BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Is There a Life After the Top500?

(Or What to Do About the Top Problems with the TOP500 List)

William Kramer

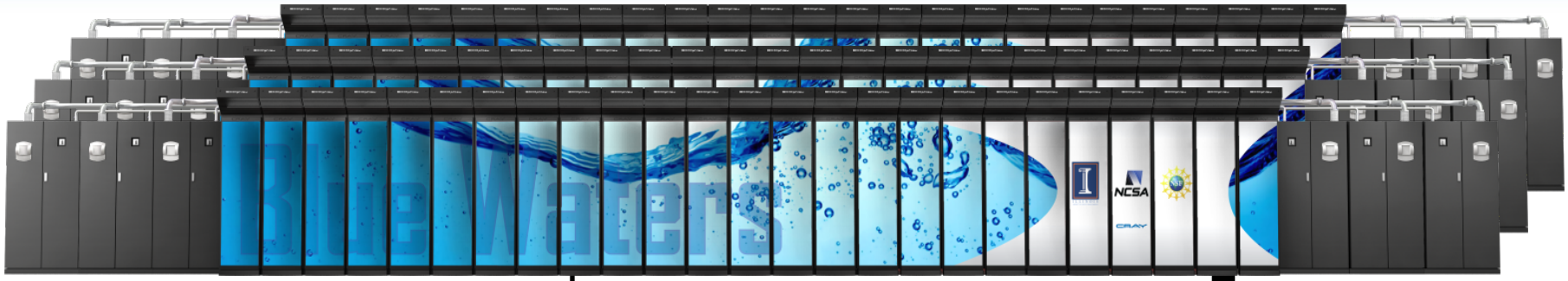
National Center for Supercomputing Applications, University of Illinois

NCSA NSF GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

A Few Words About Blue Waters

- The system has been fully connected since the mid summer.
  - Scale testing and diagnosis was useful and interesting
  - Most applications ran at large scale with little problem
  - I/O testing was impressive with 1300 controllers
- By early October, all components were installed
  - 1600 storage controllers
  - Kepler K20X GPUs
- Began testing the system soon after that has continued
  - Over 300 hundred tests are explicitly called for – some taking days
- “Friendly User Period started in early November

Blue Waters Computing System



Aggregate Memory – 1.5 PB

10/40/100 Gb Ethernet Switch

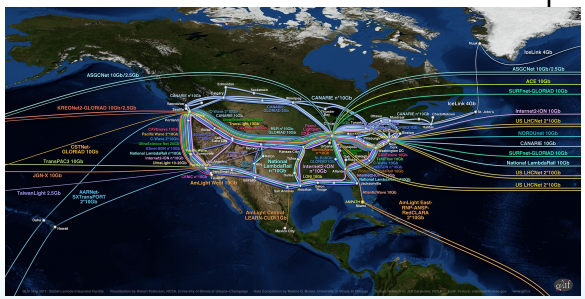
External Servers

IB Switch

>1 TB/sec

120+ Gb/sec

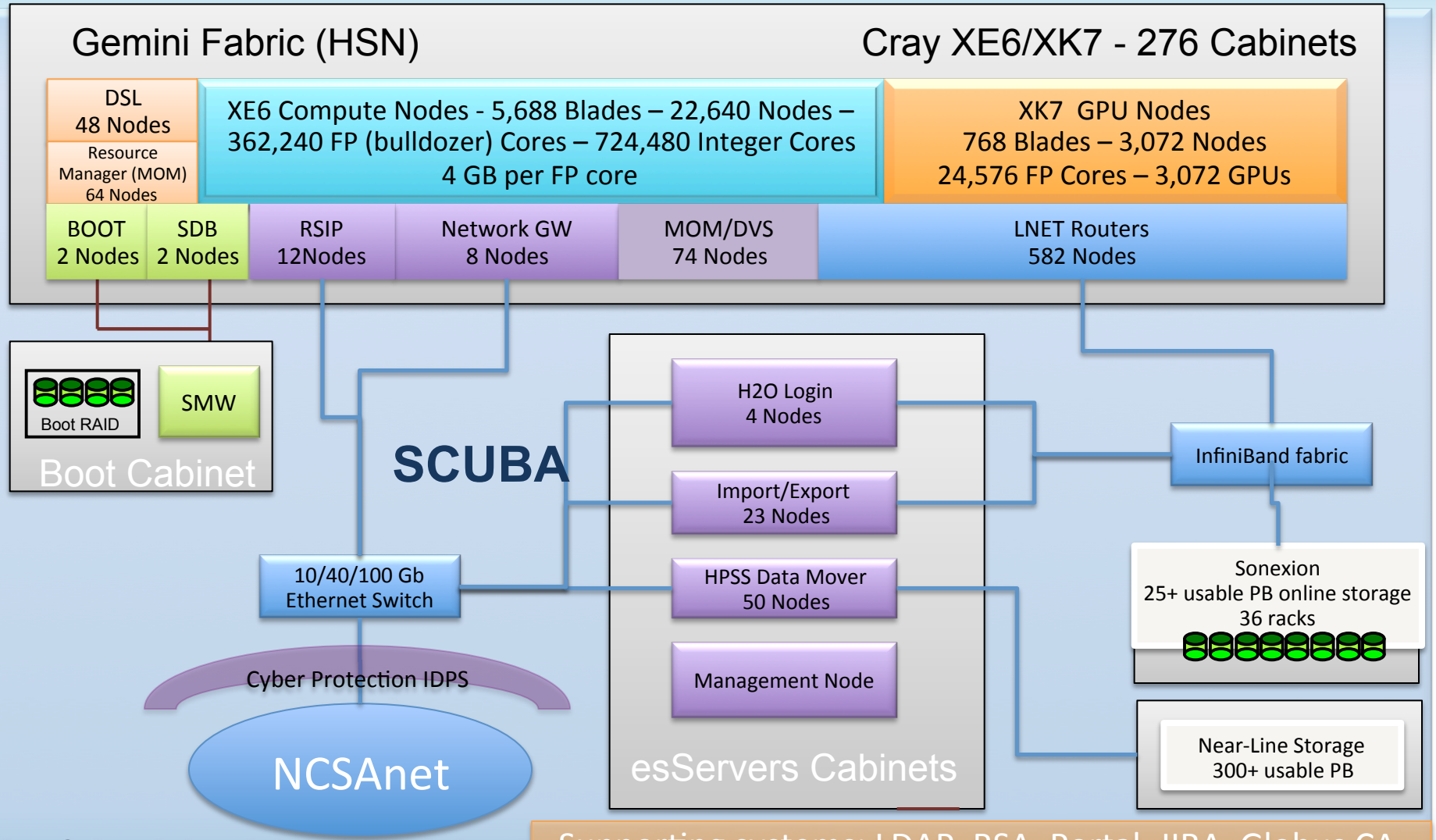
100 GB/sec



100-300 Gbps WAN

Spectra Logic: 300 usable PB

Sonexion: 26 usable PB


Gemini Fabric (HSN)
Cray XE6/XK7 - 276 Cabinets
DSL
48 Nodes
Resource Manager (MOM)
64 Nodes
XE6 Compute Nodes - 5,688 Blades – 22,640 Nodes –
362,240 FP (bulldozer) Cores – 724,480 Integer Cores
4 GB per FP core
XK7 GPU Nodes
768 Blades – 3,072 Nodes
24,576 FP Cores – 3,072 GPUs
BOOT
2 Nodes
SDB
2 Nodes
RSIP
12Nodes
Network GW
8 Nodes
MOM/DVS
74 Nodes
LNET Routers
582 Nodes
Boot RAID
SMW
Boot Cabinet
SCUBA
H2O Login
4 Nodes
Import/Export
23 Nodes
HPSS Data Mover
50 Nodes
Management Node
esServers Cabinets
InfiniBand fabric
Sonexion
25+ usable PB online storage
36 racks
10/40/100 Gb
Ethernet Switch
Cyber Protection IDPS
NCSAnet
Near-Line Storage
300+ usable PB
NPCF
Supporting systems: LDAP, RSA, Portal, JIRA, Globus CA,
Bro, test systems, Accounts/Allocations, CVS, Wiki

BW Sustained Performance Measures

- Original NSF Benchmarks
  - Full Size – QCD (MILC), Turbulence (PNSDNS), Molecular Dynamics (NAMD)
  - Modest Size – MILC, Paratec, WRF
- SPP expands the original requirements as it is a time to solution metric that is using the planned applications on representative parts of the science team problems
  - Represents end to end problem run including I/O, pre and post phases, etc.
  - Coverage for science areas, algorithmic methods, scale
- SPP Application full applications (details and method available)
  - NAMD – Molecular Dynamics; MILC, Chroma – Lattice Quantum Chromodynamics; VPIC, SPECFEM3D – Geophysical Science; WRF – Atmospheric Science; PPM – Astrophysics; NWCHEM, GAMESS – Computational Chemistry; QMCPACK – Materials Science
- The input, problem sizes, included physics, and I/O performed by each benchmark is comparable to the simulations proposed by the corresponding science team for scientific discovery.
- Well defined reference operation counts used to represent work across disciplines
- Each benchmark sized to use one-fifth to one-half of the number of nodes in the full system.
  - At least three SPP applications run at full system size

Latest Status

- All NSF benchmarks have been run and are performing within expectations
  - 3 non Petascale Applications
  - 3 Petascale Applications have run at > 25,000 nodes
    - Timing includes all necessary I/O (science and defensive) and necessary checkpoints

Application	Size	Hours	Restarts
Turbulence	25000+	72	3
MILC	25000+	52	2
NAMD	25000+	16	2

- All 8 Interlagos Sustained Petascale Performance tests running at scale
  - Timing includes all necessary I/O (science and defensive) and necessary checkpoints
  - Time to solution geometric mean > 1 PF as of 11/19/12
  - Four tests are running > 1 PF at full system scale!
  - Substantial improvements made to science codes and are being provided back to science teams
- Three of four GPU SPP tests running well within requirements at scales > 700 nodes
- Aggregate I/O performance measured with multiple trials at more than 1.1 TB/s
- Reliability and resiliency better than projected

WHY THE HPC COMMUNITY NEEDS TO DEAL WITH THE DISCONNECT BETWEEN TOP500 AND USABLE PERFORMANCE

LINPACK is a single test that solves Ax=b with dense linear equations using Gaussian elimination with partial pivoting. matrix A, that is size M x M, LINPACK requires $2/3\ M^2 + 2M^2$ operations. $O(N^2)$ memory and $O(N^3)$ Floating Point operations

Some Top500 Issues To Address

See PACT 12 for full discussion

- The TOP500 gives no indication of the cost or value of a system
- The TOP500 encourages organizations to make poor choices
- The TOP500 provides little historical value
- The TOP500 is dominated by who has the most money to spend–not what system is the best.
- The Linpack TOP500 measure takes too long to run and does not represent strong scaling
- The TOP500 metric has not kept up with changing algorithmic methods.
- The TOP500 Linpack performance test is dominated by single-core, dense linear algebra peak performance
- There is no relationship between the TOP500 ranking and real work potential, user productivity, system usability for real application workloads.
  The TOP500 list disenfranchises many important application areas.
- The Linpack benchmark serves only one or two of the four purposes of a good benchmark.

BLUE WATERS DECIDED NOT TO PARTICIPATE IN THE TOP 500 LIST BECAUSE THE BLUE WATERS MISSION IS SUSTAINED PERFORMANCE

Just to be Clear

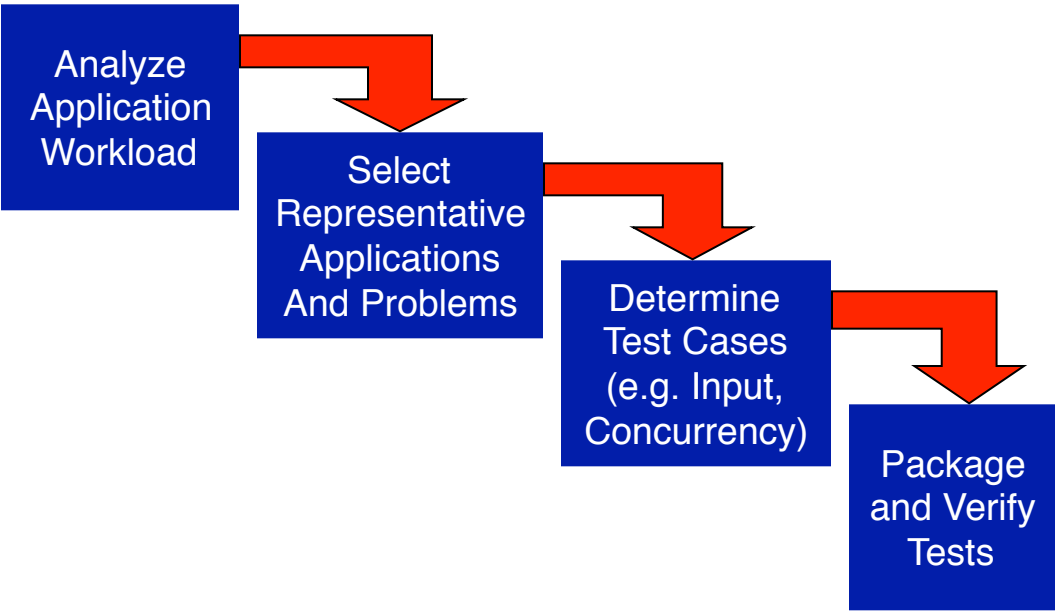
- 25 Science Teams are now 'friendly users'
- Blue Waters is working well and is running actual applications up to full scale
  - Expect to be in full service in early 2013
- If Blue Waters focused only on Peak/Linpack Flops we could have been #1 for at least several cycles
  - Would have very small memory and very little storage
- The NSF and UIUC made the decision was made early in 2012 – even before ESS was complete
- Blue Waters has run HPCC and HPL and we know exactly where we would have been on the list
  - But we won't say what it is

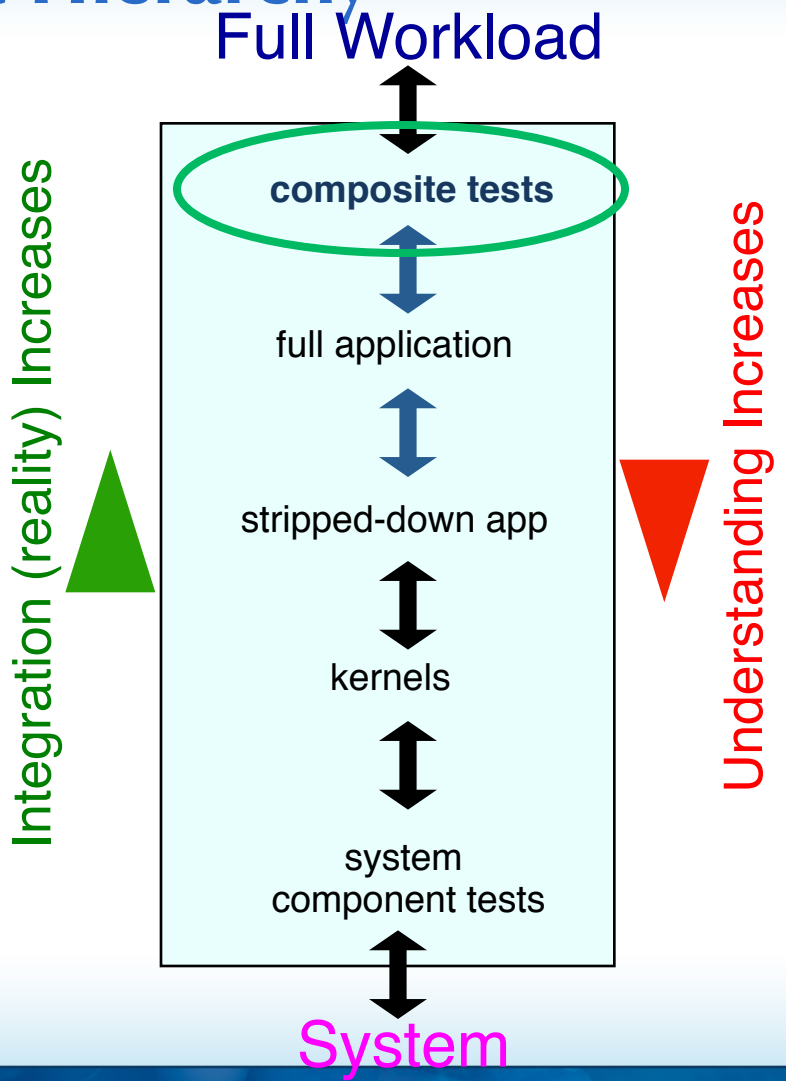
SO – WHO WANTS TO HELP CREATE AN IMPROVED METRIC METHOD?

POTENTIAL IMPROVED IDEAS – ONE SHOULD NOT CRITICIZE WITHOUT SUGGESTING BETTER ALTERNATIVES

Evolutionary and Revolutionary

Benchmark and Test Hierarchy





Slide Courtesy of Katie Antypas and the NERSC-6 Team

Time to Solution is THE Metric

- The consensus of many papers/experts is the only real, meaningful metric that can compare systems or implementations is the time it takes to solve a defined, real problem on systems.
  - Work is a task to carry out or a problem to solve
  - Just like in the real world, work is not a rate, it is not a speed, it is a quantity
- The work is meaningful effort, not overhead work or useless work
- Hence a good evaluation compares how much time it takes to do an amount of meaningful (productive) work
  - Referred to as the System's Potential to do the work
  - Cost effectiveness = system's potential/system's cost
    - Cost can have many components as well

Time to Solution is THE Metric (cont)

- Time to Solution comparisons have their own challenges
  - Defining what the work is in an discrete manner (i.e. data input set)
  - Defining the work process(es) (application/algorithm/ code path…)
  - Picking a unit to represent the work
  - Defining work across disciplines for multi use systems
  - Defining useful work vs overhead work (to parallelize, to move data, to set up, key steps)
  - Balancing practical issues
    - Complexity, testable system size, tractable length the test runs, number of tests, quality of implementation, optimizations

EVOLUTIONARY

Can be implemented immediately with little effort

Evolutionary Changes that will make the Top500 more meaningful

1. *Require (estimated) cost data be posted for every system listed*
2. *Do not allow a system to be listed until it is fully accepted and performing its mission*
3. *Require a complete description for every system listed to give information about the investment balance*
4. *Move from weak scaling to strong scaling Linpack*
   - *Could use size classes as NPBs do to address large range of system scale*

REVOLUTIONARY

Align Our Community Metric To Best Practices In Benchmarking

- Combining the criteria from (Smith,1988) and (Lilja, 2000) provides the following list of good attributes for benchmarks
- Proportionality – a linear relationship between the metric used to estimate performance and the actual performance. In other words, if the metric increases by 20%, then the real performance of the system should be expected to increase by a similar proportion.
  - A scalar performance measure for a set of benchmarks expressed in units of time should be directly proportional to the total time consumed by the benchmarks.
  - A scalar performance measure for a set of benchmarks expressed as a rate should be inversely proportional to the total time consumed by the benchmarks.
- Reliability means if the metric shows System A is faster than System B, it would be expected that System A outperforms System B in a real workload represented by the metric.
- Consistency so that the definition of the metric is the same across all systems and configurations.
- Independence so the metric is not influenced by outside factors such as a vendor putting in special instructions that just impact the metric and not the workload.
- Ease of use so the metric can be used by more people.
- Repeatability meaning that running the test for the metric multiple times should produce close to the same result.

Align the community metric to best practices in benchmarking (cont)

David Bailey – 12 Ways to Fool the Masses – 1991

1. Quote only 32-bit performance results, not 64-bit results.
2. Present performance figures for an inner kernel, and then represent these figures as the performance of the entire application.
3. Quietly employ assembly code and other low-level language constructs.
4. Scale up the problem size with the number of processors, but omit any mention of this fact.
5. Quote performance results projected to a full system.
6. Compare your results against scalar, unoptimized code on conventional systems.
7. When direct run time comparisons are required, compare with an old code on an obsolete system.
8. If Mflop/s rates must be quoted, base the operation count on the parallel implementation, not on the best sequential implementation.
9. Quote performance in terms of processor utilization, parallel speedups or Mflop/s per dollar.
10. Mutilate the algorithm used in the parallel implementation to match the architecture.
11. Measure parallel run times on a dedicated system, but measure conventional run times in a busy environment.
12. If all else fails, show pretty pictures and animated videos, and don't talk about performance.

David's Update for 2011

A. Cite performance rates for a run with only one processor core active in a shared-memory multi-core node. For example, cite performance on 1024 cores, even though the code was run on 1024 nodes, wasting 15 out of 16 cores on each node.

B. Cite performance rates only for a core algorithms (such as FFT or LU decomposition), even though the paper mentions one or more full-scale applications that were done on the system.

C. List only the best performance figure in the paper, even though the run was made numerous times.

D. Employ special hardware, operating system or compiler settings that are not appropriate for real-world usage.

E. Define "scalability" as successful execution on a large number of CPUs, regardless of performance.

- http://crd.lbl.gov/~dhbailey/dhbtalks/dhb-12ways.pdf

Align the community metric to best practices in benchmarking (cont)

David's Further Advice to Prevent Abuse

- Direct comparisons of run times on real applications are preferred.
- If results are presented for a well-known benchmark, established ground rules must be followed.
- Only actual performance results should be presented, not projections or extrapolations (unless very clearly disclosed and justified).
- Performance figures should be based on comparable levels of tuning.
- Mflop/s, Gflop/s, Tflop/s rates should be computed from operation counts based on the best practical serial algorithms.
- When computing parallel speedup figures, the denominator rate should be based on an efficient single-processor implementation.
- Any ancillary information that would affect the interpretation of the results should be fully disclosed (e.g., the use of 32-bit instead of 64-bit data, etc.).
- Special care should be taken for figures and graphs.
- Whenever possible, full background information should be provided: algorithms, hardware and software configuration, language, compiler flags, tuning, timing method, basis for operation counts, etc.

Bill's Addendum

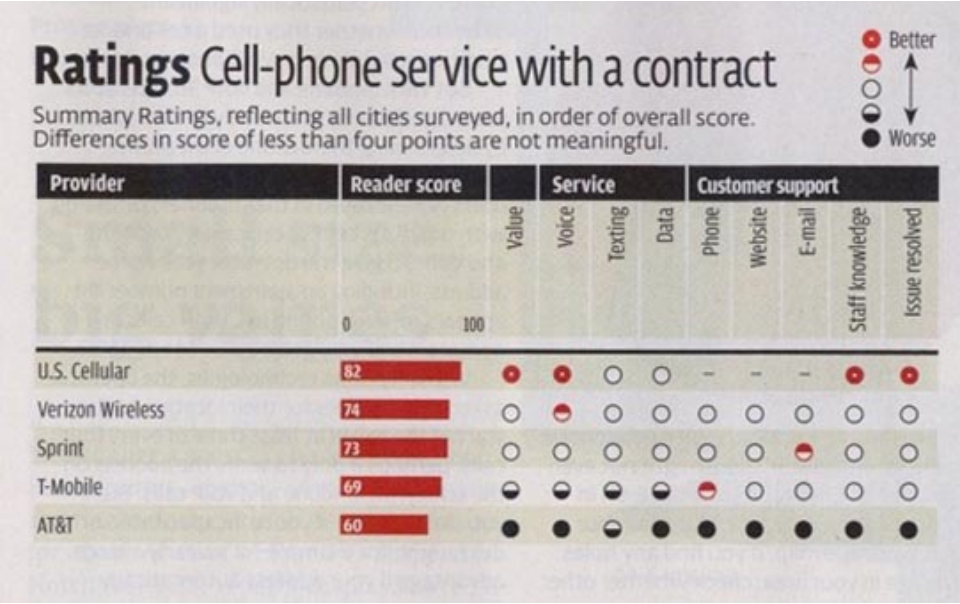
- Use all the tests from a suite unless you can show they are not relevant
- Consider Base, Optimized and Redesigned Cases
- Tests have to be kept vital – refreshed
- Evaluations have to be able to address value
  - Value = potential for work/cost function
  - NPBs did this well for a long time but few others do
    - Green500 uses only 1 surrogate for TCO – not sufficient
- Consistency should be reported
  - Standard Deviation or CoV
  - OS interference studies for single SMP
    - Inconsistency is not the same as OS Jitter

Aggregate The HPCC Metrics Into A Single Value

- If we need a single number for elevator speeches/press then this is better
- Most of the HPCC measures can be generalized into a single composite measure.
- While each test assesses different characteristics, with the exception of *randomring* latency, the HPCC tests all have one thing in common:
  - They do a certain amount of work in a given time period and the work unit is encapsulated in some activity.
  - Each test has a given amount of work represented by the total number of reference actions that have to be carried out to complete the test.
  - If the number of actions is deterministic, then the difference between two rates is only the wall clock time it takes to carry out all those actions.
  - An appropriate composite function (various means or combinations) can result so that time-to-solution for each test is the only variable.
- Again - time-to-solution is the only real measure of performance.
- Would provide the community with a single "actions/second" measure more representative of realistic system achievable performance.

Individualized rankings

- Bill Gropp calls this the “Consumer Report” style
- Provide a set of well explained evaluation measures
- Provide a more complete description of system components
- Provide some pre-defined combination rankings
- Allow individuals to query and build their own ranking based on what is important to their needs



Ratings Cell-phone service with a contract

Summary Ratings, reflecting all cities surveyed, in order of overall score. Differences in score of less than four points are not meaningful.

Better / Worse

Provider	Reader score (0–100)		Service			Customer support				
		Value	Voice	Texting	Data	Phone	Website	E-mail	Staff knowledge	Issue resolved
U.S. Cellular	82	●	●	○	○	–	–	–	●	●
Verizon Wireless	74	○	◒	○	○	○	○	○	○	○
Sprint	73	○	○	○	○	○	○	◒	○	○
T-Mobile	69	◒	◒	◒	◒	◒	○	○	○	○
AT&T	60	●	●	◒	●	●	●	●	●	●

Weighted composite of actions

- Combining improvements 2 and 3 may greatly improve the realism of a single metric
- As a thought experiment, assume there are N measures A_n (from the HPCC or other tests):
  - Example uses N = 3
    - One for an arithmetic operation for a quantum of data values,
    - One for moving a quantum of data between memory locations, and
    - One for moving a quantum of data across the interconnect.
  - Determine the number of actions each measure carries out, Na_n, Na_2, and Na_3, and measure the time to complete the amount of actions, shown as t_1, t_2 and t_3.
  - Rate of actions is $\Phi(Na_1/t_1, Na_2/t_2, Na_3/t_3)$ where Φ is a composite function (e.g. arithmetic or geometric mean)
  - Since $Na_{1\text{-}n}$ are deterministic, the only variables are the times to completing the work, giving this approach the proportional and other properties discussed above.
- Can use a weighted composite function, with weights w_1, w_2 and w_3.
  - Since systems today may have 100 to 1,000 more flops than interconnect bandwidth and 10 to 100 times more flops than memory bandwidth, it is possible to use 1, 10 and 100 for the weighting factors.
  - Using weights, particularly with some study matching the best weights to workloads, would provide a more realistic single indicator for real (sustained) performance

Create A New, Meaningful Suite Of Benchmarks

- Many benchmark suites that were held in high regard (Livermore Loops, NPBs, SPEC) over time are suites of pseudo and/or full applications.
- While the best case for any benchmark is to be a statistically representative sample of real workload, in realty, this is not possible for community tests.
- SERPOP (Sample Estimation of Relative Performance of Programs) method is best suited for a generalized test.
  - A sample of a workload is selected to represent a workload. However, the sample is not random and cannot be considered a statistical sample.
  - SERPOP methods occur frequently in performance analysis and reflect very meaning measures that span individual communities.
  - In SERPOP analysis, the workload is related to SERPOP tests, but does not indicate the frequency of usage or other characteristics of any individual workload.
- Many common benchmark suites—including SPEC, TCP and NPB, as well as many acquisition test suites—are SERPOP.

Mashey, John R. "War of the Benchmark Means: Time for a Truce." ACM SIGARCH Computer Architecture News (Association for Computing Machinery) 32, no. 4 (September 2004)

SERPOP Example – The Blue Waters Sustained Petascale Performance (SPP) Method (not a single benchmark)

- Establish a set of application codes that reflect the intended work the system will do
  - Can be any number of tests as long as they have a common measure of the amount of work
- A test consists of a complete code and a problem set reflecting the science teams' intentions
- Establish the reference amount work (ops, atoms, years simulated, etc.) the problem needs to do for a fixed concurrency
- Time each test takes to execute
  - Concurrency and/or optimization can be fixed and/or varied as desired
- Determine the rate of work done for a given "schedulable unit" (node, socket, core, task, thread, interface, etc.)
  - Work = Total work (reference operations) /total time/number of scalable units
  - Work per unit= Total work/number of scalable units used for the test
- Composite the work per schedulable unit for all tests
  - Composite functions based on circumstances and test selection criteria
  - Can be weighed or not as desired
  - BW is using the Geometric mean – lowest of all means and reduces impact of outliers
- Determine the SPP of a system by multiplying the composite work per schedulable unit by the number of schedulable units in the system
- Determine the *Sustained Petascale Performance*

$$SPP_{s,k} = \sum_{\alpha=1}^{A_{s,k}} \left(\Phi \left(W \ , P_{s,k,\alpha} \right) * N_{s,k,\alpha} \right)$$

SPP Method Coverage

Science Area	Struct Grids	Unstruct Grids	Dense Matrix	Sparse Matrix	N-Body/ Agent	Monte Carlo	FFT	PIC	Significant I/O
Climate and Weather	X	X		X		X			X
Plasmas/Magnetosphere	X				X		X		X
Stellar Atmospheres and Supernovae	X			X	X	X		X	X
Cosmology	X			X	X				
Combustion/Turbulence	X						X		
General Relativity	X			X					
Molecular Dynamics			X		X		X		
Quantum Chemistry			X	X	X	X			X
Material Science			X	X	X	X			
Earthquakes/Seismology	X	X			X				X
Quantum Chromo Dynamics	X		X	X	X		X		
Social Networks					X				
Evolution									
Engineering/System of Systems						X			
Computer Science		X	X	X			X		X

There are Many Other Possibilities

- Should learn from past attempts
- Should reflect community wide concerns
- Have multiple uses
- Reflect multiple points of view
- Tractable and portable implementations
- Can evolve over time but remain backward relatable

Summary

- Blue Waters is in excellent shape and already done unique science
- Petascale+ systems present significant challenges for performance, flexibility and data investment tradeoffs
  - Exascale will be much more challenging
- We recognize good sustained performance when we experience it – our solutions comes back faster
- The Top500 List is problematic in many ways and is not a good indicator of sustained performance for the Petascale era and beyond
- We can measure and document real sustained performance (e.g. SPP and/or other methods)
- The High Performance community must take on the challenge for realistic, explainable metrics
  - Do you want to help?
- NCSA is proposing and planning a series of workshops to refine ideas and develop alternatives – hopefully in time for SC 13

Acknowledgements

This work is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (award number OCI 07-25070) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign, its National Center for Supercomputing Applications, Cray, and the Great Lakes Consortium for Petascale Computation.

The work described is achievable through the efforts of the Blue Waters Project.

Individual Help From

- Thom Dunning, Marc Snir, Wen-mei Hwu, Bill Gropp
- Cristina Beldica, Brett Bode, Michelle Butler, Greg Bauer, Mike Showerman, Scott Lathrop, Torsten Hoefler, Irene Qualters, Sanjay Kale
- The Blue Waters Project Team and our partners
- NSF/OCI
- Cray, Inc, AMD, NVIDIA, Xyratex, Adaptive, Allinea

Background for TOP500 Issues

TOP500 ISSUES IN REVERSE ORDER OF IMPORTANCE

LINPACK is a single test that solves Ax=b with dense linear equations using Gaussian elimination with partial pivoting. matrix A, that is size M x M, LINPACK requires 2/3 M^2 + $2M^2$ operations. $O(N^2)$ memory and $O(N^3)$ Floating Point operations

10 - The Linpack benchmark serves only one or two of the four purposes of a good benchmark.

- Surveying benchmark literature benchmark uses can be grouped in four purposes.
  - Evaluation and/or selection of a system from among its competitors.
  - Validating that the selected system works as expected once it is built and/or arrives at a site.
  - Assuring the system performance stays as expected throughout the systems lifetime (e.g. after upgrades, changes,….)
  - Helping guide future system designs.
- Linpack **as commonly used** for Top500 listings serve only 1 or 2 of these at all.
  - Yes – many procurements use Linpack in some manner – but good ones use a lot more valid metrics.
  - Maybe – Top500 Linpack can only validate a system once due to run times of high performance run – and other tests can do better and faster.
  - No – Linpack, as implemented for Top500, is too intrusive for a system health metric
  - Not now - At one point it may have been useful – but Linpack no longer does address key architectural challenges.

9 - The TOP500 list disenfranchises many important application areas.

Science Area	Struct Grids	Unstruct Grids	Dense Matrix	Sparse Matrix	N-Body	Monte Carlo	FFT	PIC	Significant I/O
Climate and Weather	X	X		X		X			X
Plasmas/Magnetosphere	X				X		X		X
Stellar Atmospheres and Supernovae	X			X	X	X		X	X
Cosmology	X			X	X				
Combustion/Turbulence	X						X		
General Relativity	X			X					
Molecular Dynamics			X		X		X		
Quantum Chemistry			X	X	X	X			X
Material Science			X	X	X	X			
Earthquakes/Seismology	X	X			X				X
Quantum Chromo Dynamics	X		X	X	X		X		
Social Networks									
Evolution									
Engineering/System of Systems						X			
Computer Science		X	X	X			X		X

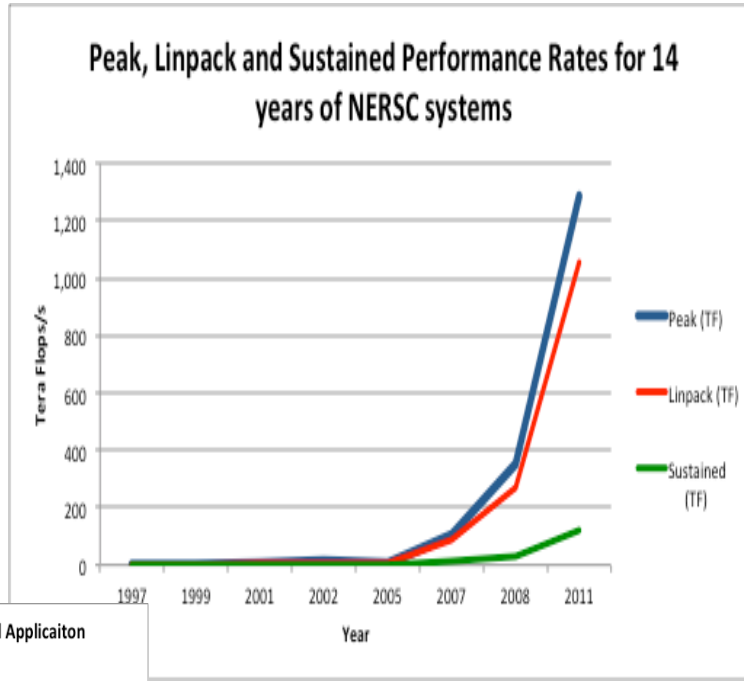
Red means Linpack does represent this method/ feature

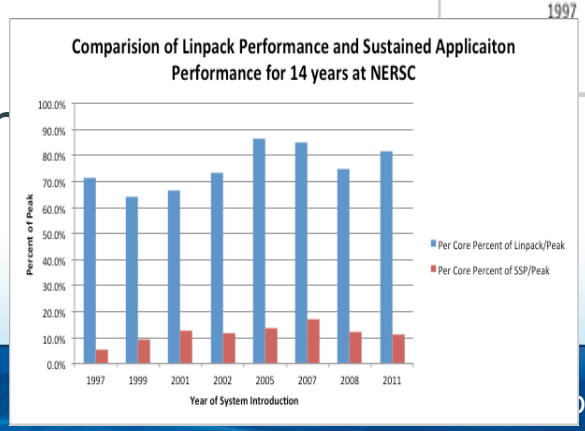
9 - The TOP500 list disenfranchises many important application areas. (cont)

- Over the past 10-15 years, many other methods have became critically important to time to solution as more sophisticated algorithms are adapted by science teams
  - Sparse Methods
  - Adaptive Methods
  - Etc.
- Hence: Few workloads are as dominated only by dense linear algebra as they were in the past

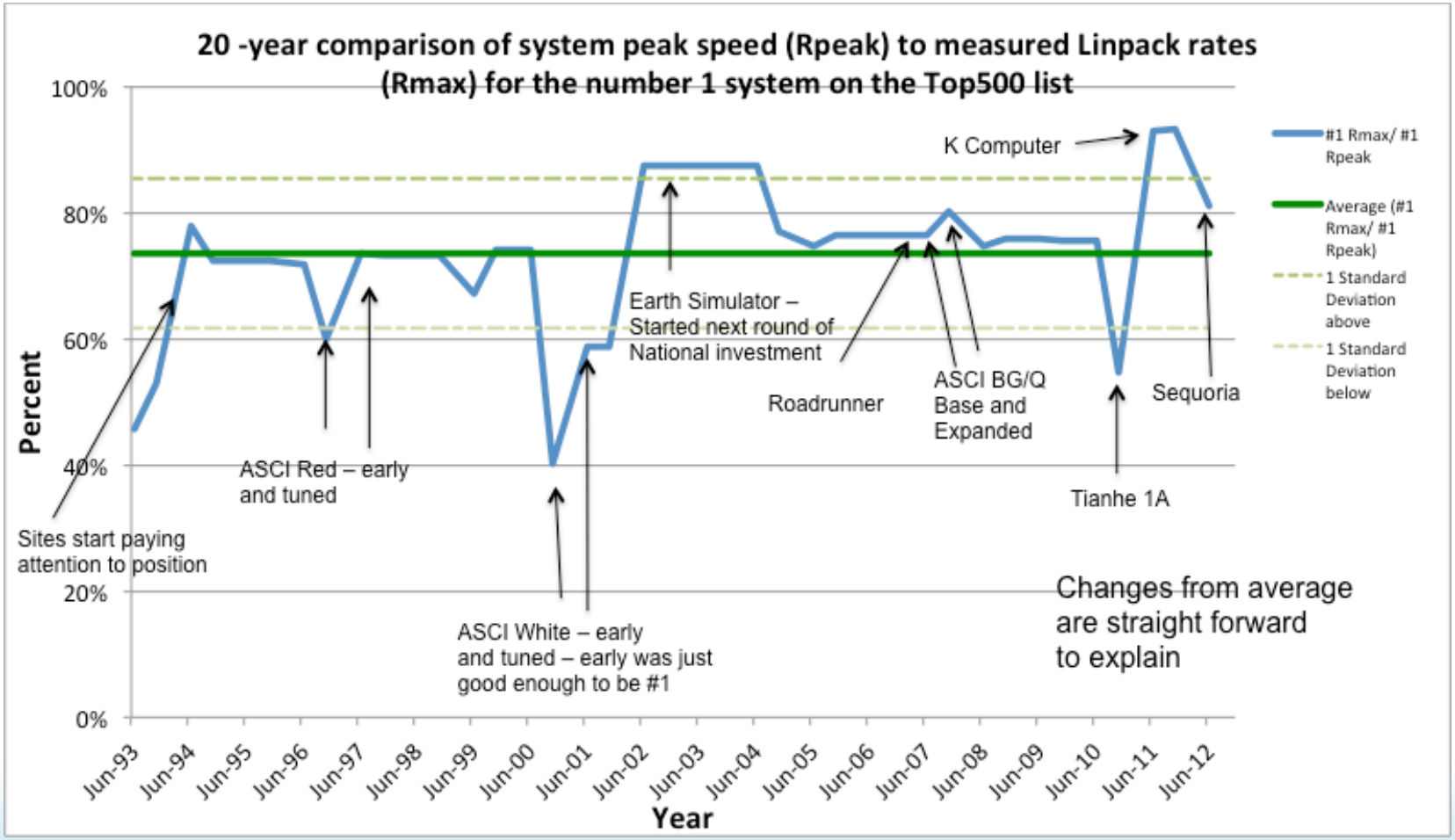
8 - There is no relationship between the TOP500 ranking and real work potential, user productivity, system usability for real application workloads.

- Top500 submission values vs measured System Sustained Performance do not correlate well
  - 13 years of systems at NERSC show this trend
  - Similar information other systems
    - DOD TI, BW, etc.





7 – The Top500 Linpack performance test is dominated by single-core, dense linear algebra peak performance.



6- The TOP500 metric has not kept up with changing algorithmic methods.

- The algorithmic methods in many applications evolve to compensate for architectural imbalances.
- Long-lived benchmarks should not be a goal – except possibly as regression tests to make sure improvements they generate stay within the design scope.
  - Rather Insight **Vitality** is the goal benchmarking
- Algorithm improvement/change is at least as important as hardware for real performance potential improvements.
- The effectiveness of a metric predicting delivered performance is founded on its accurate mapping to the target workload.
- A static benchmark(s) (even benchmark suites) eventually fail to provide an accurate means for assessing systems for real performance potential.
- Applications are entering the period of strong scaling that will drive new methods
  - Clock Speed slowdown
  - Memory capacity and data movement
  - Bandwidth is the limiting for performance at scale

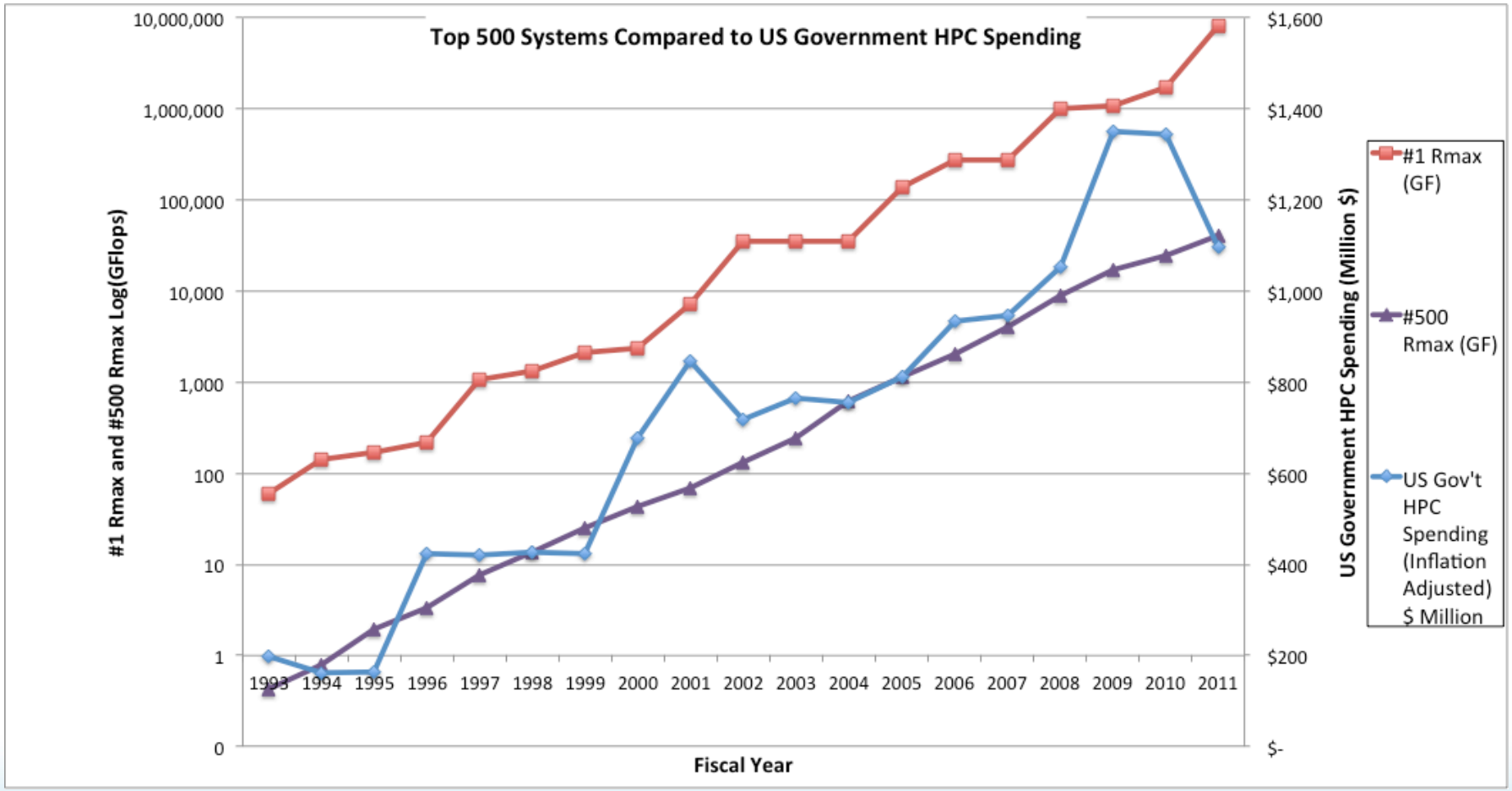
6- The TOP500 metric has not kept up with changing algorithmic methods. (cont)

- Over time, fixed benchmarks become less discriminating in predicting application workload performance.
- Once a simple benchmark gains traction in the community, system designers customize to do well on that benchmark.
  - The Livermore Loops, SPEC, LINPACK, NAS Parallel Benchmarks (NPB), etc. all had this issue.
    - E.g. Simon and Strohmaier showed, through statistical correlation analysis, that within two Moore's Law generations of technology and despite the expansion of problem sizes, only three of the eight NPBs remained statistically significant distinguishers of system performance presumably due to system designers making systems that responded more favorably to the widely used benchmark tests with hardware and software improvements.
  - It is clear LINPACK tracks peak performance in the large majority of cases.
- Must have constant introduction/validation of the "primary" tests that drive features for the future and check correct implementations today, and a constant "retirement" of the benchmarks that are no longer strong discriminators.
- But, there needs to be consistency of methodology and overlapping of benchmark generations so comparison across generations of systems is possible.
- Consequently, a metric must continue to evolve to stay current with workloads and future trends by changing both the application mix and the problem sets.
  - It is possible to compare the different measures as well so long running trends can be tracked.

5 - The Linpack TOP500 measure takes too long to run and does not represent strong scaling.

- In order to keep scaling performance high, as much work per processor as possible has to be loaded into the system's memory.
- The amount of memory used grows at $O(N^2)$; the run time to do the work grows at $O(N^3)$ and the run time for higher performance grows
- If all goes right good ranking takes long runs
  - On NERSC Cray XT-4 with ~39,000 cores and ~80 TB of aggregate memory, a single run of Linpack took 17-20 hours on the entire system
  - Blue Waters estimates, with 1.5 PB of memory and 380,000 cores, could be several days
- Large scale sites may have to run just Linpack 5-10 times to get a good run
  - Rumors of months of just Linpack time devoted to a Top 500 listing
- The early science BW system might have lost 4-8% of its useful time if we had wanted to list it on the June 2012 list. (It would have been high on the list).
  - We choose not to waste that science time.
- Note that in June 2012 Dongarra proposed the list accept results from partial runs to address this problem

4 - The TOP500 is dominated by who has the most money to spend–not what system is the best.



3 - The TOP500 provides little historical value

- What history does the Top500 really show us?
  - What sites get the most funding?
  - Moore's Law improvements?
  - Regional/organization investments?
- Other measures for history are as good or better and less intrusive
  - Combining already available information of government investment in HPC systems and IC transistor density probably equally as good.
  - Example - Gordon Bell Award equivalent "history" – but still has related issues
- Top500 entries are sometimes skewed in time by 6 or more months early than real impacts
  - Systems listed long before they are in use
  - Some systems never get into use for applications
- What important things is the Top500 not providing that is important for historical assessments?
  - Indication of real application performance
  - Indication of algorithmic improvements for application
  - Ease of use

2 - The TOP500 encourages organizations to make poor choices.

- Linpack is memory constrained scaling "which is attractive to vendors because such speed ups are high" (Culler and Singh 1999).
  - The Top500 is now a simplified marketing tool
  - Marketing use is encouraged by some supporters of the metric
- Notable examples of systems being ill-configured in order to increase the ranking on the Top500 list.
  - Leaves systems that are imbalanced and less efficient for their application workloads.
    - Repeatedly, storage capacity and bandwidth and memory capacity are sacrificed in order to increase the number of peak (and therefore Linpack) flops in a system.
  - In these cases–which include some very large systems–it is often the case that the types of applications that can be run well on the resulting system are limited.

2 - The TOP500 encourages organizations to make poor choices. (cont)

- Also, the goal of listing a system can be so important that organizations may actually defer real use of the system.
  - Delays in Service and/or inability to go into Service
- Pressure of the list drives organizations to maximize peak Flops
  - Example –
  - For the same dollar investment, Blue Waters could have 3 to 4x peak/ Linpack performance if we had minimized memory and I/O-storage.
    - Would have ensures a very high ranking for quite multiple list cycles.
    - Doing such a design is counter to the desires and best interests of the application teams
  - Instead Blue Waters is one of the most balanced systems in all time that has the largest aggregate memory and most intense storage capability as well as exceptional computational usability

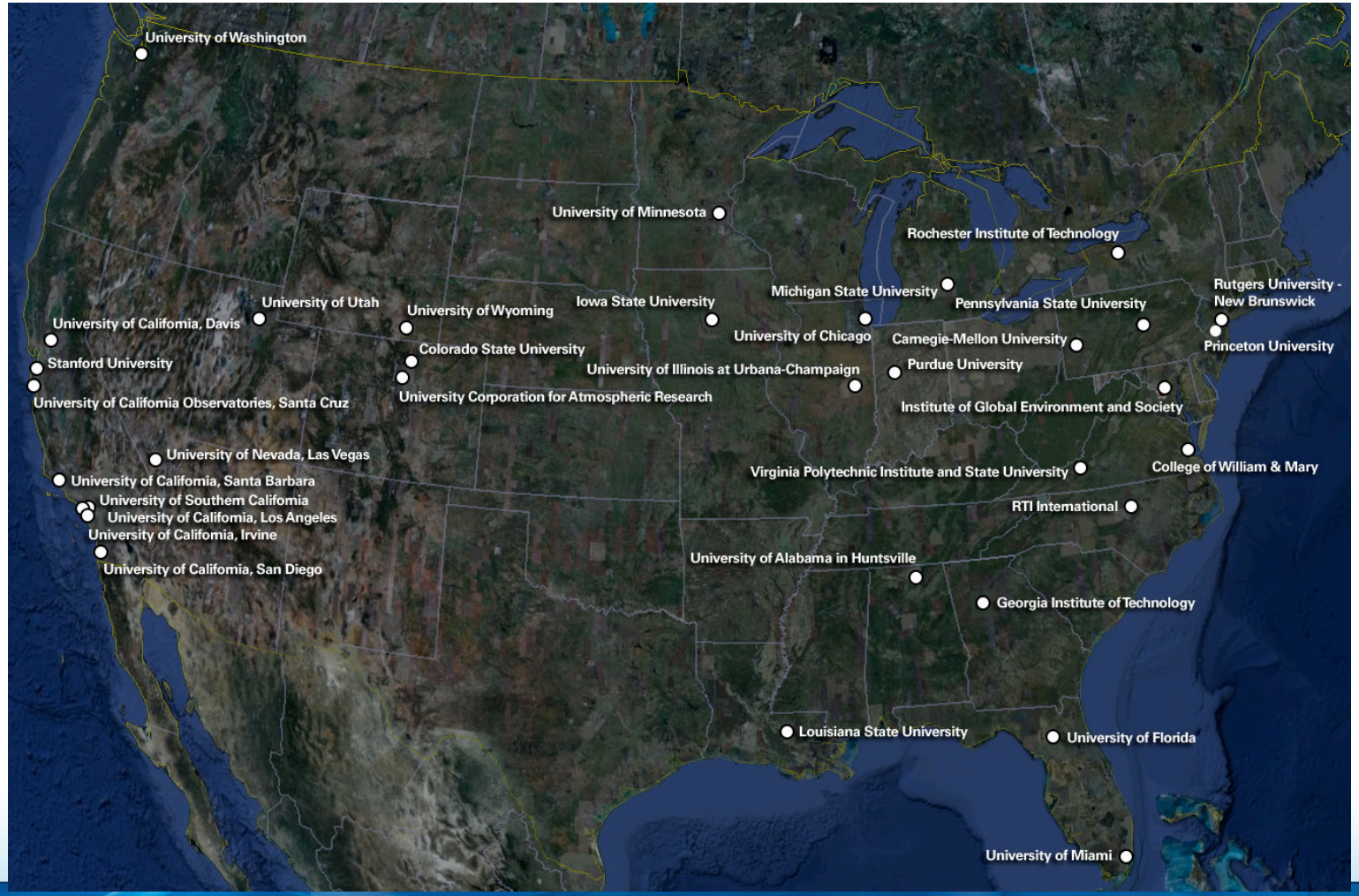
1 - The TOP500 gives no indication of the cost or value of a system.

- The Top500 list is not able to compare the value of different system architectures or implementations since a dominant factor in performance and list ranking is how much money was spent.
  - Example – Earth Simulator investment may have been 5-10x that of the #2 system listed at the time.
    - Moral equivalent of claiming is it relevant to compare Donald Trump's house to the one most of us need?
- Red Herring excuses for the list not providing cost information
  - E.g. cost information is not exactly comparable due to discounts, circumstances, etc. Since it is to some degree inexact why bother to provide cost at all? –
- **Key fact** – Cost is needed to establish Value
  - Value = Potential/Cost - Potential alone is not really usable
- Gathering reasonably accurate cost information is possible and feasible
  - Many press releases about new systems state contract cost totals
  - Many procurement documents provide the estimated available funds
  - Not to exceed prices can be derived from published price lists
  - Specialist organizations like IDC are willing to provide accurate "street prices"

1 - The TOP500 gives no indication of the cost or value of a system. (cont)

- Existence proofs that gathering useful cost data exist
  - The original NAS parallel benchmark rules were specific that a system would not be listed unless a cost estimate was provided for that system.
  - The NPBs were highly successful in capturing meaningful performance information for many years.
  - Energy usage listings succeeding but energy use while running one test is only a one dimensional cost of ownership data

Blue Waters PRAC PI Institutions – September2012



NSF PRAC Major Science Teams

PI	Award Date	Project Title
Sugar	04/15/2009	Lattice QCD on Blue Waters
Bartlett	04/15/2009	Super instruction architecture for petascale computing
Nagamine	04/15/2009	Peta-Cosmology: galaxy formation and virtual astronomy
Bissett	05/01/2009	Simulation of contagion on very large social networks with Blue Waters
O'Shea	05/01/2009	Formation of the First Galaxies: Predictions for the Next Generation of Observatories
Schulten	05/15/2009	The computational microscope
Stan	09/01/2009	Testing hypotheses about climate prediction at unprecedented resolutions on the NSF Blue Waters system
Campanelli	09/15/2009	Computational relativity and gravitation at petascale: Simulating and visualizing astro-physically realistic compact binaries
Yeung	09/15/2009	Petascale computations for complex turbulent flows
Schnetter	09/15/2009	Enabling science at the petascale: From binary systems and stellar core collapse To gamma-ray bursts
Woodward	10/01/2009	Petascale simulation of turbulent stellar hydrodynamics
Tagkopoulos	10/01/2009	Petascale simulations of Complex Biological Behavior in Fluctuating Environments
Wilhelmson	10/01/2009	Understanding tornadoes and their parent supercells through ultra-high resolution simulation/analysis
Wang	10/01/2009	Enabling large-scale, high-resolution, and real-time earthquake simulations on petascale parallel computers
Jordan	10/01/2009	Petascale research in earthquake system science on Blue Waters
Zhang	10/01/2009	Breakthrough peta-scale quantum Monte Carlo calculations
Haule	10/01/2009	Electronic properties of strongly correlated systems using petascale computing
Lamm	10/01/2009	Computational chemistry at the petascale

NSF PRAC Major Science Teams (cont)

PI	Award Date	Project Title
Karimabadi	11/01/2010	Enabling Breakthrough Kinetic Simulations of the Magnetosphere via Petascale Computing
Mori	01/15/2011	Petascale plasma physics simulations using PIC codes
Voth	02/01/2011	Petascale multiscale simulations of biomolecular systems
Woosley	02/01/2011	Type Ia supernovae
Cheatham	02/01/2011	Hierarchical molecular dynamics sampling for assessing pathways and free energies of RNA catalysis, ligand binding, and conformational change
Wuebbles	04/15/2011	Using petascale computing capabilities to address climate change uncertainties
Gropp	06/01/2011	System software for scalable applications
Klimeck	09/15/2011	Accelerating nano-scale transistor innovation
Pande	09/15/2011	Simulating vesicle fusion on Blue Waters
Elghobashi	05/18/2012	Direct Numerical Simulation of Fully Resolved Vaporizing Droplets in a Turbulent Flow
Quinn	05/18/2012	Evolutions of the Small Galaxy Populations From High Redshift to the Present
Wood/Reed	06/12/2012	Collaborative Research: Petascale Design and Management of Satellite Assets to Advance Space Based Earth Science
Pogorelov	0613/2012	Modeling Heliophysics and Astrophysics Phenomena with a Multi-Scale Fluid Kinetic Simulation Suite
Bernholc	07/15/2012	Petascale quantum simulations of nano systems and biomolucles
Stein	08/01/2012	Ab Initio Models of Solar Activity

Science Area	Number of Teams	Codes	Struct Grids	Unstruct Grids	Dense Matrix	Sparse Matrix	N-Body	Monte Carlo	FFT	PIC	Significant I/O
Climate and Weather	3	CESM, GCRM, CM1/WRF, HOMME	X	X		X		X			X
Plasmas/Magnetosphere	2	H3D(M),VPIC, OSIRIS, Magtail/ UPIC	X				X		X		X
Stellar Atmospheres and Supernovae	5	PPM, MAESTRO, CASTRO, SEDONA, ChaNGa, MS-FLUKSS	X			X	X	X		X	X
Cosmology	2	Enzo, pGADGET	X			X	X				
Combustion/Turbulence	2	PSDNS, DISTUF	X						X		
General Relativity	2	Cactus, Harm3D, LazEV	X			X					
Molecular Dynamics	4	AMBER, Gromacs, NAMD, LAMMPS			X		X		X		
Quantum Chemistry	2	SIAL, GAMESS, NWChem			X	X	X	X			X
Material Science	3	NEMOS, OMEN, GW, QMCPACK			X	X	X	X			
Earthquakes/Seismology	2	AWP-ODC, HERCULES, PLSQR, SPECFEM3D	X	X			X				X
Quantum Chromo Dynamics	1	Chroma, MILC, USQCD	X		X	X	X		X		
Social Networks	1	EPISIMDEMICS									
Evolution	1	Eve									
Engineering/System of Systems	1	GRIPS,Revisit						X			
Computer Science	1			X	X	X			X		X