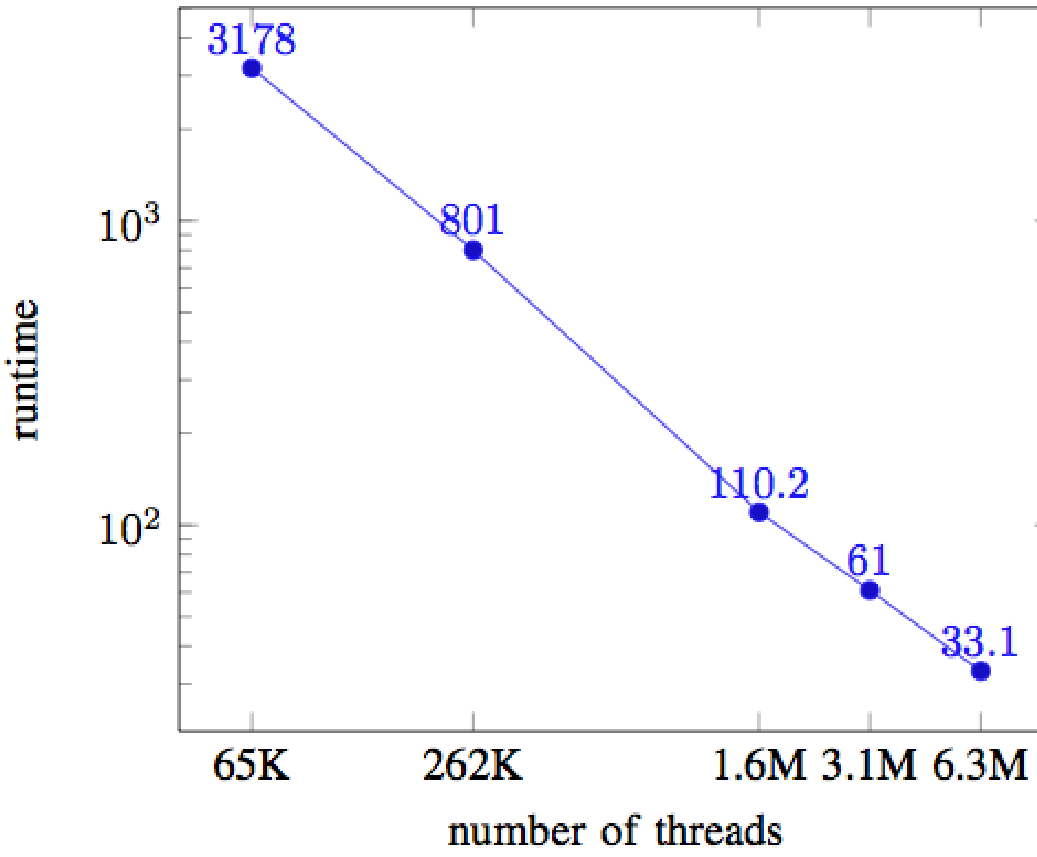
Breaking the Power Wall in Exascale Computing

Dr. Costas Bekas

IBM Research - Zurich.

Success in Petascale computing: BG/Q Results

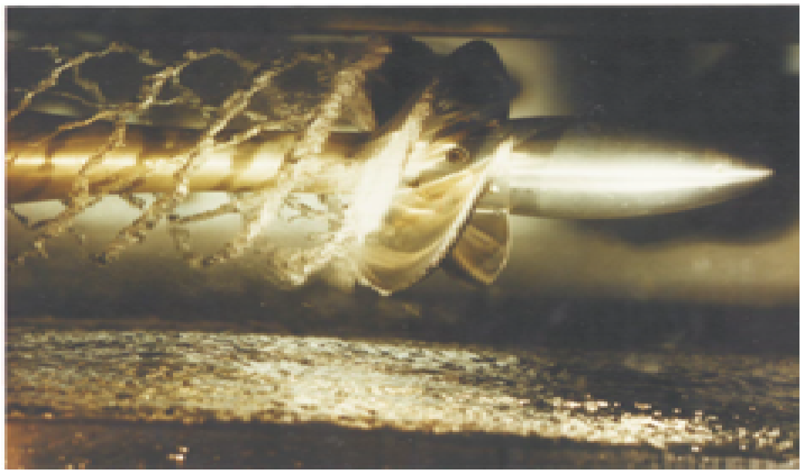
Implementing Exact-Exchange in CPMD
>99% Parallel Efficiency to over 6.2M threads
Studying Li-Air Batteries, 1736 atoms, 70Ry cuttof



V. Weber, T. Laino, C. Bekas, A. Curioni, A. Bertsch, S. Futral IPDPS 13

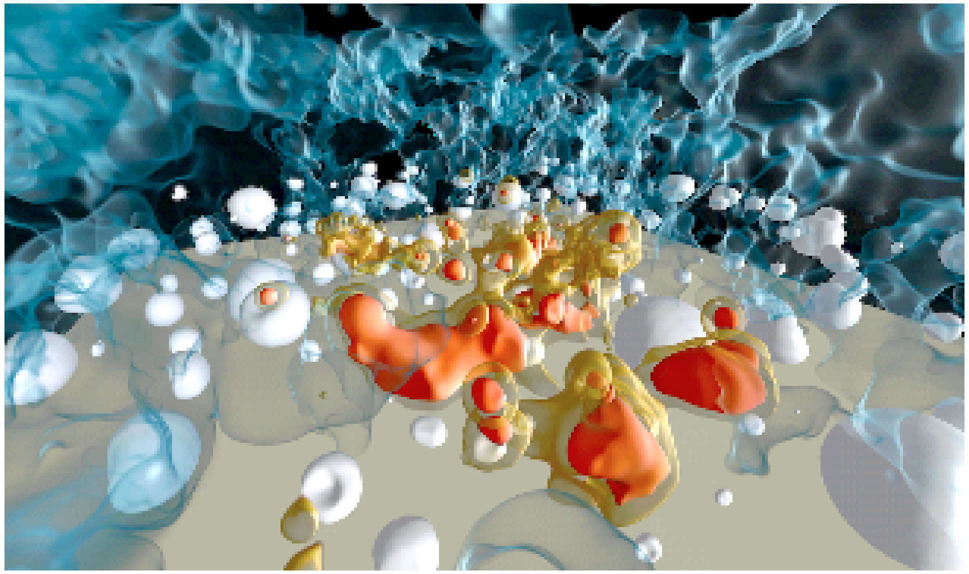
Success in Petascale computing: BG/Q Results

Cloud cavitation collapse



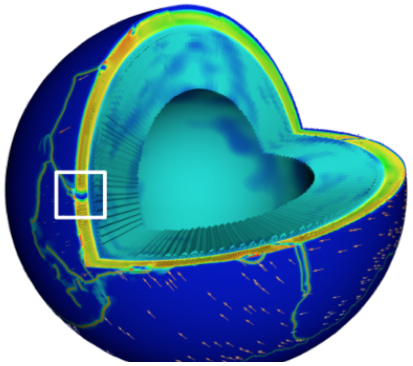
ACM Gordon Bell Prize 2013
14.4 PFLOP/S @73% of peak perf.

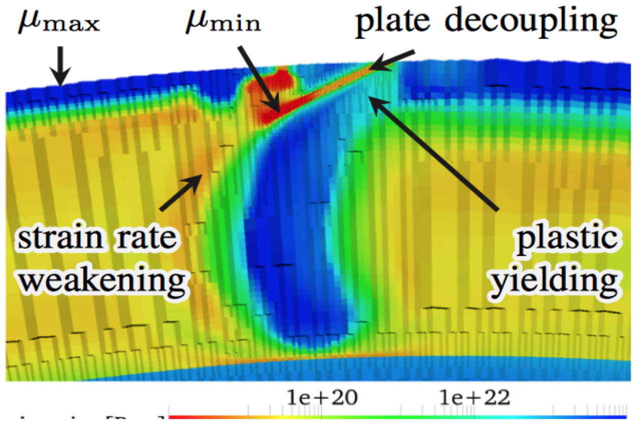
13 Trillion elements
6.4 M threads



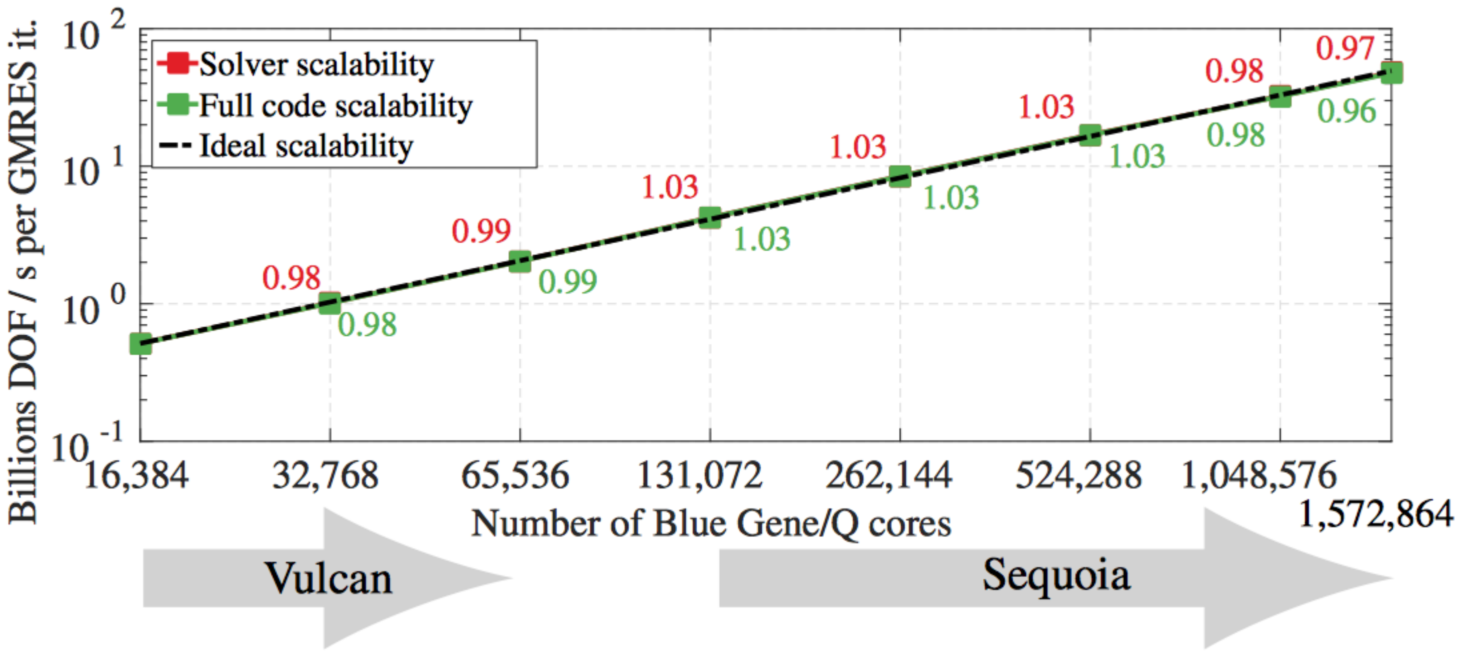
Success in Petascale computing: BG/Q Results

Mantle Simulations





ACM Gordon Bell finalist 2015
97% of sustained scalability for a fully implicit solver. 1.6M cores
3.2M MPI processes



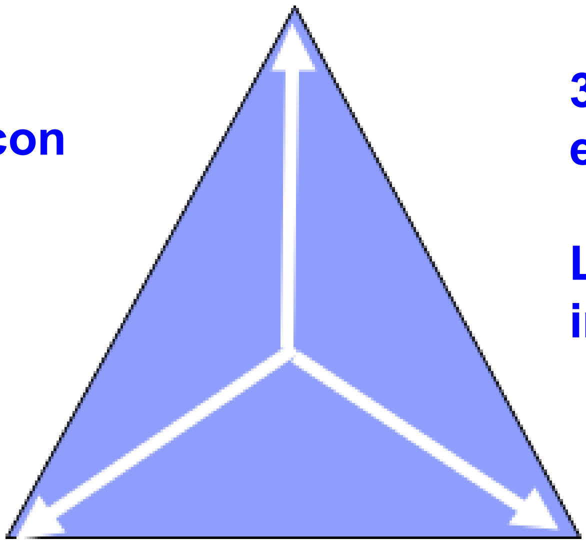
This talk is about Reaching Exascale and Beyond:

The Energy/Power Barrier and How Algorithmic Re-engineering Can Open the Way

Exascale Targets: Difficulties Along all Axes

Sustained Performance / $
50x improvement needed

~5x more area of silicon Expected. 50x more compute pipelines



3-4 technology generations expected

Linear dimensions: 3x-4x improvements expected

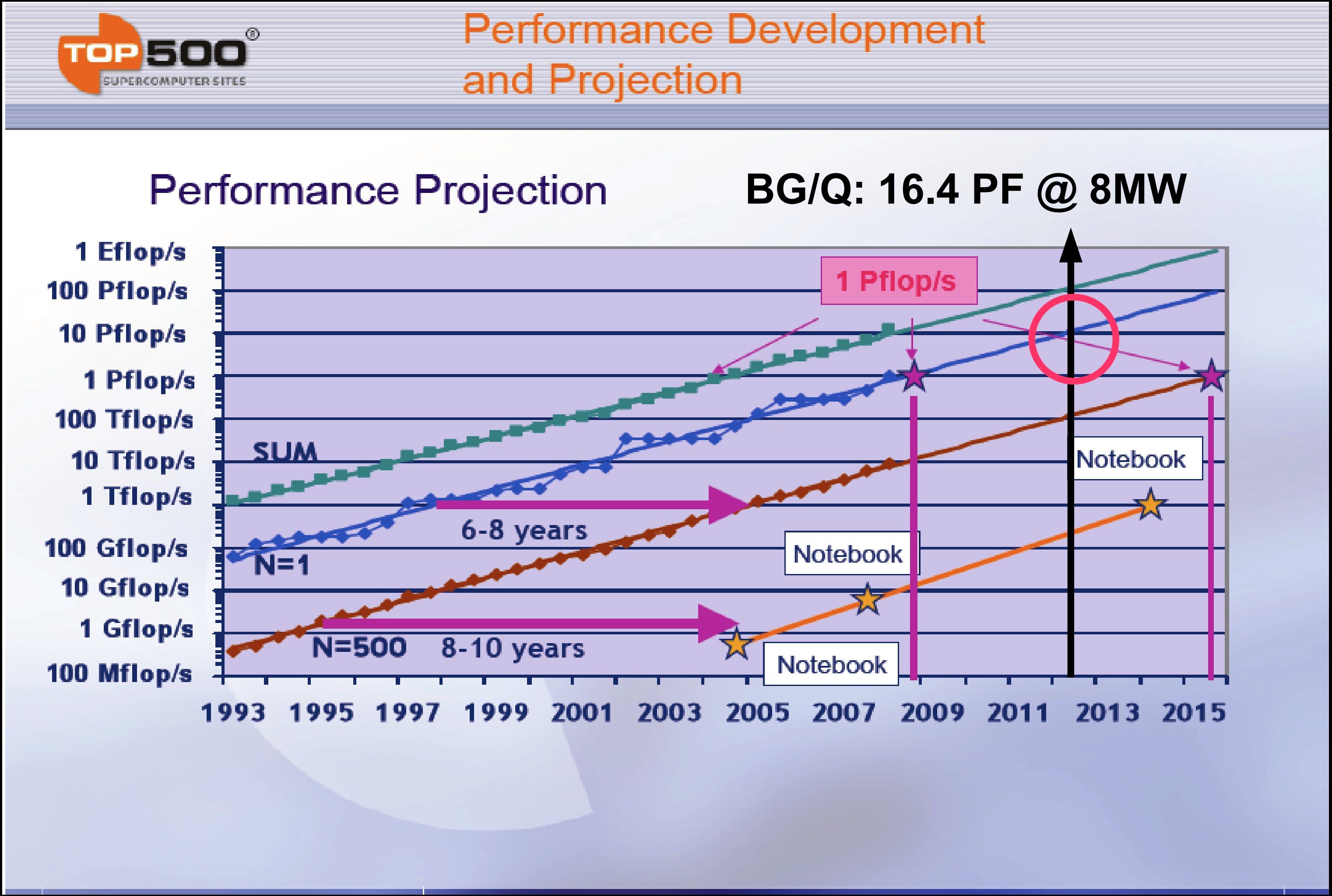
Ease of Use / Reliability
Broad scientific impact
50x improvement needed

Sustained Performance/Watt
20x improvement needed

***Improvements relative to 2011/2012 BG/Q 20 PF/s systems**

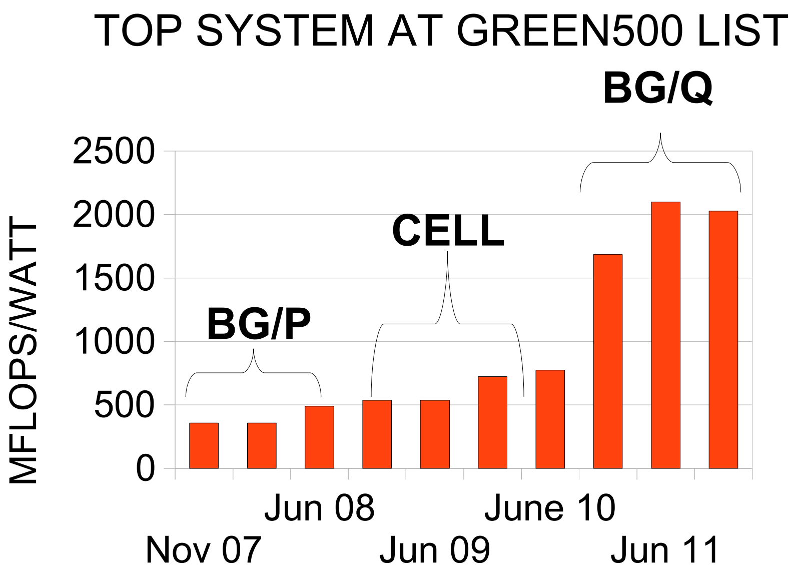
Measuring performance in HPC:

The traditional way ... MFLOP/SEC … has brought us this far!



Source: Top500.org

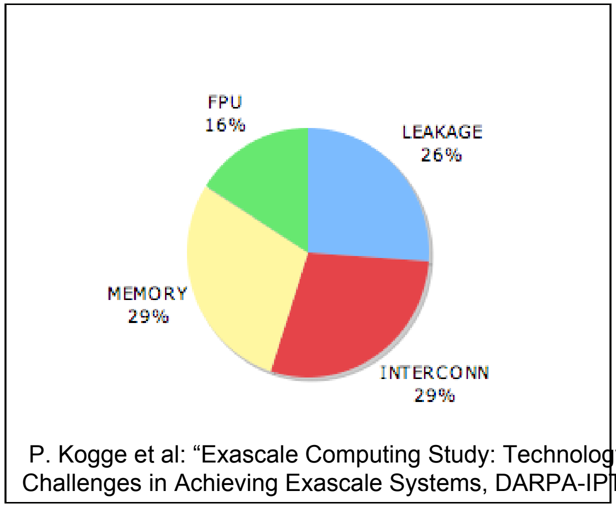
Achieving Exascale: The Energy/Power Wall



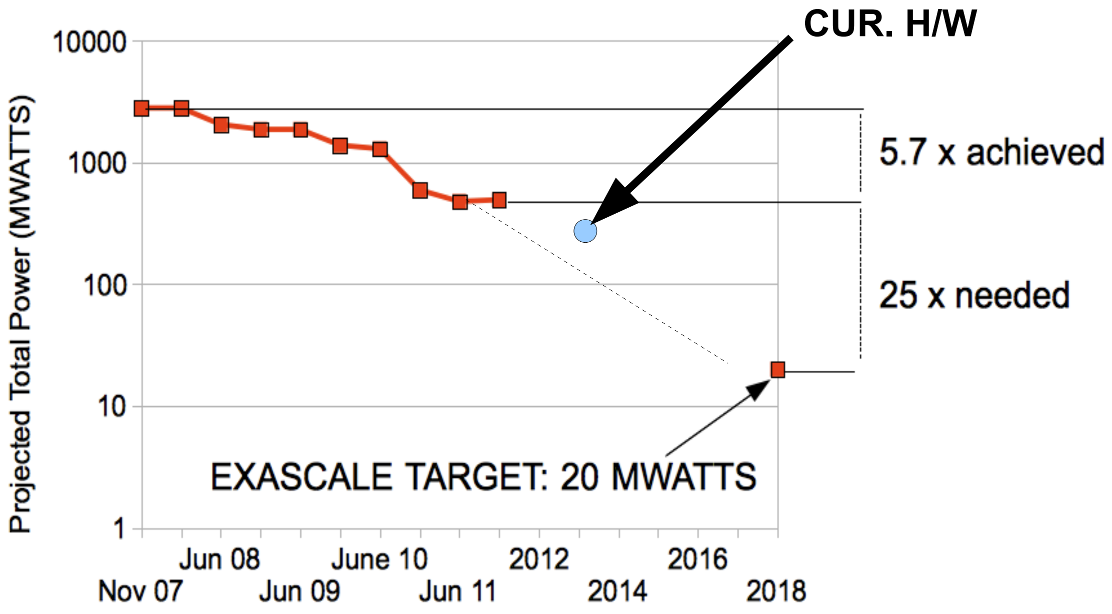
Green HPC
Focus moves from MFLOPS to MFLOPS/WATT

Given a power budget target maximize operations

www.green500.org: derived from www.top500.org



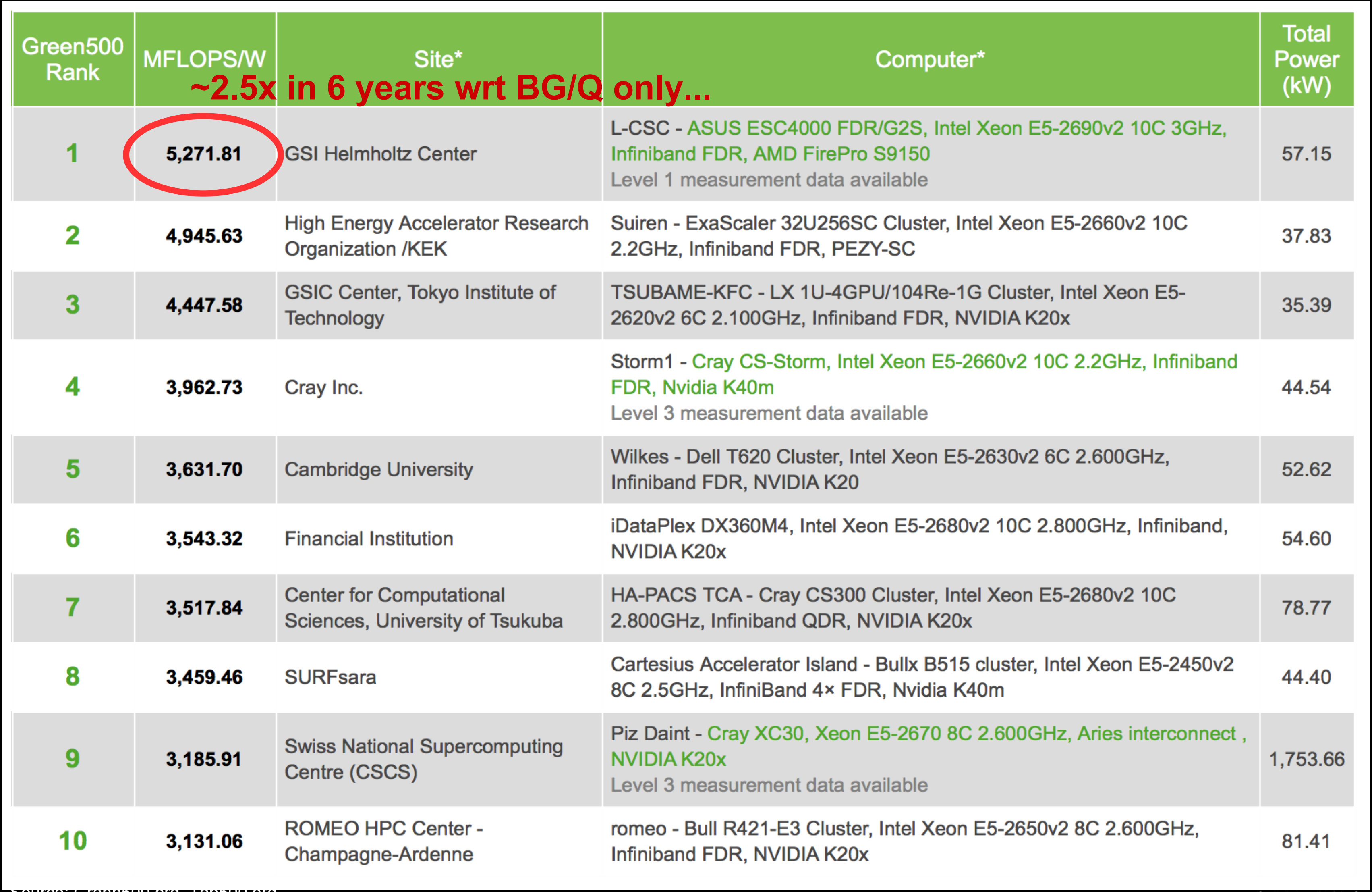
P. Kogge et al: "Exascale Computing Study: Technology Challenges in Achieving Exascale Systems, DARPA-IPTO, 2008



Exascale systems:
- FPU to cost a fraction of total energy (16%)
- Total data movements: ~60%

Measuring performance in HPC: A major step forward

The green way... MFLOPS/Watt: www.green500.org

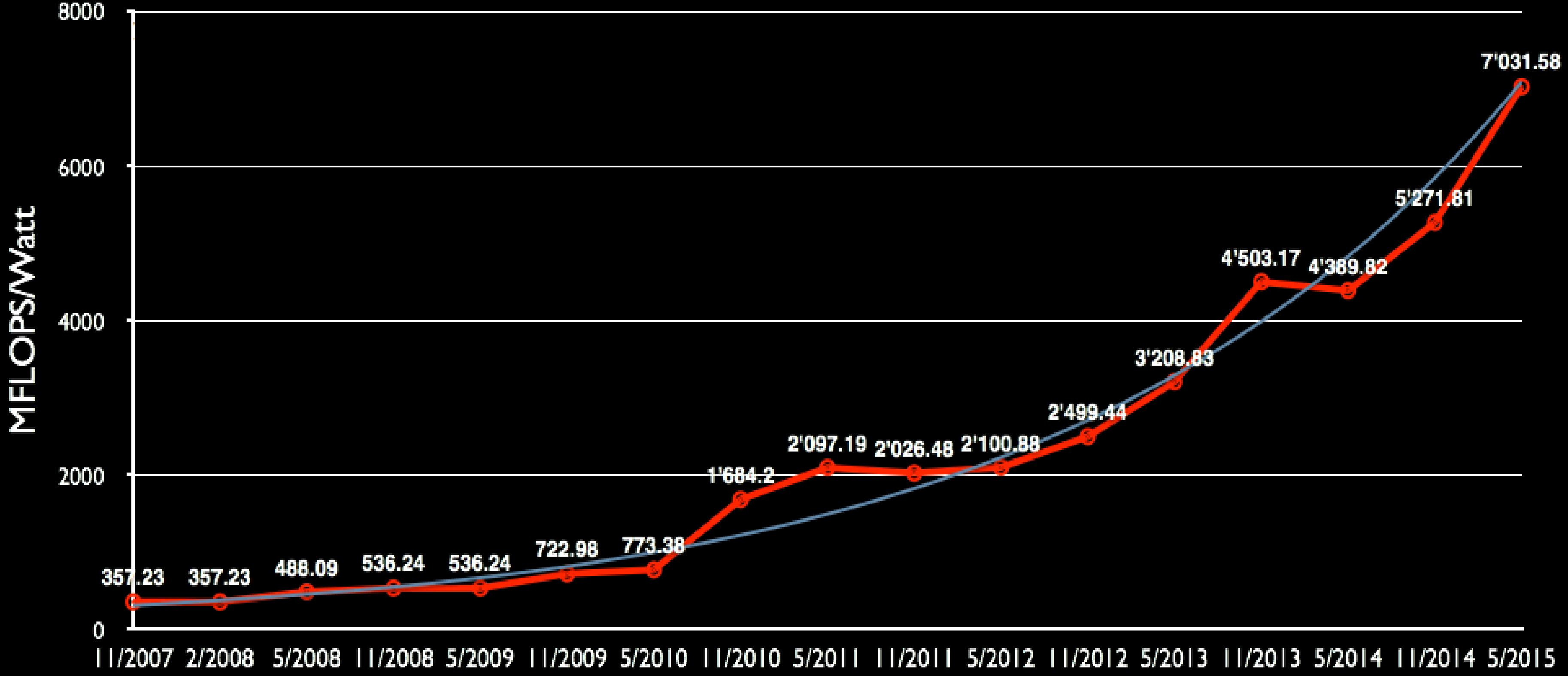


~2.5x in 6 years wrt BG/Q only...

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	**5,271.81**	GSI Helmholtz Center	L-CSC - ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150 Level 1 measurement data available	57.15
2	**4,945.63**	High Energy Accelerator Research Organization /KEK	Suiren - ExaScaler 32U256SC Cluster, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, PEZY-SC	37.83
3	**4,447.58**	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x	35.39
4	**3,962.73**	Cray Inc.	Storm1 - Cray CS-Storm, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, Nvidia K40m Level 3 measurement data available	44.54
5	**3,631.70**	Cambridge University	Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20	52.62
6	**3,543.32**	Financial Institution	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	54.60
7	**3,517.84**	Center for Computational Sciences, University of Tsukuba	HA-PACS TCA - Cray CS300 Cluster, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband QDR, NVIDIA K20x	78.77
8	**3,459.46**	SURFsara	Cartesius Accelerator Island - Bullx B515 cluster, Intel Xeon E5-2450v2 8C 2.5GHz, InfiniBand 4× FDR, Nvidia K40m	44.40
9	**3,185.91**	Swiss National Supercomputing Centre (CSCS)	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Level 3 measurement data available	1,753.66
10	**3,131.06**	ROMEO HPC Center - Champagne-Ardenne	romeo - Bull R421-E3 Cluster, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR, NVIDIA K20x	81.41

Source: Grenn500.org, Top500.org

We start to see an exponential behavior in the Green500. But is this really affecting the top line? 5 years ago: 2.1 GF/W, now 1.9 GF/W



Source: Grenn500.org, Top500.org

Measuring performance in HPC the Green way

- Main idea: run LINPACK on power optimized hardware...
- Hardware is power optimized for **LINPACK** specific tasks
  - **FLOP intensive calculations**
  - **Heavy memory hierarchy utilization**
  - **Heavy interconnect utilization**
- Thus: *if all goes well...*We can do more flops for each available watt

- ✓ But: Is this what Green computing is about?
- ✓ Real target: Total Energy Spent
- ✓ Can the FLOPS/WATT metric give a good indication?

FTTSE (Bekas-Curioni, EnaHPC, Hamburg, Sept, 2010)

- Energy aware performance metric

  FTTSE = f(tts) x Energy

  - **tts: time to solution**
  - **f(tts) a function of time to solution**

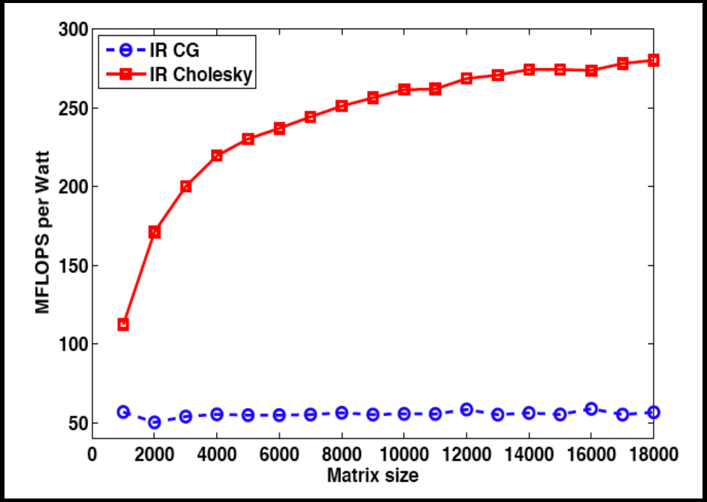
- FTTSE v.s. F/W
  - F/W still promotes power hungry algorithms:
    - Why: Flops and Watts are optimized separately
    - Thus: Once a satisfactory power budget is achieved then users tend to maximize sustained flops
    - High sustained flops comes from algorithms that make full use of the hardware

- FTTSE v.s. F/W
  - F/W is a “natural” green extension of the original F/S metric
    - Fix a certain benchmark (LINPACK: solution of dense linear systems) and then compare machines flops per watt wrt. this benchmark.
  - Moving to FFTSE demands for simultaneous minimization of power consumption and time to solution:
    - Architectures cannot any longer be measured against a single benchmark! LINPACK is not enough.
    - Instead: Collection of benchmarks (i.e. 7-13 Colella's Dwarfs)
    - Example: Optimize architecture for sparse computations, FFT's (heterogeneous chips?)

Examples of Algorithmic Rethinking

Solving a Dense Symmetric Positive Definite Linear System

Ax=b



Typically this is a "no-brainer"

Use Cholesky: BLAS3..thus optimal...

But is it?

SOLVING DENSE SPD LINEAR SYSTEMS

Cholesky Decomposition:

$$\text{If } A \text{ is SPD: } A=R^TR$$

R is upper triangular. Then solving Ax=b becomes

$$x=A^{-1}b = (R^TR)^{-1}b = R^{-T}R^{-1}b$$

Inverting (solving: back substitution) triangular matrices is cheap! $O(n^2)$

But the Cholesky decomposition costs $O(n^3)$

Observe: n=1M, already requires Exaflop like resources.

Can we do better? Can we accelerate?

DIVE IN THE PAST: ITERATIVE REFINEMENT

Consider the linear system: $Ax = b$ and assume we have an initial "guess" x_0

- **Compute the residual:** $r = b - Ax_0$
- **Solve for the residual:** $Ad = r$
- **Update the solution:** $x_1 = x_0 + d$

Repeat steps 1-3 if remainder is not small enough: $||r||_2$ □ tol

What if steps 1-3 could be done in infinite precision (*no rounding errors*):

1. $d = A^{-1}r = A^{-1}(b - Ax_0)$
2. $d = x - (A^{-1}A)x_0 = x - x_0$
3. $x_1 = x_0 + x - x_0 = x$

Thus, we would have a completely accurate result in 1 step!
But, round-off is inevitable. So, why does IR work?

Computing r and d accurately "enough" is adequate to bring improvement to x_1

MIXED PRECISION ITERATIVE REFINEMT
WHAT IF WE HAD FAST/LOW POWER/ HARDWARE AVAILABLE?

Consider two modes of machine precision:

- ✓ **LOW PRECISION: LP**
- ✓ **HIGH PRECISION: HP**

```
1. Compute the Cholesky factorization: $A=R^TR$.   Cost: $O(1/3n^3)$     ACCELARATION
2. Compute initial solution: $R^T(R x_0) = b$.     Cost: $O(n^2)$
3. Compute initial residual: $r_0 = b - Ax_0$. Cost: $O(n^2)$
4. $k = 0$
5. REPEAT
    1. Solve for residual:   $R^T(R d_k) = r_k$        Cost: $O(n^2)$
    2. Update solution:      $x_{k+1} = x_k + d_k$     Cost: $O(n)$
    3. Compute residual:     $r_{k+1} = b - Ax_{k+1}$  Cost: $O(n^2)$
    4. $k = k + 1$
•  UNTIL $||r_{k+1}||$ □ tol
```

Key properties:

1. Overall cost $O(1/3n^3)$. But performed in LOW PRECISION. Cost in HP is $O(n^2)$
2. We can take great advantage of fast single precision hardware!

Mixed Precision IR: Does it converge?

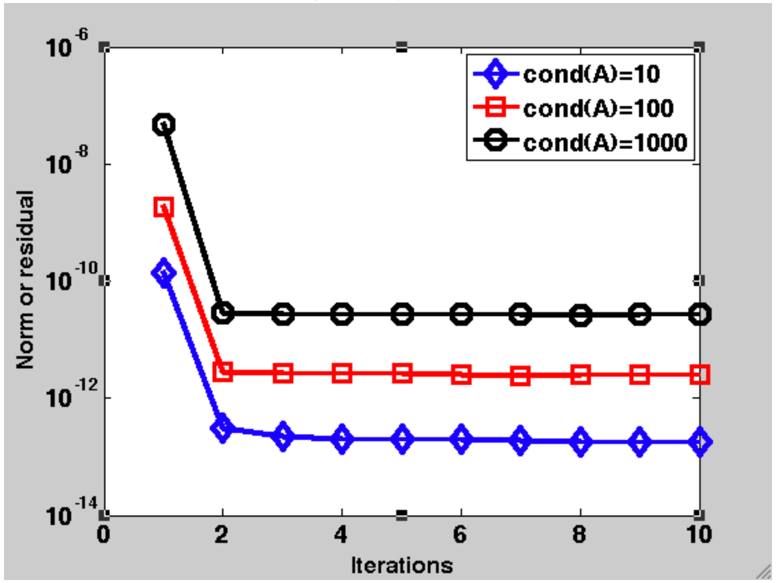
Theory

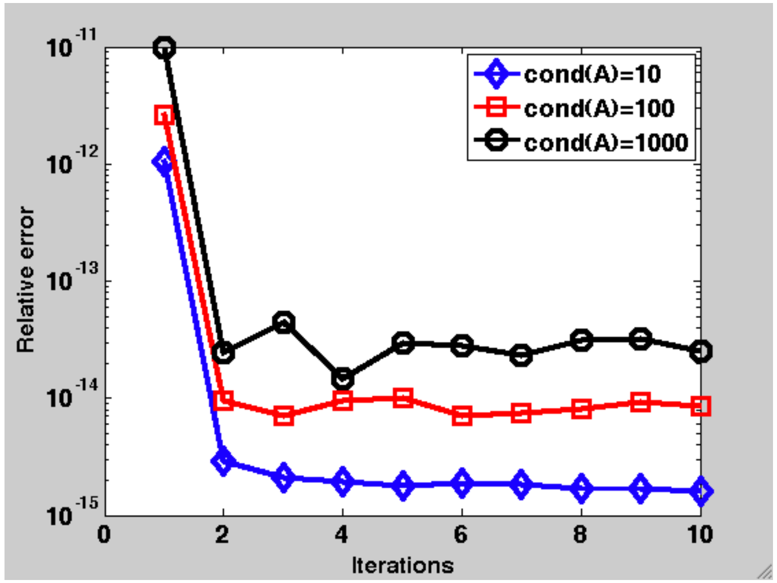
Mixed Precision IR converges so long as the solver we use for a system Ay = c satisfies for the computed solution y':

$$(A + €)\, y' = c, \qquad \|A^{-1} E\|_1 < 1$$

Indeed we can approximate a result in nearly full High Precision:

RESIDUAL **REL. ERROR**





Mixed Precision IR: Fast Low Precision

Consider two modes of machine precision:

- ✓ **LOW PRECISION: LP**
- ✓ **HIGH PRECISION: HP**

1. **Compute the Cholesky factorization: $A=R^TR$. Cost: $O(1/3n^3)$**

.

.

.

We can take great advantage of very fast low precision hardware!

- ✓ **Dominant cost $O(1/3n^3)$ is all in low precision**
- ✓ **Thus we can accelerate computations…**
- ✓ **We benefit from reduced memory traffic (compare 4 bytes of IEEE single to 8 bytes for IEEE) double**

So…what is the catch?

- ✓ **Cost remains cubic! Intractable to solve large systems (very large n). How about parallel?**
- ✓ **Cholesky is well known to present difficulties in parallel scaling**

WHY DOES IR: WORK?

Theory

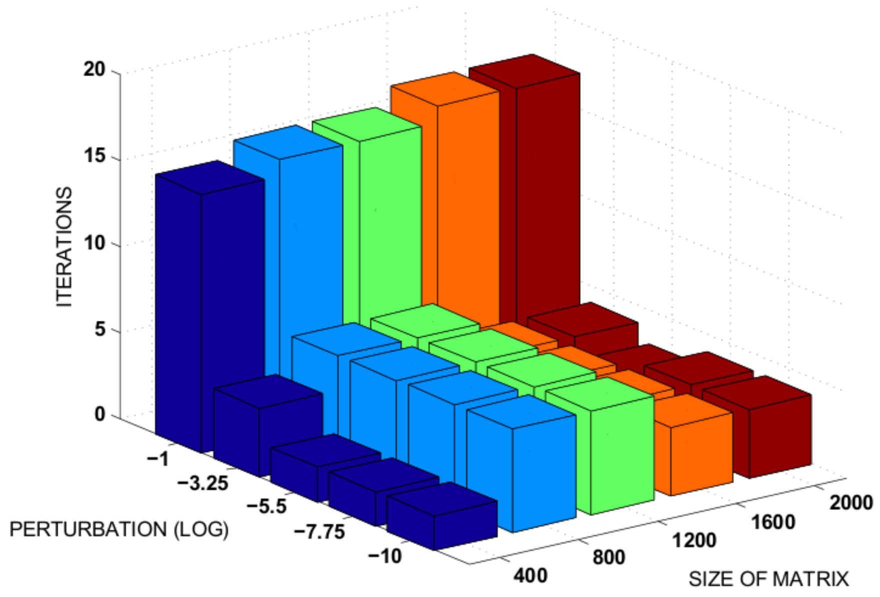
Mixed Precision IR converges so long as the solver we use for a system Ay = c satisfies for the computed solution y':

$$(A + €)\, y' = c, \qquad \|A^{-1} €\|_1 < 1$$

Can we relax solver accuracy?

Can we use "dirty/noisy" solvers?

Answer: YES



Using iterative solvers instead of Cholesky

- ✓ **The cubic complexity of standard Iterative Refinement stems from the Cholesky decomposition**
- ✓ **We saw that we could utilize a significantly less accurate solver**

We propose:

- **Substitute the dense solver (Cholesky based) with an iterative one**
- **For SPD linear systems this will be the Conjugate Gradient solver**
- ✓ **Perform only a small (constant) number of CG steps, k<<n**
- **Total cost reduces from $O(n^3)$! $O(kn^2)$, for a small k**

CG Based Iterative Refinement

- ✓ **LOW PRECISION: LP**
- ✓ **HIGH PRECISION: HP**
- ✓ **Let CG(A,y,k) be a procedure implementing k steps of CG in single precision**

```
• Compute initial solution: $x_0$=CG(A,b,k)                  Cost: $O(kn^2)$
• Compute initial residual: $r_0 = b - Ax_0$                 Cost: $O(n^2)$
• $k = 0$
• REPEAT
    • Solve for residual:    $d_k$= CG(A,$r_k$,k)            Cost: $O(kn^2)$
    • Update solution:       $x_{k+1} = x_k + d_k$           Cost: $O(n)$
    • Compute residual:      $r_{k+1} = b - Ax_{k+1}$        Cost: $O(n^2)$
    • $k = k + 1$
•UNTIL $||r_{k+1}||$ □ tol
```

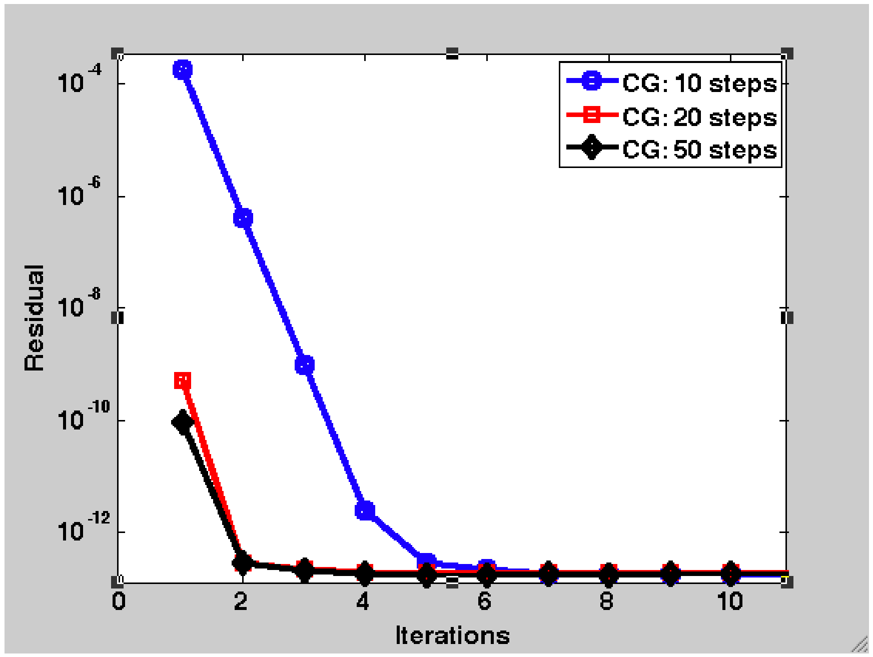
Key properties:

Dominant cost $O(kn^2)$. Performed in LOW PRECISION. Cost in HP is $O(n^2)$
We can take great advantage of fast single precision hardware!

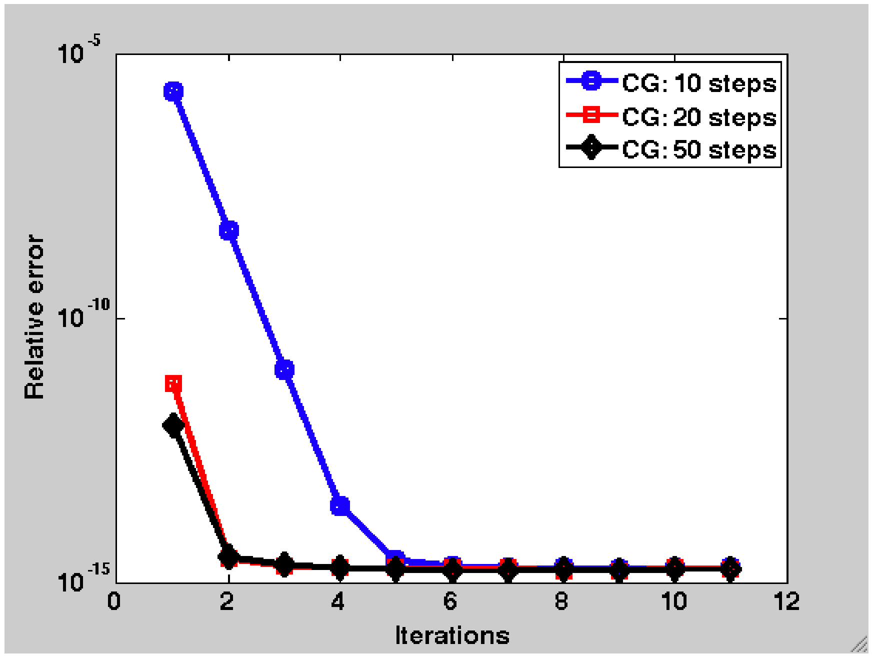
CG IR: Does it work?

Dense matrix A (n=1000)

Residual



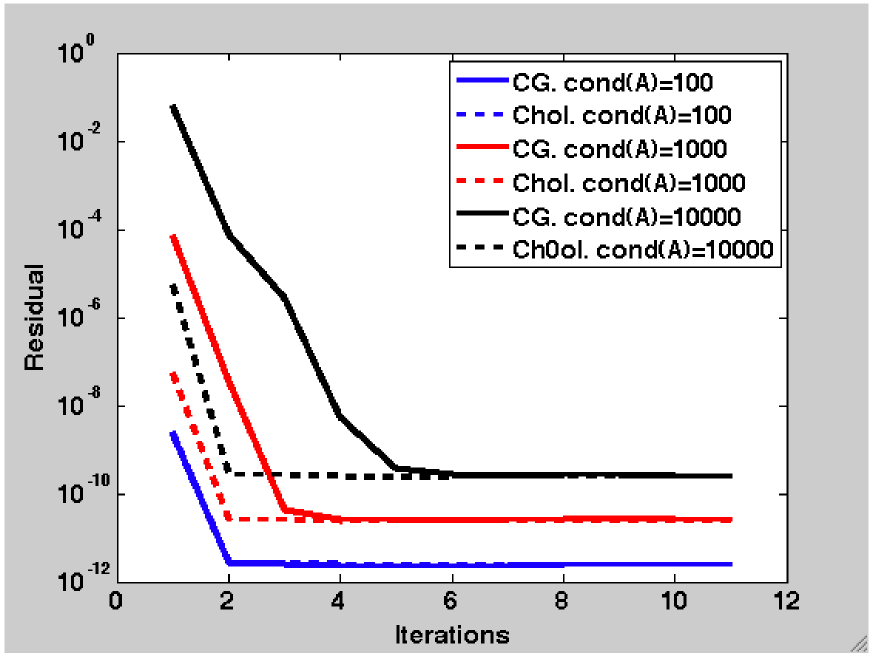
Relative error



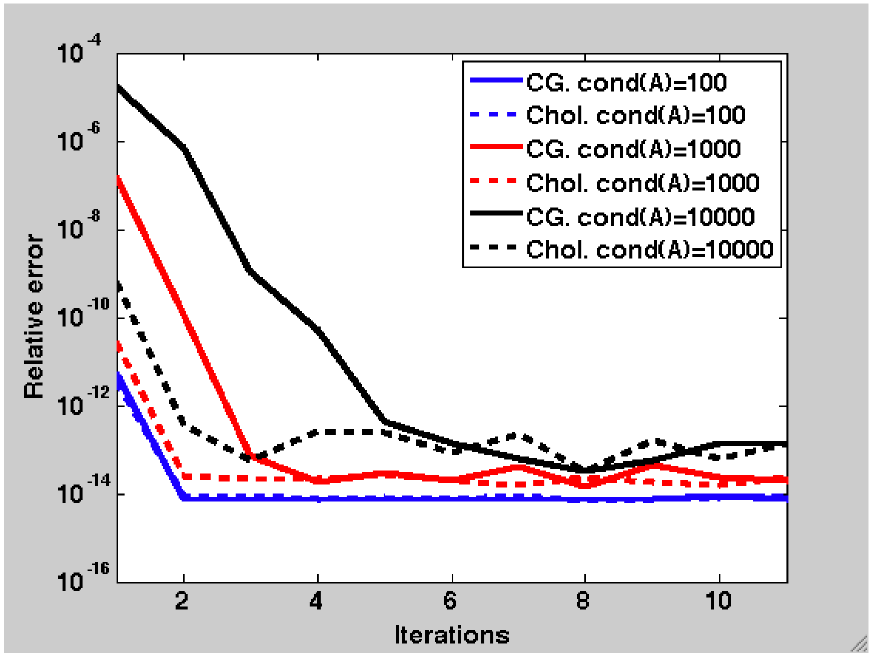
Cholesky IR v.s. CG IR: Accuracy

Matrix size n=1000. Varying condition numbers, cond(A)=100, 1000, 10000
CG steps: 100

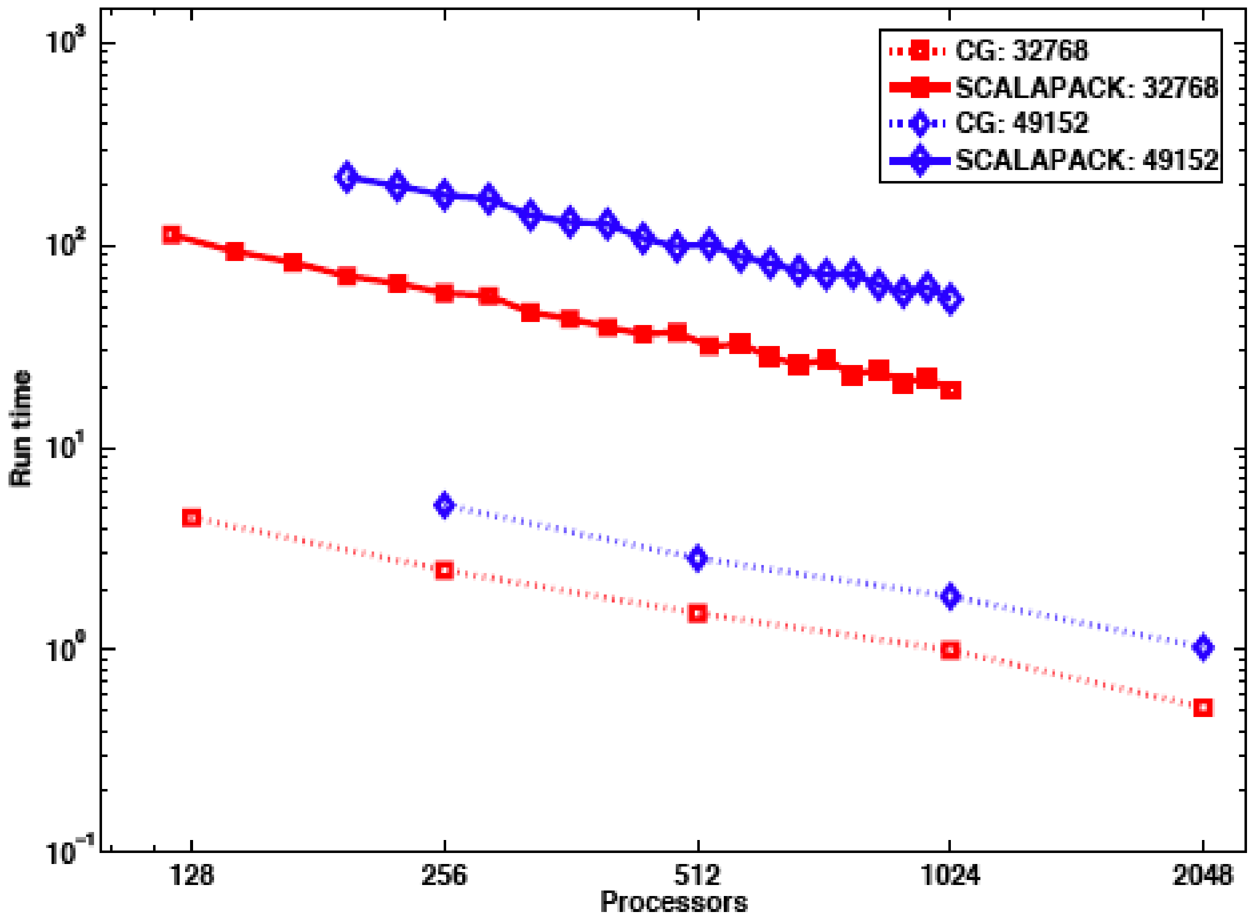
RESIDUAL



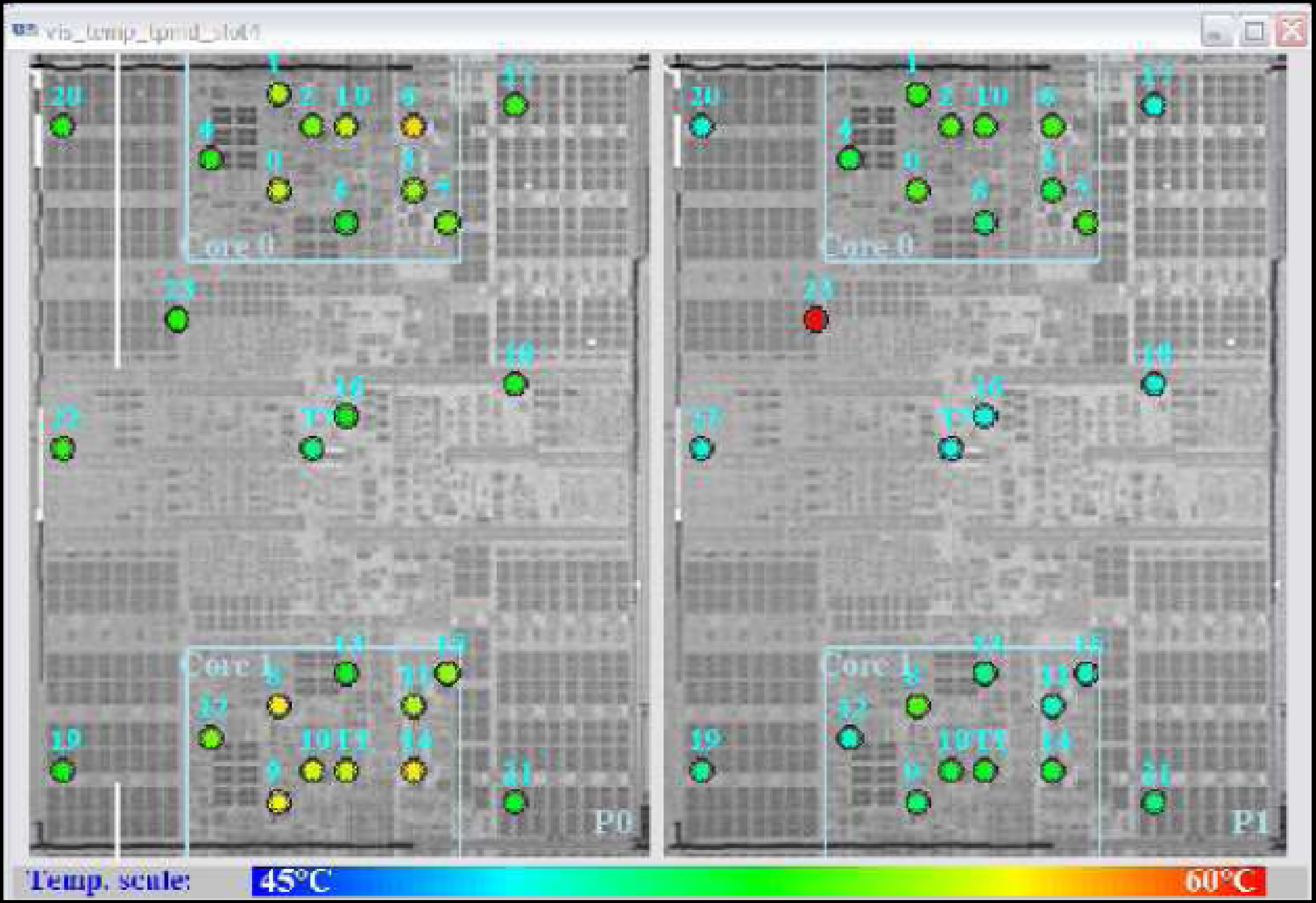
REL. ERROR



Cholesky IR v.s. CG IR: Scaleout



Actual On Chip Measurements

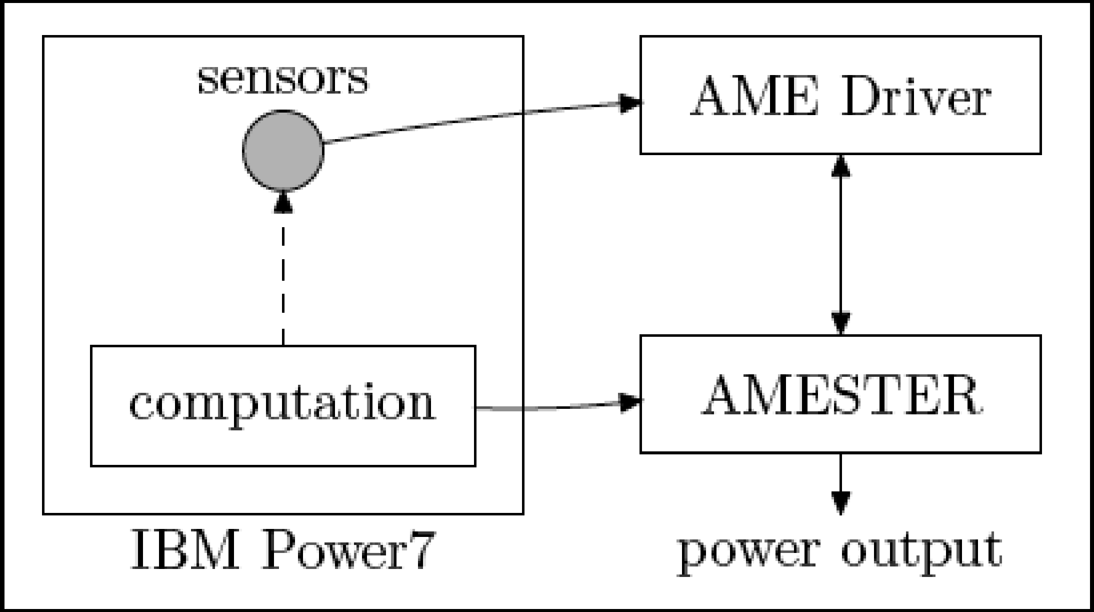


Power 7 chip has thermal sensors. Their readings can be calibrated to instantaneous power consumption with quite small error (<5%) (C. Lefurgy et al, Hot Chips 2010)

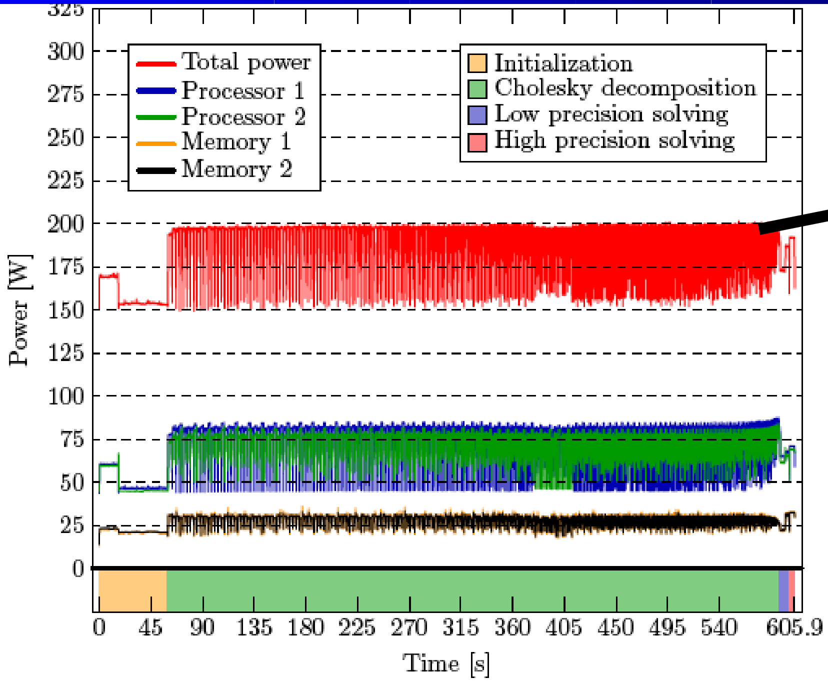
Measuring Power Consumption: A Interactive Framework

- AME driver that collects sensor data and calculates power consumption

- An external tool, AMESTER, connects to the service processor of the Power7 based server and gathers the readings. *Resolution of 75ms routinely achieved, potential for 1ms resolution is there. Power resolution 0.1Watts*
- No load on the system CPU / no measurement noise
- User application can also communicate with AMESTER: Put tags at run time

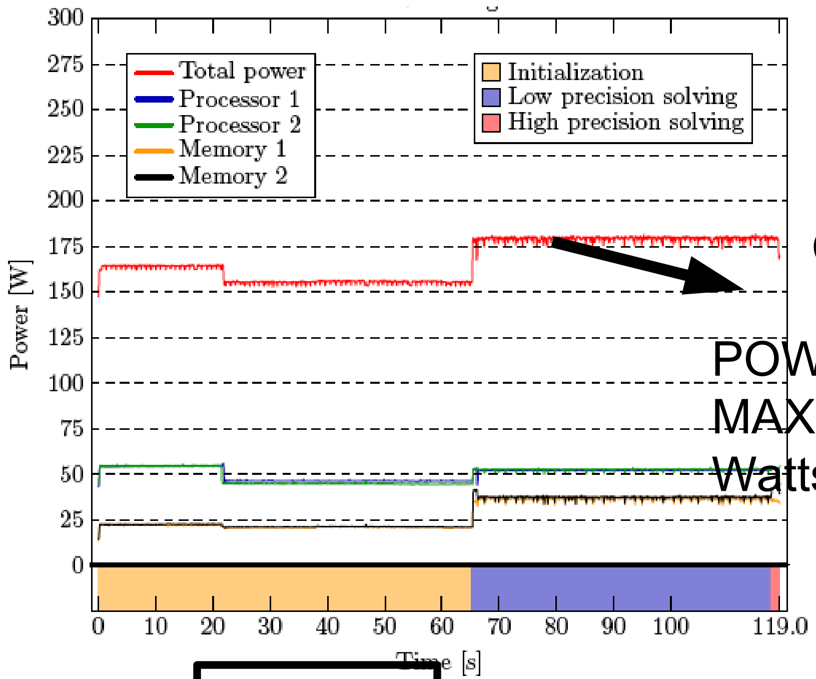


Power Consumption? Power7 system. H/W Power sensors



CHOLESKY

POWER MAX 200Watts



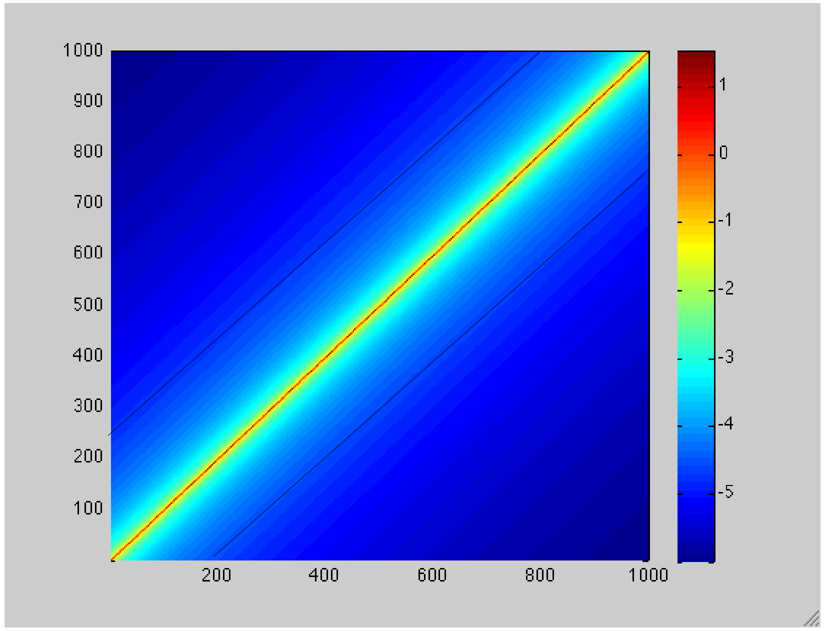
CG

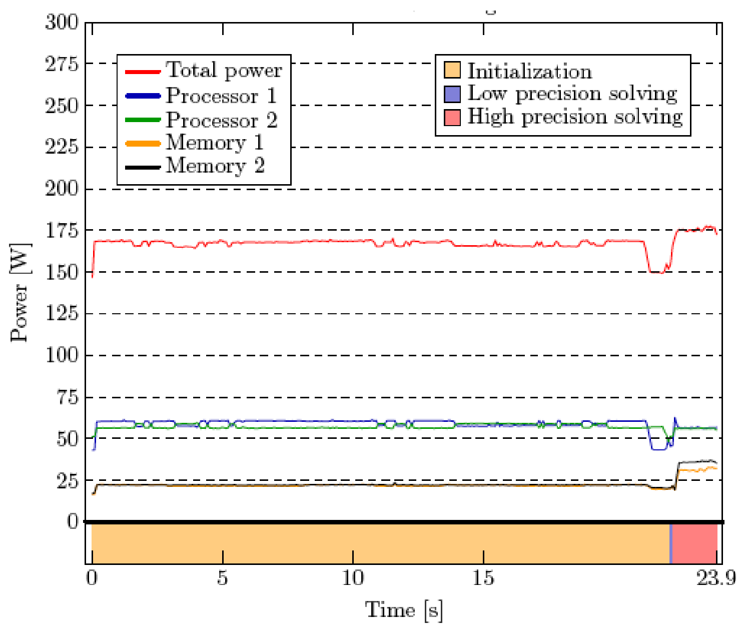
POWER MAX 179 Watts

CG 1 RHS	53.8 s	179.0 W, s.e. 1.8 W	9.6 kW· s	15.7	0.09
CG 32 RHS's	125.5 s	195.0 W, s.e. 10.8 W	24.6 kW· s	222.2	1.13
Cholesky	546.0 s	190.0 W, s.e. 13.5 W	103.7 kW· s	214.4	1.11

Can we push for more?

Data Analytics. Working with Covariance matrices. Typically they exhibit a decaying behavior away from the main diagonal. What if we make it banded? Converges!





Method	Time	Average power	Energy	GFlops	GFlops/W
banded CG 1 RHS	1.8 s	174.1 W, s.e. 4.9 W	0.3 kW· s	5.5	0.03
banded CG 32 RHS's	8.4 s	172.6 W, s.e. 14.2 W	1.5 kW· s	37.8	0.22
CG 1 RHS	53.8 s	179.0 W, s.e. 1.8 W	9.6 kW· s	15.7	0.09
CG 32 RHS's	125.5 s	195.0 W, s.e. 10.8 W	24.6 kW· s	222.2	1.13
Cholesky	546.0 s	190.0 W, s.e. 13.5 W	103.7 kW· s	214.4	1.11

IN GENERAL: CONSIDER

- ✓ LOW PRECISION, LOW COST, LOW POWER: LP
- ✓ HIGH PRECISION, HIGH POWER: HP
- ✓ Let SLV(A,y,) be a LP procedure approximating Ax=b
  SLV: Analog? Neuromorphic (spikes?), Neural Nets?, Machine Learning?

```
•   Compute initial solution: $x_0$=SLV(A,b)           Cost: really low time/power
•   Compute initial residual: $r_0 = b - Ax_0$          Cost: $n^2$
•   $k = 0$
•   REPEAT
•       Solve for residual: $d_k$= SLV(A,$r_k$)         Cost: really low time/power
        Update solution:    $x_{k+1} = x_k + d_k$       Cost: n
        Compute residual: $r_{k+1} = b - Ax_{k+1}$      Cost: $n^2$
        $k = k + 1$
1.  UNTIL $\|r_{k+1}\|$ □ tol
```

Key properties:

Overall cost: $O(n^2)$, instead of $O(n^3)$

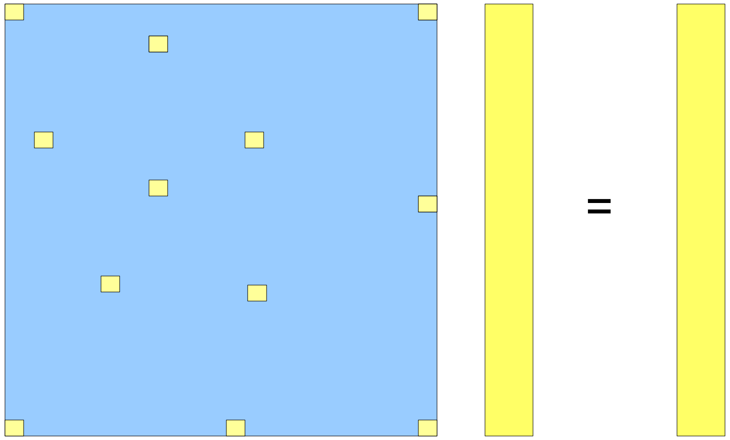
Most of arithmetic is performed on Low Power platform

Some thoughts on possible low power solutions:

- Learning approaches
  - Machine Learning / Statistical approach
  - Neural Networks
- **Neuromorphic approaches**
  - **Spike computing to simulate numerics**
- Hardware approaches
  - Accelerators (GPUs)
  - FPGAs
  - SPDs
  - Low reliability hardware (low voltage)

Examples...

Learning / stochastic approach: Reduce dimension by random sampling
XDATA DARPA PROJECT (2012-2016)



How will we decide which sampling?
- Estimate prion probabilities?
- Compare with “similar” cases?
- “Sparsify” full graph? Dynamicaly
- Changing network?
- Learn starting vector?

See recent work by Drineas, Mahoney, Claskson, Boutsidis and others)

Analog emulation or “inexact”

Digital computation: Threshold computing? (inexact bolean algebra)
- Specially designed FPGAs

Spike computing numerical linear algebra: investigation

The Roadmap to Exascale poses great challenges

- ...
- ...
- Power

Emphasis on power: Algorithms have a potentially very large margin of improvement. Accelerate computations by replacing power hungry digital arithmetic with green but noisy alternative computing: Low Prec. Digital / Neuromorphic/ Learning / Analog

How are we addressing the challenge: Introducing "noise" and stochasticity...allows for different kind of hybrid computing.

Algorithms: There is "plenty of room up there"