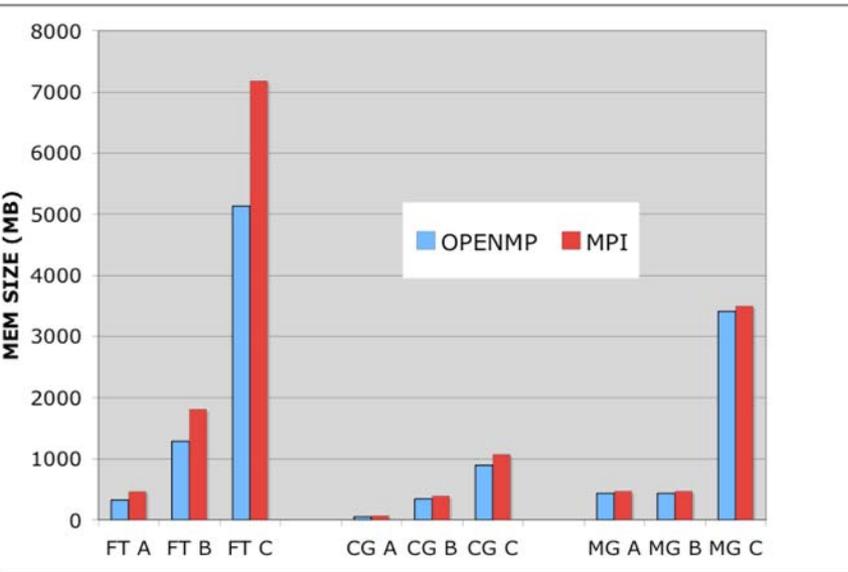
Intra-node Programming Model Requirements

- Express fine-grained parallelism in scalable manner (strong-scaling)
- Better abstracts notion of threads
- Does not make mapping of domain-to-core explicit to the programmer
- Ubiquitous
- As desirable on small machines as on the largest ones
- Able to elegantly express important scientific algorithms
- Scalable memory footprint

- **Flat model**
 - MPI
 - UPC

- **Hierarchical model (MPI+?)**
 - +UPC
 - +OpenMP
 - +Ct
 - +CUDA or OpenCL?

Performance



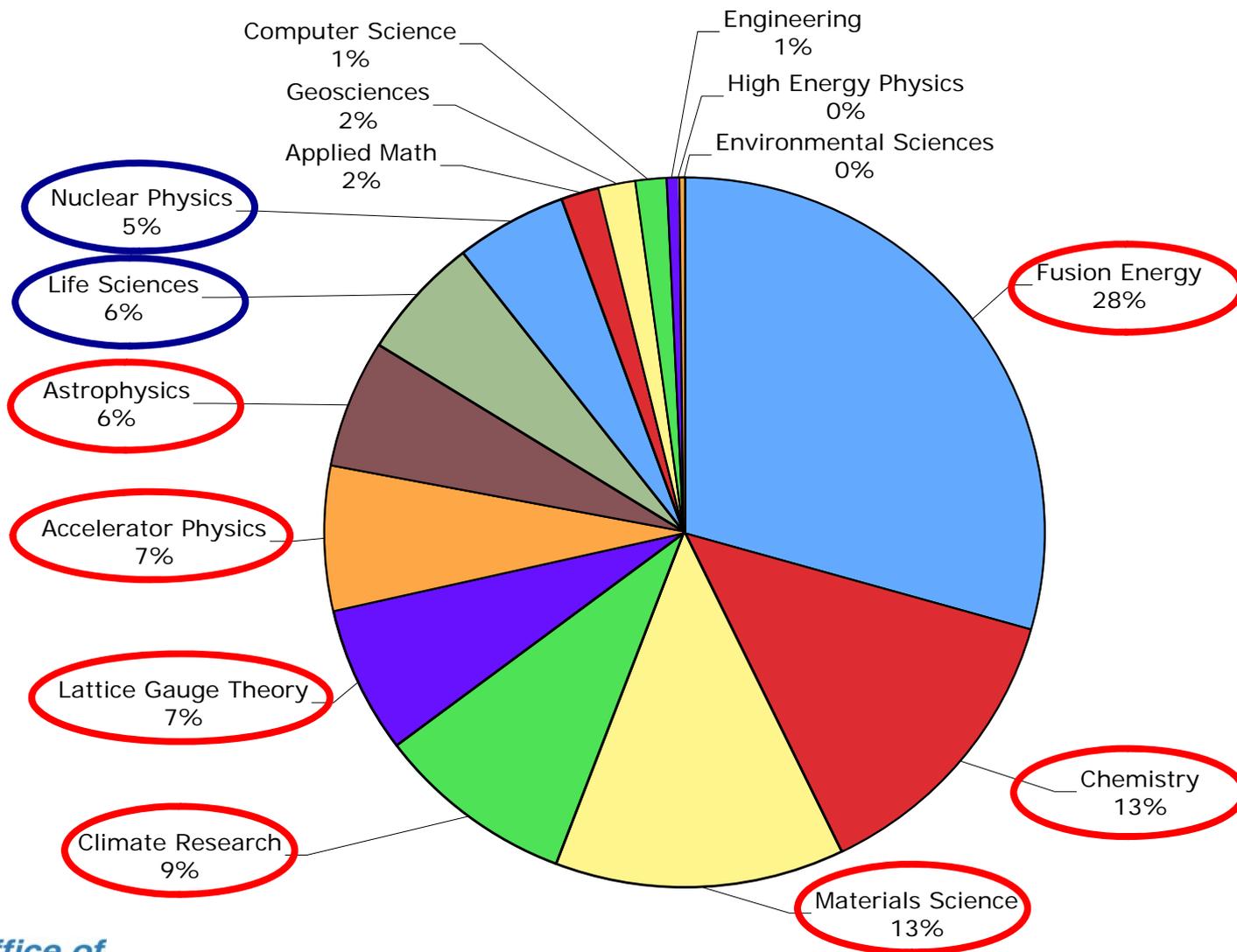
- **Memory Footprint**
 - MPI is higher in some cases
- **Performance**
 - OpenMP better on single socket
 - Worse on multi-socket
 - *UPC TBD*

Short Walk-Thru

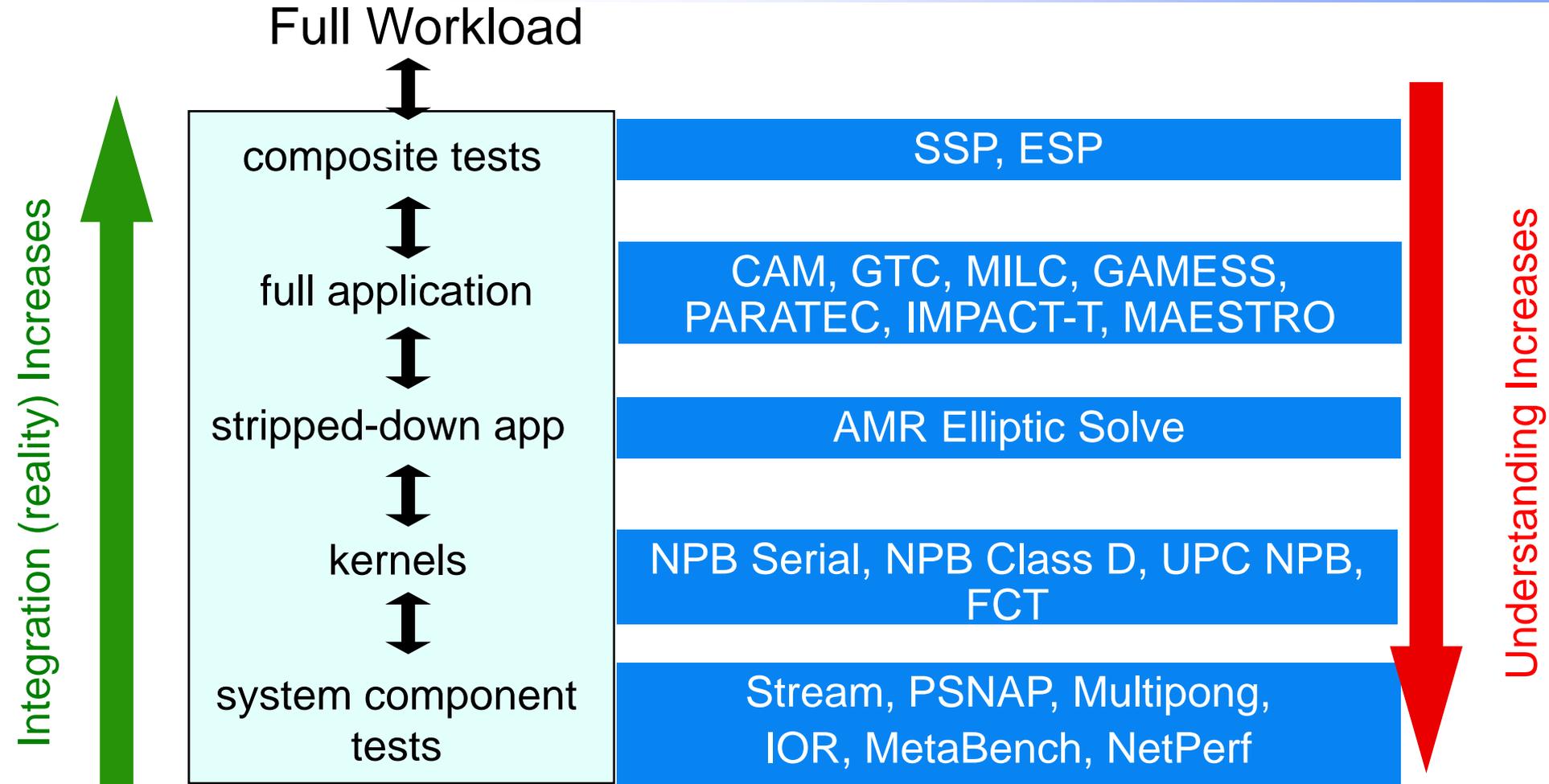
Balancing Requirements

- **NERSC Workload overview**
 - ~3000 users
 - ~300-400 projects representing a broad range of science
 - ~500-700 codes (~2 codes per project on average!)
 - 15 science areas for 6 Office of Science divisions
- **Select a subset (<10) codes to represent the requirements of the workload**
 - Contribution workload (workload coverage)
 - Contribution to each area of science (algorithm/science-area coverage)
- **Must cover algorithm usage across science areas**
 - Assumes evolving workload (don't alienate science areas)
 - Search for islands of coherence in the codes or algorithm selection by different scientific disciplines
 - *Still daunting*

Focus on Science Areas



Benchmark Hierarchy



Consistency is measured over all benchmarks as Coefficient of variation from 5 consecutive runs.

Benchmark Selection Criteria

- **Coverage**
 - Cover science areas
 - Cover algorithm space
- **Portability**
 - Robust ‘build’ systems
 - Not architecture specific implementation
- **Scalability**
 - Do not want to emphasize applications that do not justify scalable HPC resources
- **Open Distribution**
 - No proprietary or export-controlled code
- **Availability of Developer for Assistance/Support**

NERSC-6 Application Benchmarks

<i>Benchmark</i>	<i>Science Area</i>	<i>Algorithm Space</i>	<i>Base Case Concurrency</i>	<i>Problem Description</i>	<i>Lang</i>	<i>Libraries</i>
CAM	Climate (BER)	Navier Stokes CFD	56, 240 Strong scaling	D Grid, (~.5° resolution); 240 timesteps	F90	netCDF
GAMESS	Quantum Chem (BES)	Dense linear algebra	384, 1024 (Same as Ti-09)	DFT gradient, MP2 gradient	F77	DDI, BLAS
GTC	Fusion (FES)	PIC, finite difference	512, 2048 Weak scaling	100 particles per cell	F90	
IMPACT-T	Accelerator Physics (HEP)	PIC, FFT	256,1024 Strong scaling	50 particles per cell	F90	
MAESTRO	Astrophysics (HEP)	Low Mach Hydro; block structured-grid multiphysics	512, 2048 Weak scaling	16 32³ boxes per proc; 10 timesteps	F90	Boxlib
MILC	Lattice Gauge Physics (NP)	Conjugate gradient, sparse matrix; FFT	256, 1024, 8192 Weak scaling	8x8x8x9 Local Grid, ~70,000 iters	C, assem.	
PARATEC	Material Science (BES)	DFT; FFT, BLAS3	256, 1024 Strong scaling	686 Atoms, 1372 bands, 20 iters	F90	Scalapack, FFTW

Algorithm Diversity

<i>Science areas</i>	<i>Dense linear algebra</i>	<i>Sparse linear algebra</i>	<i>Spectral Methods (FFT)s</i>	<i>Particle Methods</i>	<i>Structured Grids</i>	<i>Unstructured or AMR Grids</i>
Accelerator Science		X	X	X	X	X
Astrophysics	X	X	X	X	X	X
Chemistry	X	X	X	X		
Climate			X		X	X
Combustion					X	X
Fusion	X	X		X	X	X
Lattice Gauge		X	X	X	X	
Material Science	X		X	X	X	

NERSC users require a system which performs adequately in all areas

N6 Benchmarks Coverage

Science areas	Dense linear algebra	Sparse linear algebra	Spectral Methods (FFT)s	Particle Methods	Structured Grids	Unstructured or AMR Grids
Accelerator Science		X	X IMPACT-T	X IMPACT-T	X IMPACT-T	X
Astrophysics	X	X MAESTRO	X	X	X MAESTRO	X MAESTRO
Chemistry	X GAMESS	X	X	X		
Climate			X CAM		X CAM	X
Combustion					X MAESTRO	X AMR Elliptic
Fusion	X	X		X GTC	X GTC	X
Lattice Gauge		X MILC	X MILC	X MILC	X MILC	
Material Science	X PARATEC		X PARATEC	X	X PARATEC	

NERSC-6 Composite SSP Metric

The largest concurrency run of each full application benchmark is used to calculate the composite SSP metric

NERSC-6 SSP

CAM
240p

GAMESS
1024p

GTC
2048p

IMPACT-T
1024p

MAESTRO
2048p

MILC
8192p

PARATEC
1024p

For each benchmark measure

- FLOP counts on a reference system*
- Wall clock run time on various systems*

Example of N6 SSP on Hypothetical System

Hypothetical N6 System	Results			
	Tasks	System Gflopcnt	Time	Rate per Core
CAM	240	57,669	408	0.589
GAMESS	1024	1,655,871	2811	0.575
GTC	2048	3,639,479	1493	1.190
IMPACT-T	1024	416,200	652	0.623
MAESTRO	2048	1,122,394	2570	0.213
MILC	8192	7,337,756	1269	0.706
PARATEC	1024	1,206,376	540	2.182
GEOMETRIC MEAN				0.7

Rate Per Core =
Ref. Gflop count /
(Tasks*Time)

Flop count
measured on
reference
system

Measured wall
clock time on
hypothetical
system

Geometric
mean of
'Rates per
Core'

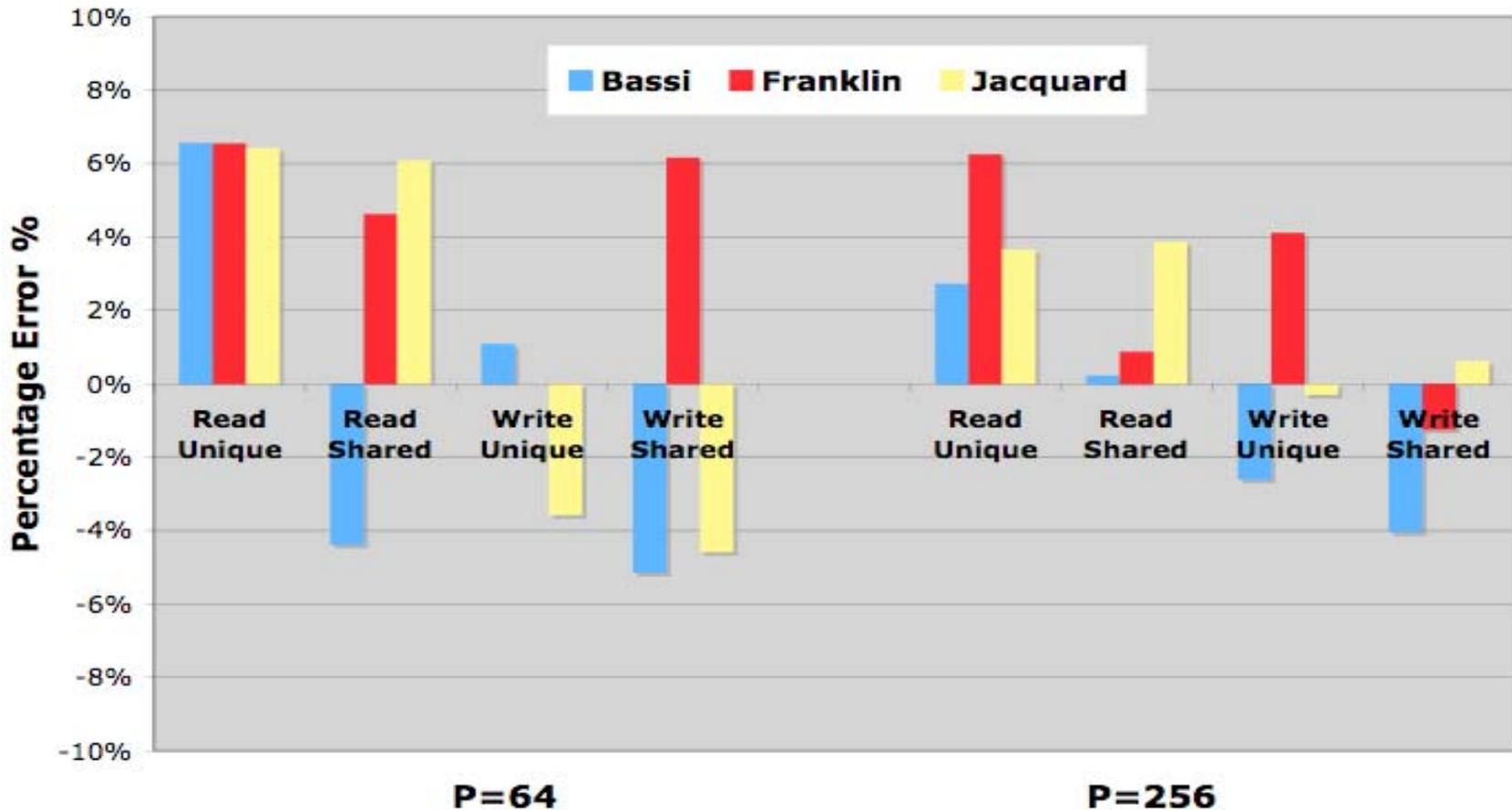
SSP (TF) = Geo mean of rates per core * # cores in system / 1000

N6 SSP of 100,000 core system = $0.7 * 100,000 / 1000 = 70$

N6 SSP of 200,000 core system = $0.7 * 200,000 / 1000 = 140$

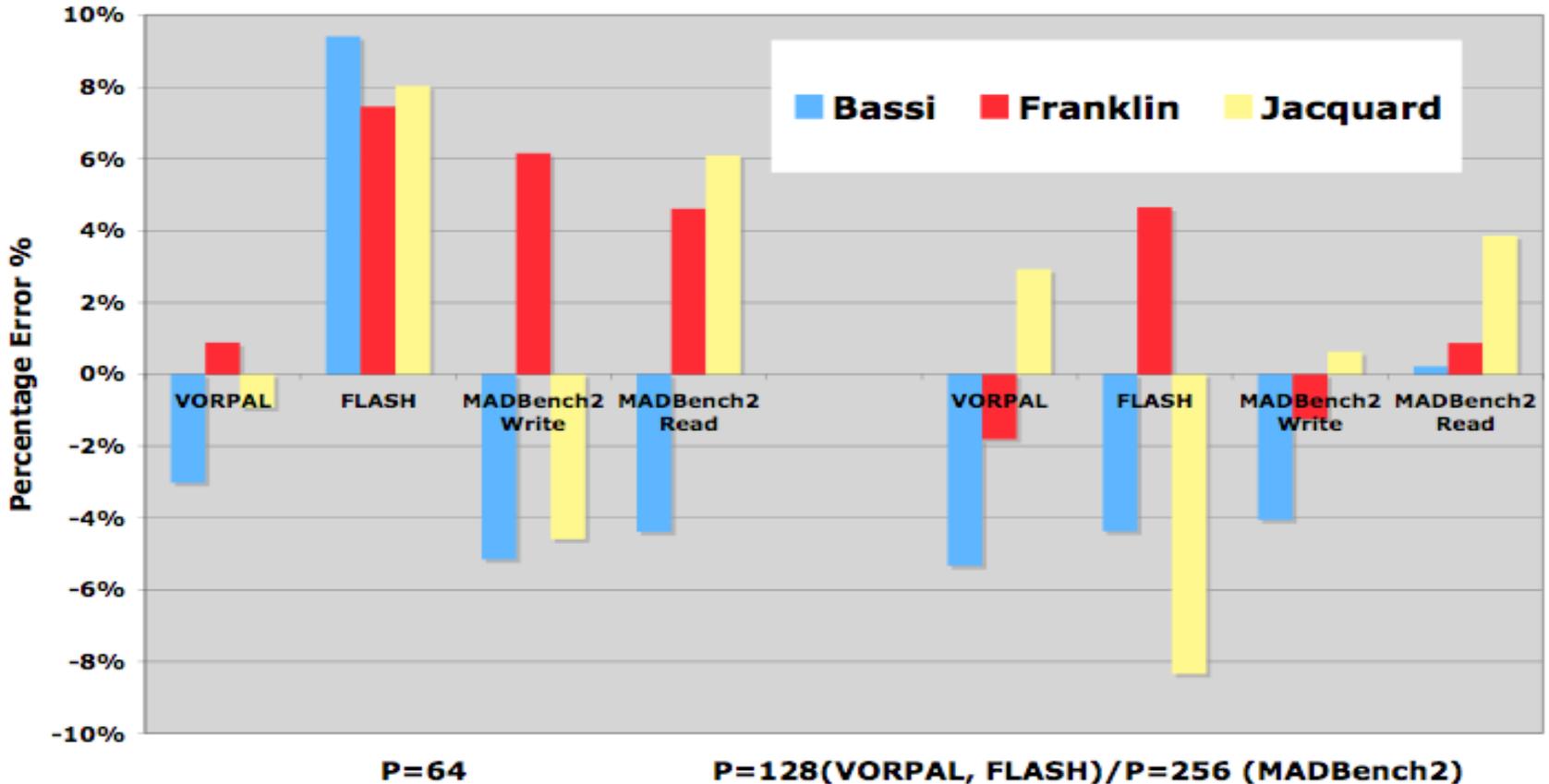
Allows vendors to size systems based on benchmark performance

I/O Performance Prediction for MADBench2



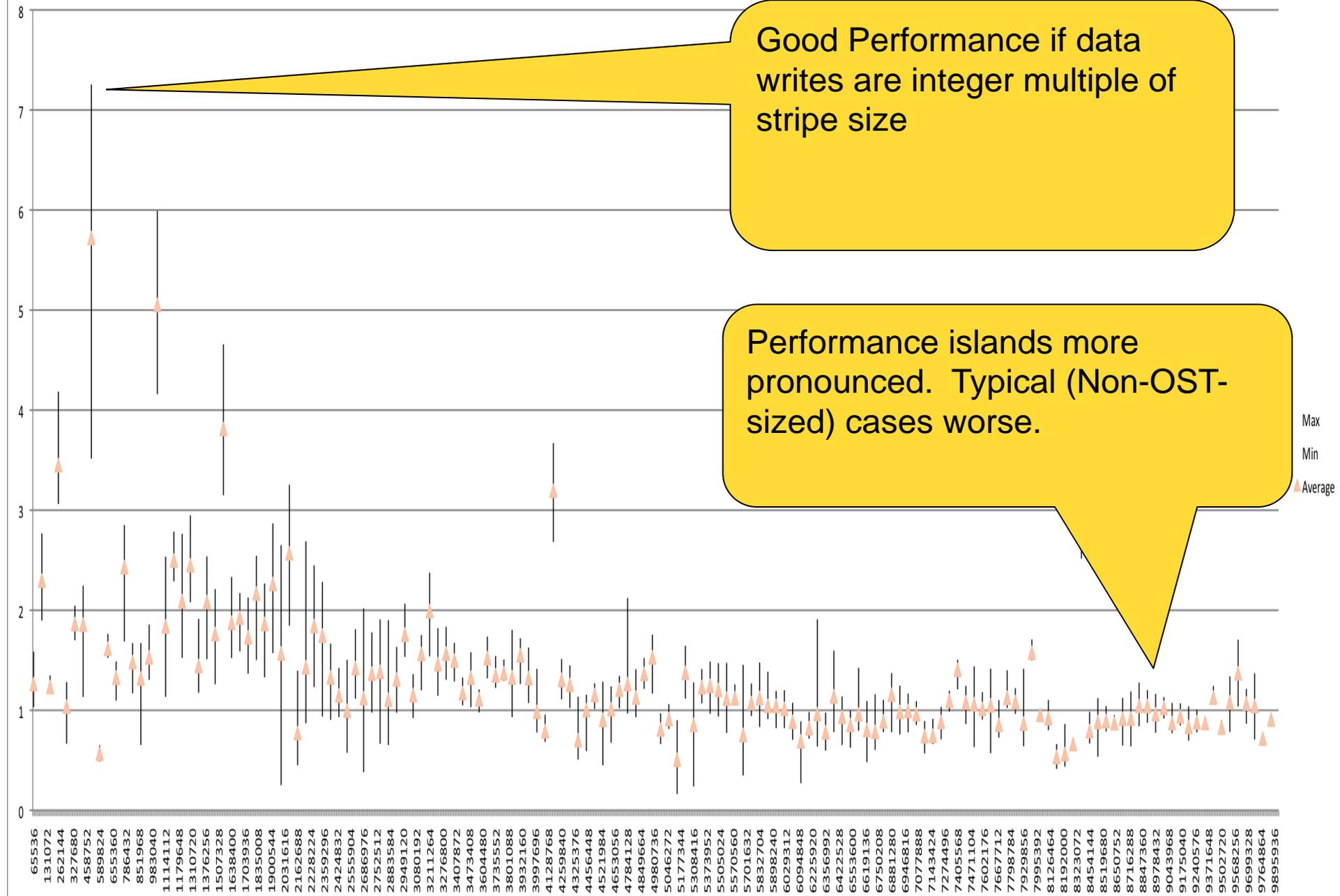
IOR captures essential features of madbench2 IO behavior, both access patterns and performance.

I/O Performance Prediction



- Prediction error is within 10% in the worst case

80 processors with 80 OSTs (Shane case)

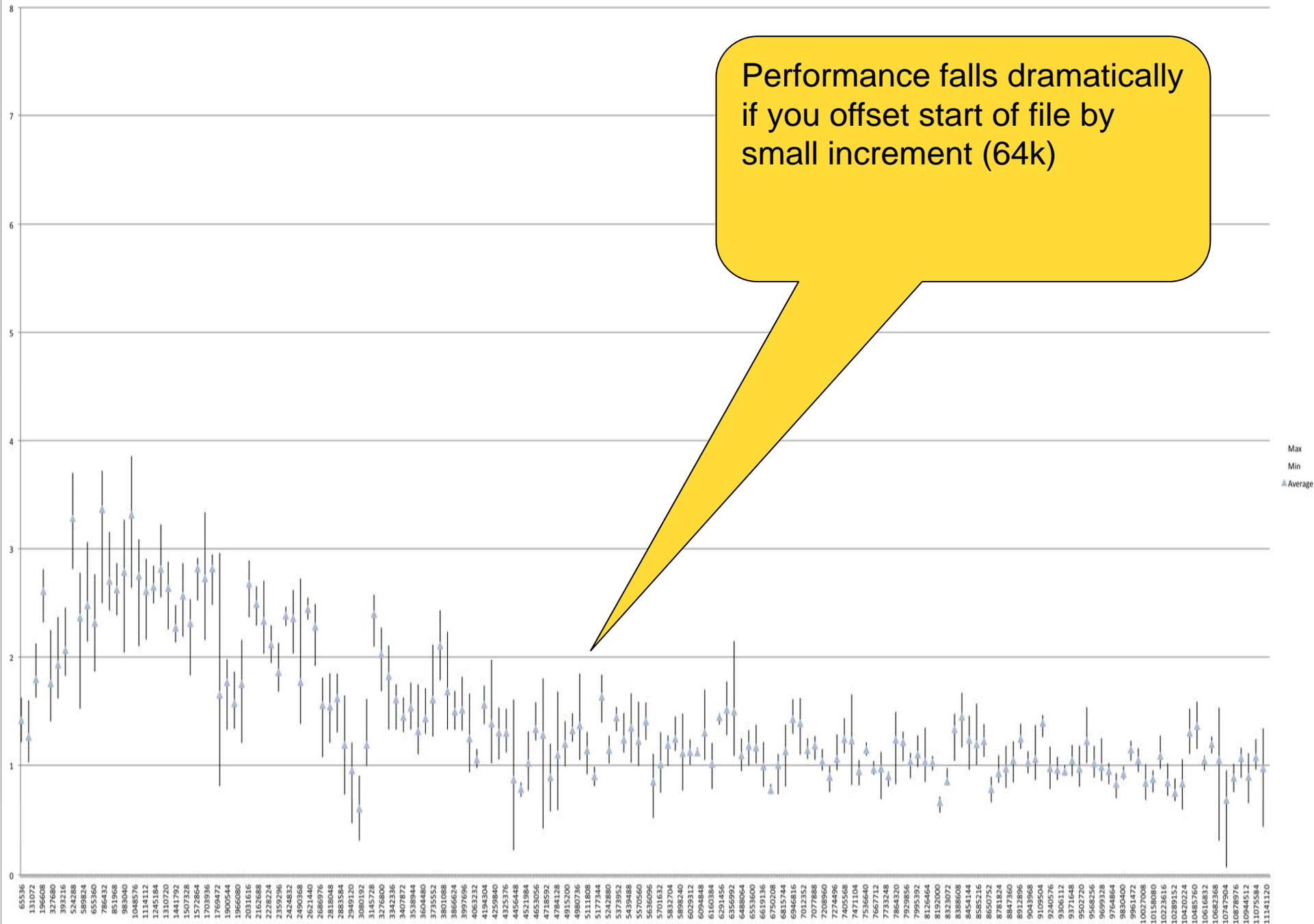


Good Performance if data writes are integer multiple of stripe size

Performance islands more pronounced. Typical (Non-OST-sized) cases worse.

80 processors with 40 OST : offset file start by 64k

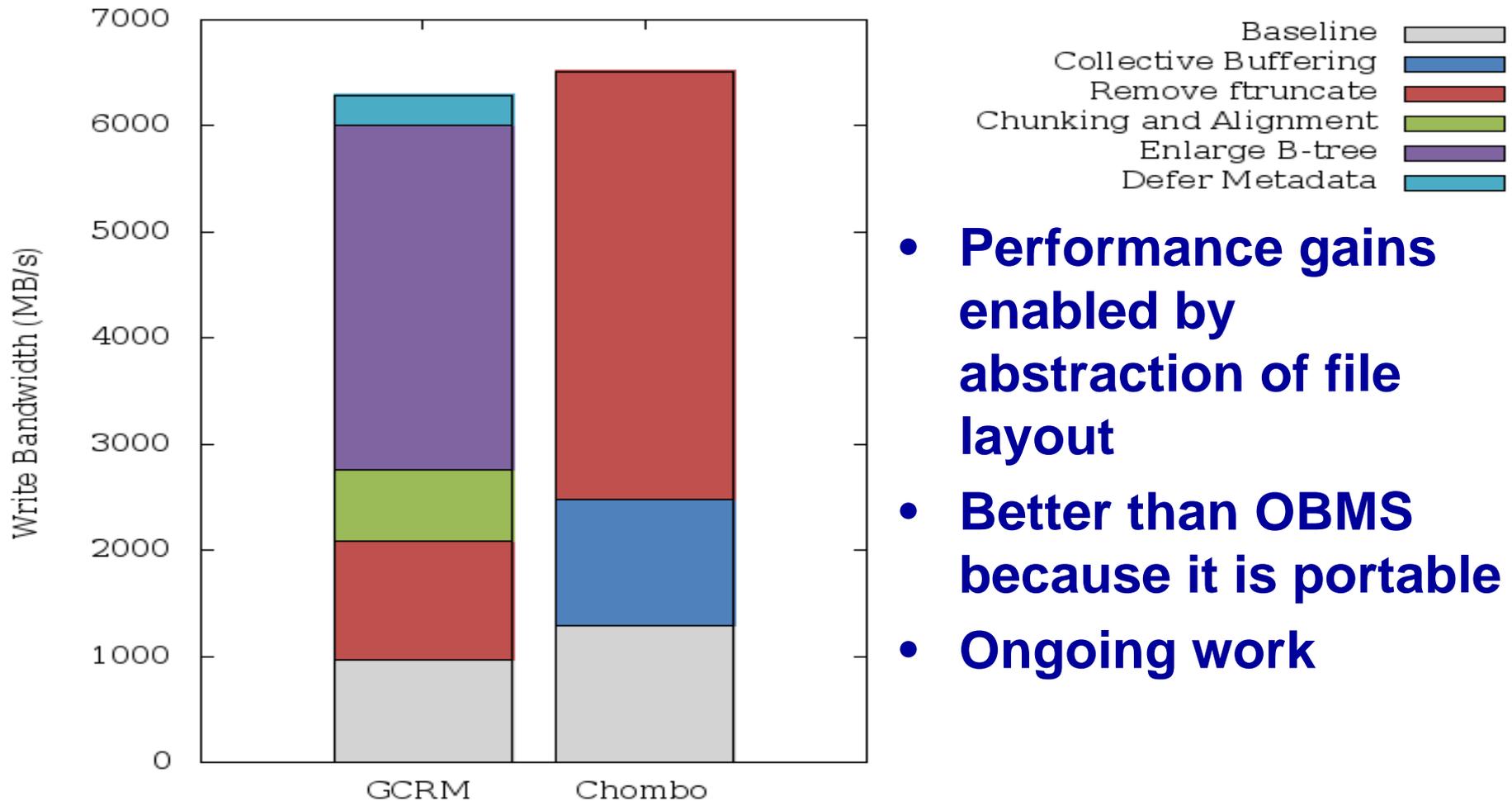
Performance falls dramatically if you offset start of file by small increment (64k)



Performance Tuning Results

(first two applications)

HDF I/O Tuning



- Performance gains enabled by abstraction of file layout
- Better than OBMS because it is portable
- Ongoing work