[HPC]^3 Workshop 2012

February 4-8, King Abdullah University of Science and Technology

Implications of Emerging Architectures on Hyperbolic Solvers

Hyperbolic PDEs are at the heart of critical energy and environmental applications, so we seek to improve the utility of their simulation in pursuit of scientific prediction and engineering design by improving (1) the fidelity of the models, (2) the quantified accuracy of the results, (3) the efficiency of the computations, and (4) the productivity of the users. The first two objectives push us into extreme computing environments, where the latter two objectives become increasingly daunting. Much mathematics and software appears to be missing if the hardware is to be used at its potential. Drawing upon three recent workshops on exascale hardware, multiphysics applications, and synchronization-reducing algorithms, as well as the second conclave of [HPC]^3, we point out ways in which the hyperbolic solvers community will need to adapt to emerging architectures and some means of adaptation.

[HPC]^3 Workshop 2012

February 4-8, King Abdullah University of Science and Technology

Implications of Emerging Architectures on Hyperbolic Solvers

David Keyes

Dean, Division of Mathematical and Computer Sciences and Engineering,

King Abdullah University of Science and Technology

Remembering Von Neumann

Born 28 Dec 1903 in Budapest, Hungary

Died 8 Feb 1957 in Washington D.C., USA

Remembering Von Neumann: June 1945

First Draft of a Report on the EDVAC

by

John von Neumann

Contract No. W–670–ORD–4926

Between the

United States Army Ordnance Department

and the

University of Pennsylvania

The logical control of the device, that is the proper sequencing of its operations can be most efficiently carried out by a central control organ. If the device is to be *elastic*, that is as nearly as possible all purpose, then a distinction must be made between the specific instructions given for and defining a particular problem, and the general control organs which see to it that these instructions are carried out…

Any device which is to carry out long and complicated sequences of operations (specifically of calculations) must have a considerable memory…

For partial differential equations the initial conditions and the boundary conditions may constitute an extensive numerical material, which must be remembered throughout a given problem…

For partial differential equations of the hyperbolic or parabolic type, integrated along a variable t, the (intermediate) results belonging to the cycle t must be remembered for the calculation of the cycle $t + dt$.

Remembering Von Neumann: October 1947

BULLETIN *of the* AMERICAN MATHEMATICAL SOCIETY

NUMERICAL INVERTING OF MATRICES OF HIGH ORDER

JOHN VON NEUMANN AND H. H. GOLDSTINE

PREFACE

The purpose of this paper is to derive rigorous error estimates in connection with the inverting of matrices of high order. The reasons for taking up this subject at this time in such considerable detail are essentially these: First, the rather widespread revival of mathematical interest in numerical methods, and the introduction of new procedures and devices which make it both possible and necessary to perform operations of this type on matrices of much higher orders than was even remotely practical in the past. Second, the fact that considerably diverging opinions are now current as to the extremely high or extremely low precisions which are required when inverting matrices of orders $n \geqq 10$. (Cf. in this connection footnotes 10, 11, 12 below.)

Remembering Von Neumann: November 1950

Numerical Integration of the Barotropic Vorticity Equation

By J. G. CHARNEY, R. FJÖRTOFT[1], J. von NEUMANN

The Institute for Advanced Study, Princeton, New Jersey[2]

V. The Computational Stability of the Finite Difference Equations

If the finite difference solution is to approximate closely the continuous solution, Δs and Δt must be small in comparison to the space and time scales of the physically relevant motions. But this does not alone insure accuracy; the small-scale motions for which there is inevitably a large distortion may possibly be amplified in the course of computation to such an extent that they will totally obscure the significant large-scale motions. In the following we shall derive the criteria which insure that no such amplification will take place. We shall employ a heuristic procedure which is, however, patterned after the rigorous method of COURANT, FRIEDRICHS, and LEWY (1928).

If, as before, the boundary is rectangular, the perturbation may be expanded into a finite Fourier series and it is enough to take an individual harmonic of the form

$$z' = e^{i(kx+\mu y+\nu t)},$$

where

$$k = \frac{\pi l}{p\,\Delta s} = \frac{\pi l}{L_x}\ (l = 1, 2, \ldots, p-1),$$

$$\mu = \frac{\pi m}{q\,\Delta s} = \frac{\pi m}{L_y}\ (m = 1, 2, \ldots, q-1), \qquad (24)$$

ν is the frequency, which may be complex, and L_x and L_y are the x and y dimensions of the grid rectangle. Substitution into (23) then gives, after some manipulation of terms,

Von Neumann's corpus

- **20 papers in physics**
  - **quantum mechanics, ergodic theory, fluid mechanics, geophysics**
- **60 papers in pure mathematics**
  - **set theory, geometry, game theory**
- **60 papers in applied and computational mathematics**
  - **functional analysis, numerical analysis, linear programming, statistics**
- **Other**
  - **computer science, economics, politics**

Von Neumann's service


JvN's Los Alamos badge photo

- **One of the first commissioners of the U.S. Atomic Energy Commission**
  - **today, called the Department of Energy**
  - **principal in the "Manhattan Project" at Los Alamos**
  - **developed the plutonium trigger for the uranium bomb**
  - **developed theories of nuclear proliferation and "mutually assured destruction"**
  - **developed plans for intercontinental ballistic missiles (ICBMs) and submarine launched missiles**
- **Consulted for**
  - **U.S. Army, CIA, RAND Corporation, Standard Oil (today Exxon-Mobile), GE, IBM**

Quotations from Von Neumann

- **"All stable processes we shall predict. All unstable processes we shall control."**

- ***"There is a time lapse between a mathematical discovery and the moment it becomes useful; […] the whole system seems to function without any direction, without any reference to usefulness, and without any desire to do things which are useful."***

- **"In mathematics you don't understand things. You just get used to them."**

Quotations from Von Neumann

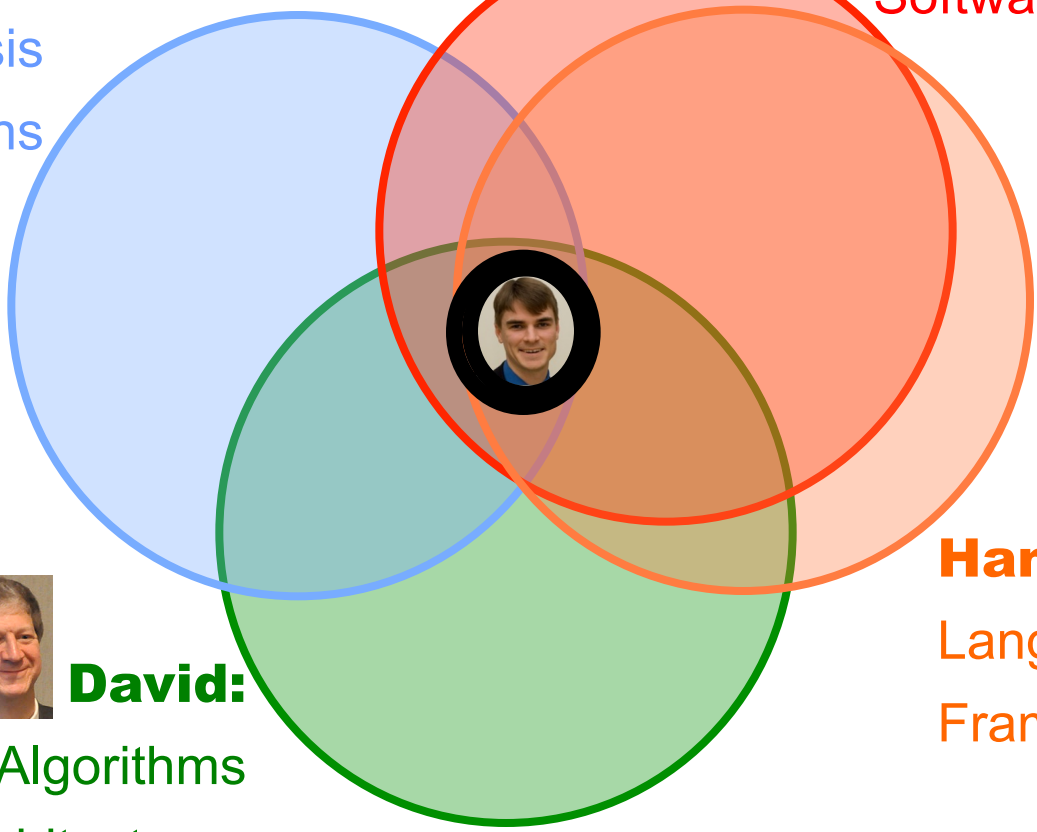
- **"The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work."**

- ***"It would appear that we have reached the limits of what it is possible to achieve with computer technology, although one should be careful with such statements, as they tend to sound pretty silly in 5 years."***

Workshop Context

Randy:
Analysis
Applications

Ridg:
Languages
Software



David:
Algorithms
Architectures

Hans Petter:
Languages
Frameworks

For background, see the archives at *www.exascale.org*

INTERNATIONAL EXASCALE SOFTWARE PROJECT ROADMAP 1.0

Jack Dongarra, Pete Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean-Claude Andre, David Barkai, Jean-Yves Berthou, Taisuke Boku, Bertrand Braunschweig, Franck Cappello, Barbara Chapman, Xuebin Chi

Alok Choudhary, Sudip Dosanjh, Thom Dunning, Sandro Fiore, Al Geist, Bill Gropp, Robert Harrison, Mark Hereld, Michael Heroux, Adolfy Hoisie, Koh Hotta, Yutaka Ishikawa, Fred Johnson

Sanjay Kale, Richard Kenway, David Keyes, Bill Kramer, Jesus Labarta, Alain Lichnewsky, Thomas Lippert, Bob Lucas, Barney Maccabe, Satoshi Matsuoka, Paul Messina, Peter Michielse, Bernd Mohr

Matthias Mueller, Wolfgang Nagel, Hiroshi Nakashima, Michael E. Papka, Dan Reed, Mitsuhisa Sato, Ed Seidel, John Shalf, David Skinner, Marc Snir, Thomas Sterling, Rick Stevens, Fred Streitz

Bob Sugar, Shinji Sumimoto, William Tang, John Taylor, Rajeev Thakur, Anne Trefethen, Mateo Valero, Aad van der Steen, Jeffrey Vetter, Peg Williams, Robert Wisniewski, Kathy Yelick

SPONSORS

Office of Science U.S. Department of Energy, NSF, ANR, CEA, CEPMIA, CRAY, EDF, EPSRC Engineering and Physical Sciences Research Council, FUJITSU, INRIA, GENCI, NVIDIA, RIKEN, 東京大学 The University of Tokyo

The International Exascale Software Roadmap,

J. Dongarra, P. Beckman, et al., *International Journal of High Performance Computer Applications* **25**(1), 2011, ISSN 1094-3420.

Contexts

- **The scientific computing world is at a crossroads with respect to the push towards extreme scale**
- **Proceeded steadily for three decades from mega- (1970s) to giga- (1988) to tera- (1998) to peta- (2008)**
  - **exa- is qualitatively different and will be much harder**
- **Software and numerics represented at this meeting will ultimately confront exascale to maintain sponsor relevance**
  - **though obviously, there are *many* fruitful directions in mathematics and software in the modeling of hyperbolic equations are *architecture-neutral***

Overlap of interests: IESP and HPC3

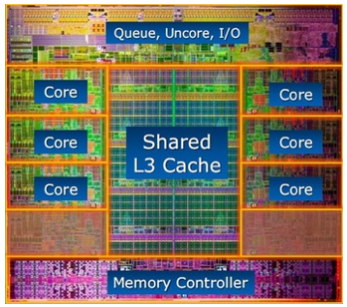
- **Exascale's extremes change the game**
  - **mathematicians are now on the front line**
    - **without contributions in the form of new mathematics (including statistics), the passage to the exascale will yield little fruit**
  - **mathematical scientists will find the computational power to do things many have wanted**
    - **room for creativity in "post-forward" problems**
    - **mathematical scientists will participate in cross-disciplinary integration – "third paradigm" *and* "fourth paradigm"**
    - **remember that *exascale at the lab* means *petascale on the desk***
- **Let's mention some mathematical opportunities, after quickly reviewing the hardware challenges**

Why exa- is different

Which steps of FMADD take more energy?

64-bit floating-point fused multiply add **or** moving four 64-bit operands 20 mm across the die

```
      934,569.299814557                    input
x          52.827419489135904              input
-----------------------------
= 49,370,884.442971624253823
+           4.20349729193958               input
-----------------------------
= 49,370,888.64646892                      output
```



20 mm

(Intel Sandy Bridge, 2.27B transistors)

Going across the die requires up to an order of magnitude more !

DARPA study predicts that by 2019:

- **Double precision FMADD flop: 11pJ**
- **cross-die per word access (1.2pJ/mm): 24pJ (= 96pJ overall)**

c/o T. Schulthess (ETHZ); c/o P. Kogge (ND) *et al.* DARPA study

Why exa- is different, cont.

Moore's Law (1965) does not end but Dennard's MOSFET scaling (1972) does

Table 1
Scaling Results for Circuit Performance

Device or Circuit Parameter	Scaling Factor
Device dimension t_{ox}, L, W	$1/\kappa$
Doping concentration N_a	κ
Voltage V	$1/\kappa$
Current I	$1/\kappa$
Capacitance $\epsilon A/t$	$1/\kappa$
Delay time/circuit VC/I	$1/\kappa$
Power dissipation/circuit VI	$1/\kappa^2$
Power density VI/A	1

Table 2
Scaling Results for Interconnection Lines

Parameter	Scaling Factor
Line resistance, $R_L = \rho L/Wt$	κ
Normalized voltage drop IR_L/V	κ
Line response time $R_L C$	1
Line current density I/A	κ

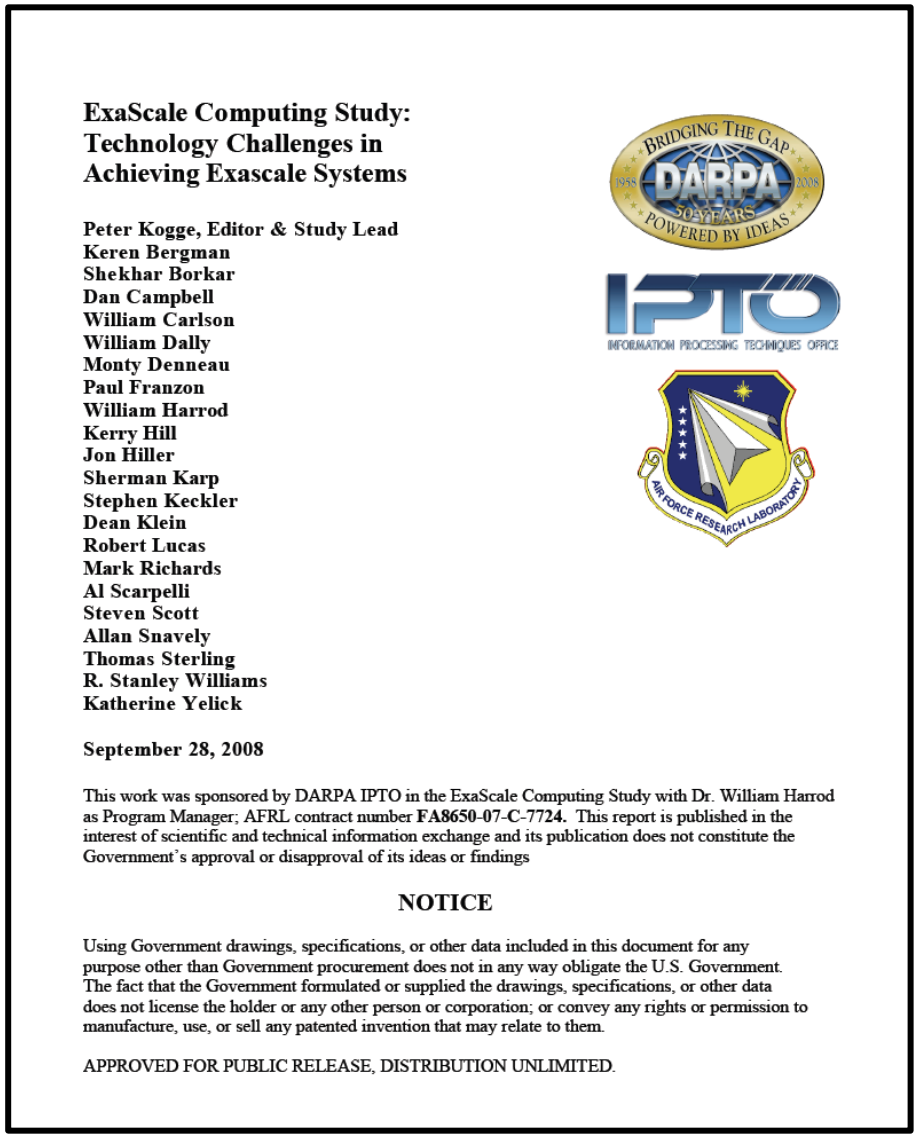
Robert Dennard, IBM
(inventor of DRAM, 1966)

Eventually processing will be limited by transmission

Parsimony in communication

"A pJ wasted on moving data is not only a pJ that is not available for communication, but also a pJ that must often be matched by other pJs that must be expended to monitor the progress of the data during the movement, to correct for any errors, and to manage the suspension and reawakening of any circuits that source or sink the data."

P. Kogge, *et al.* (2008), p. 244



ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems

Peter Kogge, Editor & Study Lead
Keren Bergman
Shekhar Borkar
Dan Campbell
William Carlson
William Dally
Monty Denneau
Paul Franzon
William Harrod
Kerry Hill
Jon Hiller
Sherman Karp
Stephen Keckler
Dean Klein
Robert Lucas
Mark Richards
Al Scarpelli
Steven Scott
Allan Snavely
Thomas Sterling
R. Stanley Williams
Katherine Yelick

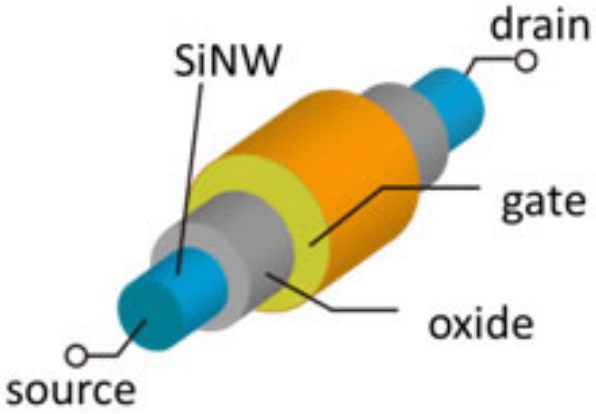
September 28, 2008

This work was sponsored by DARPA IPTO in the ExaScale Computing Study with Dr. William Harrod as Program Manager; AFRL contract number **FA8650-07-C-7724.** This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings

NOTICE

Bootstrapping to exa- with nano- on peta-

- **The 2011 Gordon Bell peak prize won by Yukihiro Hasegawa *et al.* of RIKEN at more than 3 Pflop/s sustained**
  - **First-principles calculation of electronic states of a silicon nanowire with 100,000 atoms on the "Kei computer"**
  - **taking industry leading 22 nm device size in 2011 down to 10 nm**
  - **10 nm not expected for CMOS transistors until 2015**
  - **bootstrapping the next generation of semiconductors!**



What will first "general purpose" exaflop/s machines look like?

- **Hardware: many potentially exciting paths beyond today's CMOS silicon-etched logic, but not commercially at scale within the decade**
- **Software: many ideas for general-purpose and domain-specific programming models beyond "MPI + X", but not penetrating the main CS&E workforce within the decade**

Prototype exascale hardware: a heterogeneous, distributed memory *GigaHz KiloCore MegaNode* system

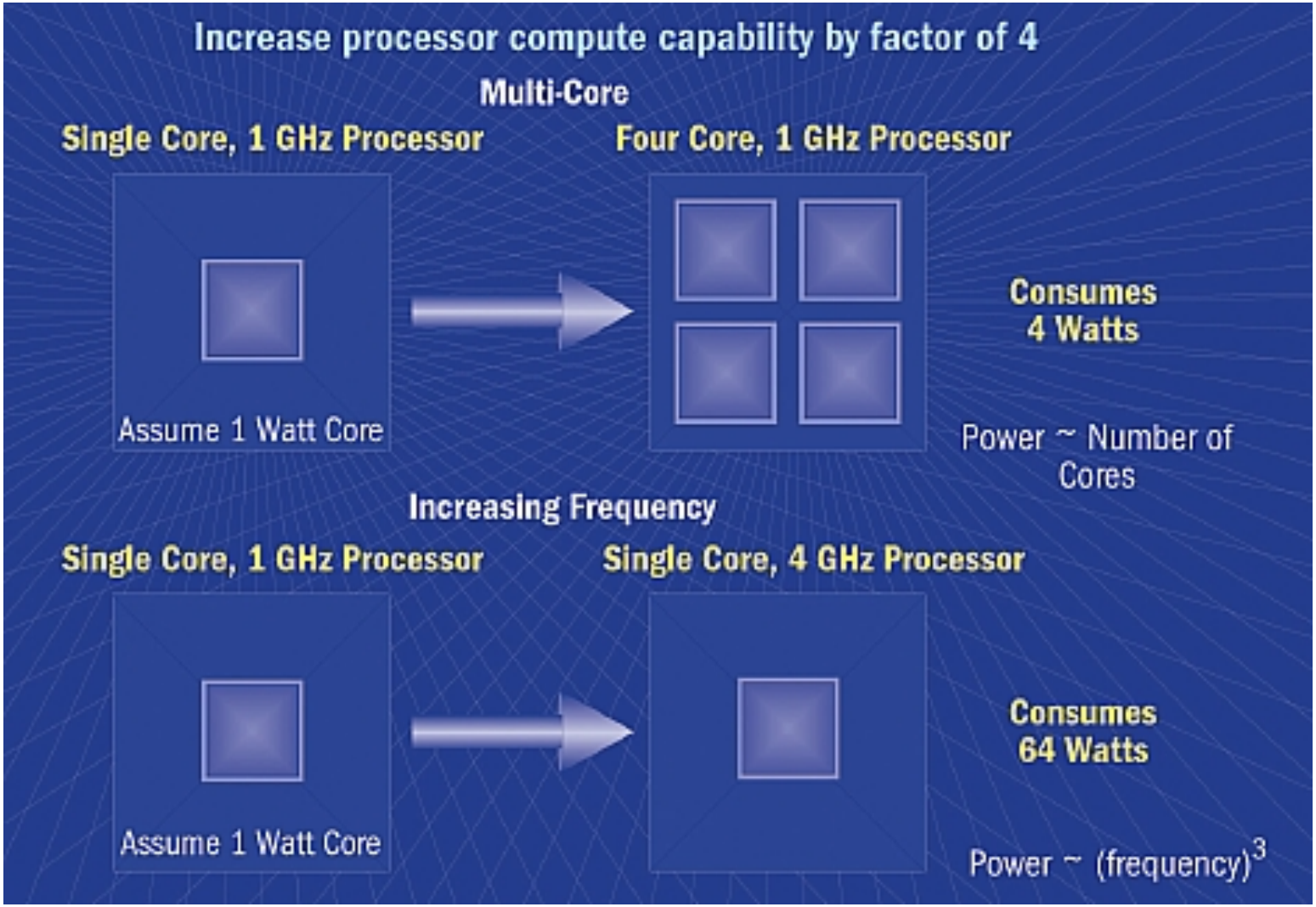
Systems	2009	2018	Difference Today & 2018
System peak	**2 Pflop/s**	**1 Eflop/s**	**O(1000)**
Power	**6 MW**	**~20 MW**	**~3**
System memory	0.3 PB	32 - 64 PB [**.03 Bytes/Flop**]	O(100)
Node performance	125 GF	1,2 or 15TF	O(10) – O(100)
Node memory BW	25 GB/s	2 - 4TB/s [**.002 Bytes/Flop**]	O(100)
Node concurrency	12	O(1k) or 10k	O(100) – O(1000)
Total Node Interconnect BW	3.5 GB/s	200-400GB/s (1:4 or 1:8 from memory BW)	O(100)
System size (nodes)	18,700	O(100,000) or O(1M)	O(10) – O(100)
Total concurrency	225,000	O(billion) [O(10) to O(100) for latency hiding]	O(10,000)
Storage	15 PB	500-1000 PB (>10x system memory is min)	O(10) – O(100)
IO	0.2 TB	60 TB/s (how long to drain the machine)	O(100)
MTTI	days	O(1 day)	- O(10)

c/o P. Beckman (ANL)

Some exascale themes (see reports)

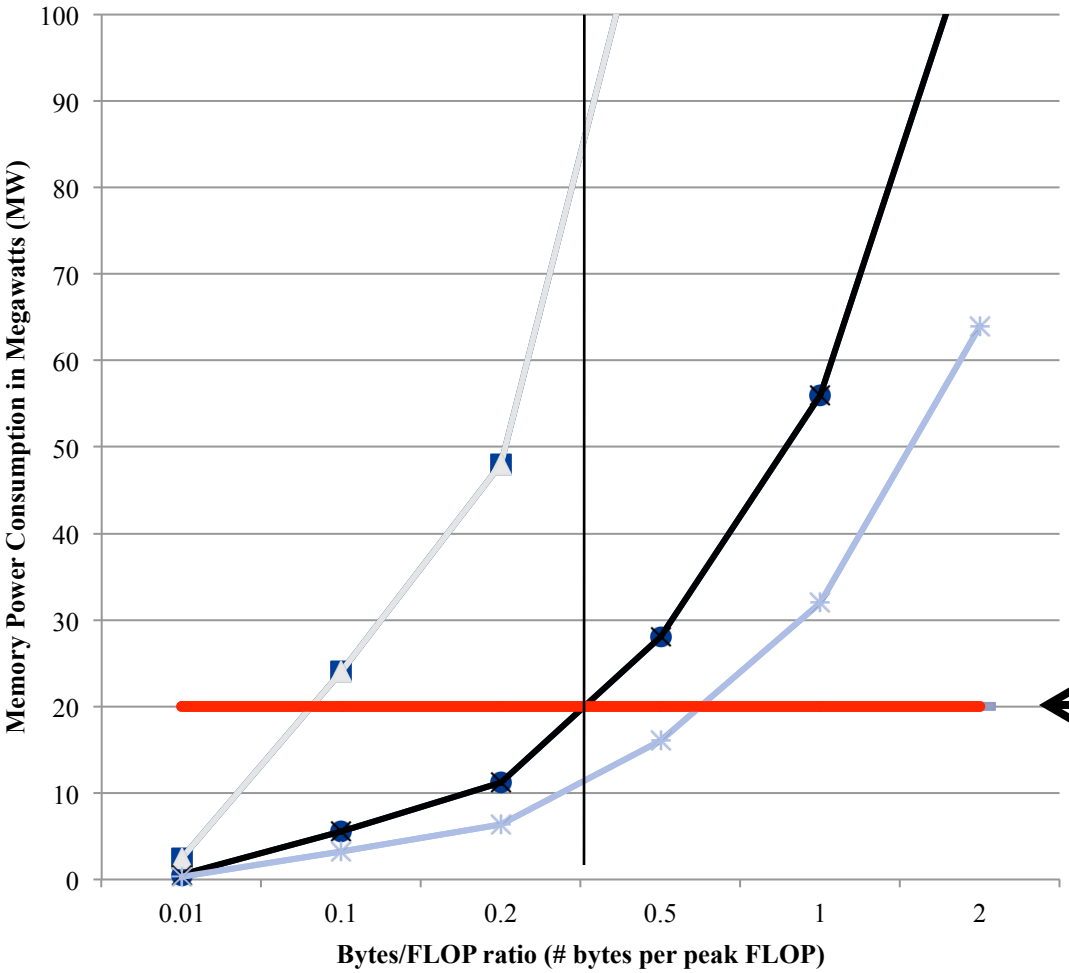
- **Clock rates cease to increase while arithmetic capacity continues to increase dramatically w/concurrency consistent with Moore's Law**
- **Storage capacity diverges exponentially below arithmetic capacity**
- **Transmission capacity diverges exponentially below arithmetic capacity**
- **Mean time between hardware interrupts shortens**
- **Billions of dollars of scientific software hang in the balance until better algorithms arrive to span the architectural gap**

Hurdle #1: power requires slower clocks and greater concurrency



c/o *SciDAC Review* 16, February 2010

Hurdle #2: memory bandwidth could eat up the entire power budget



Rule of prefixes ☺

"Pico times Exa is Mega"

Example:

"PicoJ times Exa/s is MegaW"

Stacked JEDEC 30pj/bit 2018 ($20M)

Advanced 7pj/bit Memory ($100M)

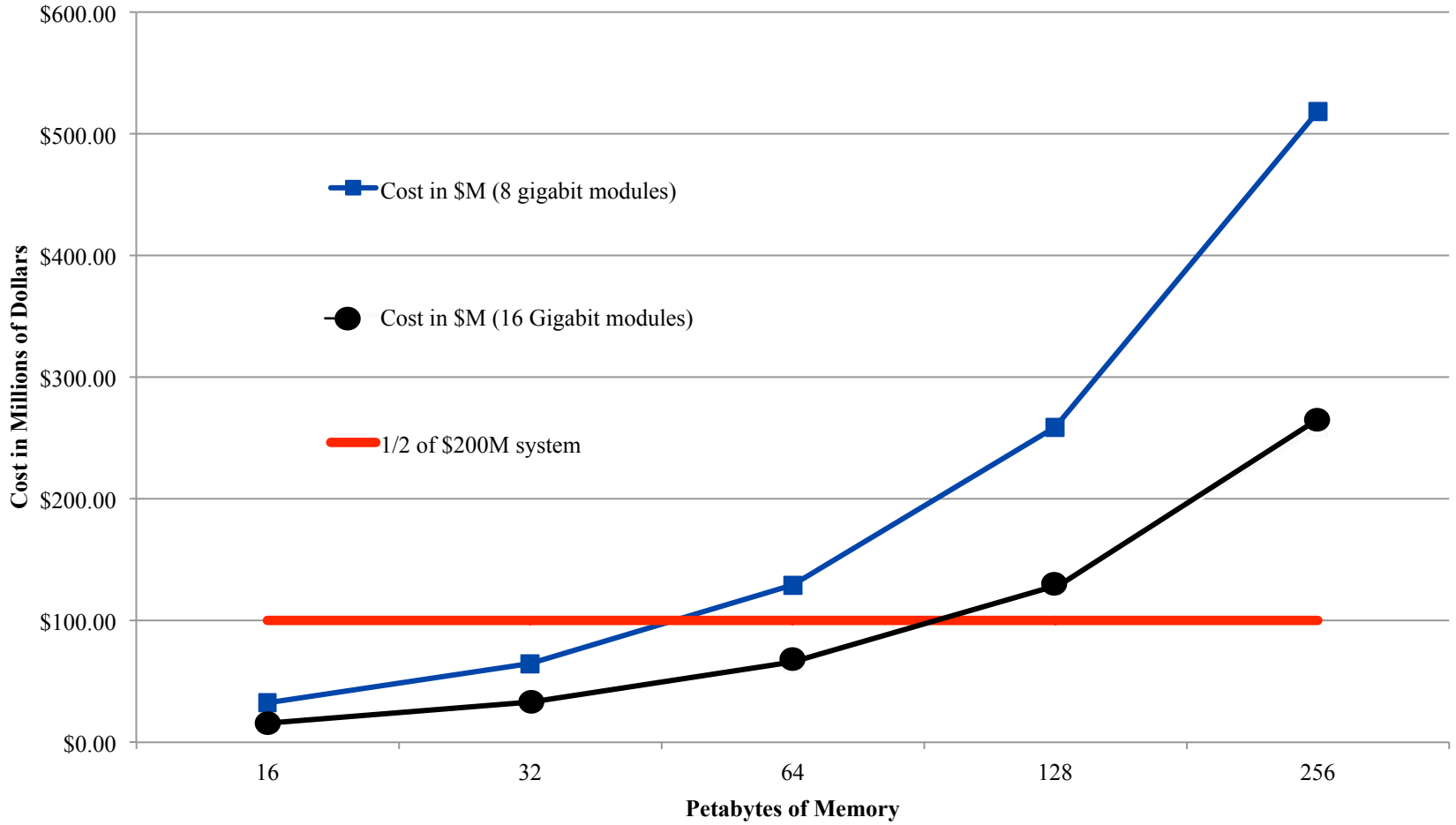
Enhanced 4pj/bit Advanced Memory ($150M cumulative)

Feasible Power Envelope (20MW)

20 MW is power consumption of a town of 14,000 people in an OECD country

c/o John Shalf (LBNL)

Hurdle #3: memory capacity could eat up the entire fiscal budget



c/o John Shalf (LBNL)

Implications of operating on the edge

- **Draconian reduction required in power per flop and per byte will make computing and copying data less reliable**
  - **voltage difference between "0" and "1" will be reduced**
  - **circuit elements will be smaller and subject to greater physical noise per signal**
  - **there will be more errors that must be caught and corrected**
- **Power may have to be cycled off and on or clocks slowed and speeded based on compute schedules and based on cooling capacity**
  - **makes per node performance rate unreliable**
  - **"Race-to-Halt" (RTH) is proposed as an alternative to power cycling, due to persistent leakage currents (budget similar to human brain basal energy), but thermal protection may overrule**

Implications of operating on the edge

- **Expanding the number of nodes (processor-memory units) beyond 10^6 would *not* a serious threat to algorithms that lend themselves to well-amortized precise load balancing**
  - **provided that the nodes are performance reliable**
- **A real challenge is expanding the number of cores on a node to 10^3**
  - **must be done while memory and memory bandwidth per node expand by (at best) ten-fold less (basically "strong" scaling)**
- **It is already about 10^3 slower to to retrieve an operand from main DRAM memory than to perform an arithmetic operation – will get worse by a factor of ten**
  - **almost all operands must come from registers or upper cache**

Why push to extreme scale? (DOE CSGF application essay question #3)

- Better resolve model's full, natural range of length or time scales
- Accommodate physical effects with greater fidelity
- Allow the model degrees of freedom in all relevant dimensions
- Better isolate artificial boundary conditions (e.g., in PDEs) or better approach realistic levels of dilution (e.g., in MD)
- Combine multiple complex models
- Solve an inverse problem, or perform data assimilation
- Perform optimization or control
- Quantify uncertainty
- Improve statistical estimates

"Third paradigm"

- Operate without models (machine learning)

"Fourth paradigm"

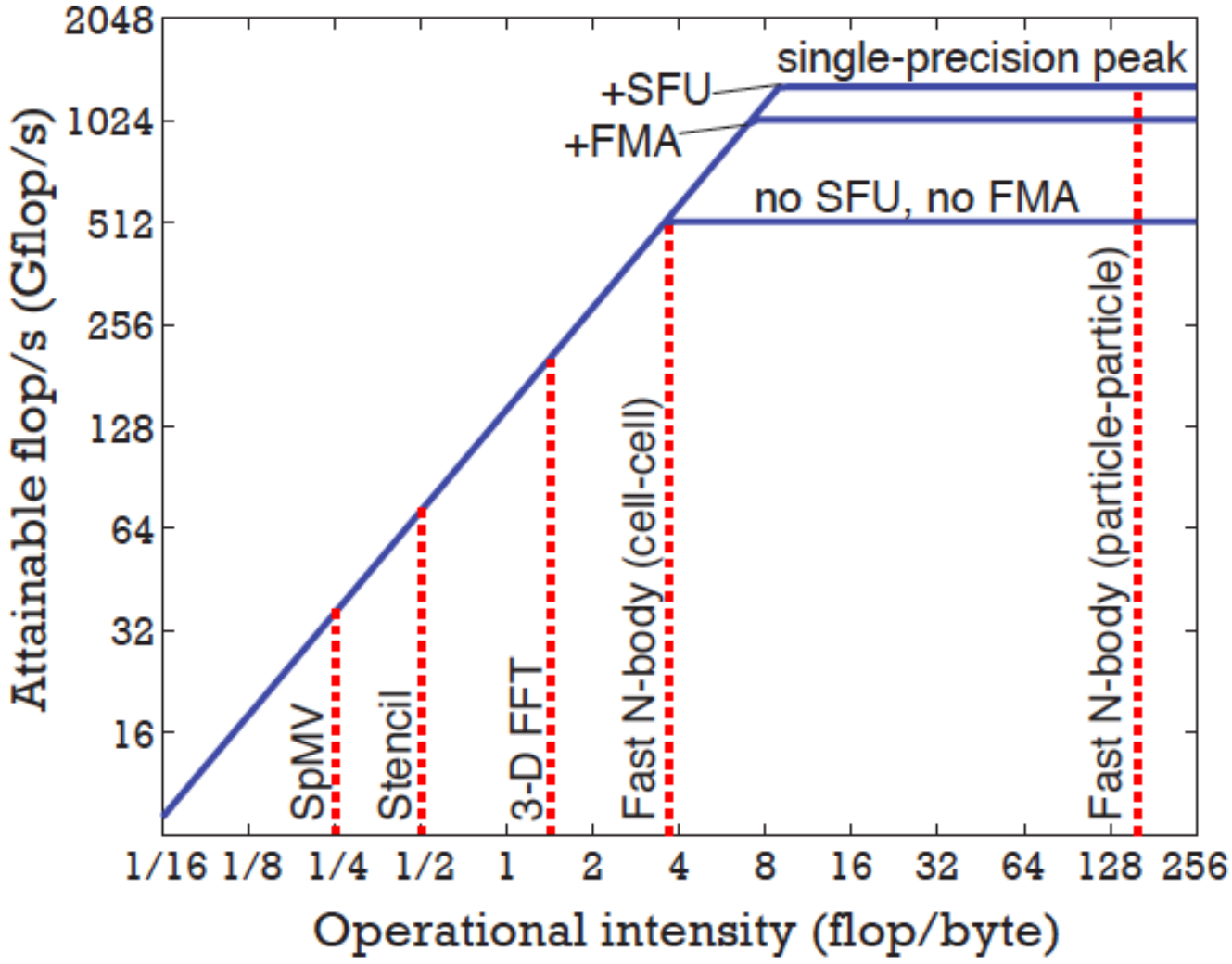
"Missing" mathematics

- **New formulations with**
  - **greater arithmetic intensity (flops per bytes moved into and out of registers and upper cache)**
  - **reduced communication**
  - **reduced synchronization**
  - **assured accuracy with (adaptively) less floating-point precision**
- **Quantification of trades between limiting resources**
- ***Plus* all of the exciting analytical agendas that exascale is meant to exploit**

Arithmetic intensity illustration

Roofline model of numerical kernels on an NVIDIA C2050 GPU (Fermi). The 'SFU' label is used to indicate the use of special function units and 'FMA' indicates the use of fused multiply-add instructions.

(The order of fast multipole method expansions was set to p = 15.)



c/o L. Barba (BU); cf. "Roofline Model" of S. Williams (Berkeley)

Miracles "need not apply"

- **We should not expect to escape causal dependencies**
  - **if the input-to-output map of a problem description has nonlocal data dependencies, and if we need the solution accurately everywhere, we will have nonlocal communication**
- **But we should ask fundamental questions:**
  - **for the science of interest, do we need to evaluate the output everywhere?**
    - **note that FEniCS allows users to specify this**
  - **is there another formulation that can produce the same required scientific observables in less time and energy?**

How are most workhorse simulations implemented at the infra-petascale today?

- **Explicit methods or iterative iterative methods based on stencil-op/SpMV with data decomposition and message-passing**
  - **each individual processor works on a portion of the original problem and exchanges information at its boundaries with other processors that own portions with which it interacts causally, to evolve in time or to establish equilibrium**
  - **computation and neighbor communication are both fully parallelized and their ratio remains constant in weak scaling**
- **The programming model is SPMD/BSP/CSP**
  - **Single Program, Multiple Data**
  - **Bulk Synchronous Programming**
  - **Communicating Sequential Processes**

Estimating scalability

- **Given complexity estimates of the leading terms of:**
  - **the concurrent computation (per iteration phase)**
  - **the concurrent communication**
  - **the synchronization frequency**
- **And a model of the architecture including:**
  - **internode communication (network topology and protocol reflecting horizontal memory structure)**
  - **on-node computation (effective performance parameters including vertical memory structure)**
- **One can estimate optimal concurrency and optimal execution time**
  - **on per-iteration basis**
  - **simply differentiate time estimate in terms of problem size *N* and processor number *P* with respect to *P***

3D stencil computation weak scaling

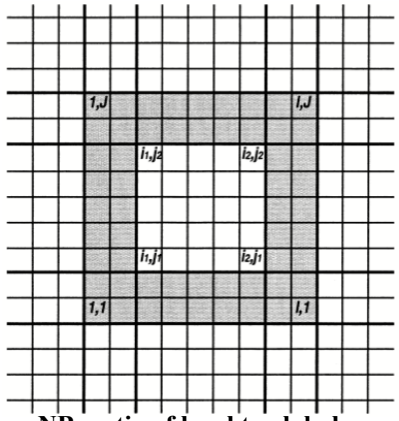
(assume fast local network, tree-based global reductions)

- **Total wall-clock time per iteration (ignoring local comm.)**

$$T(N,P) = A\frac{N}{P} + C\log P$$

- **For optimal P,** $\dfrac{\partial T}{\partial P} = 0$ **, or** $-A\dfrac{N}{P^2} + \dfrac{C}{P} = 0$

or $P_{opt} = \dfrac{A}{C}N$

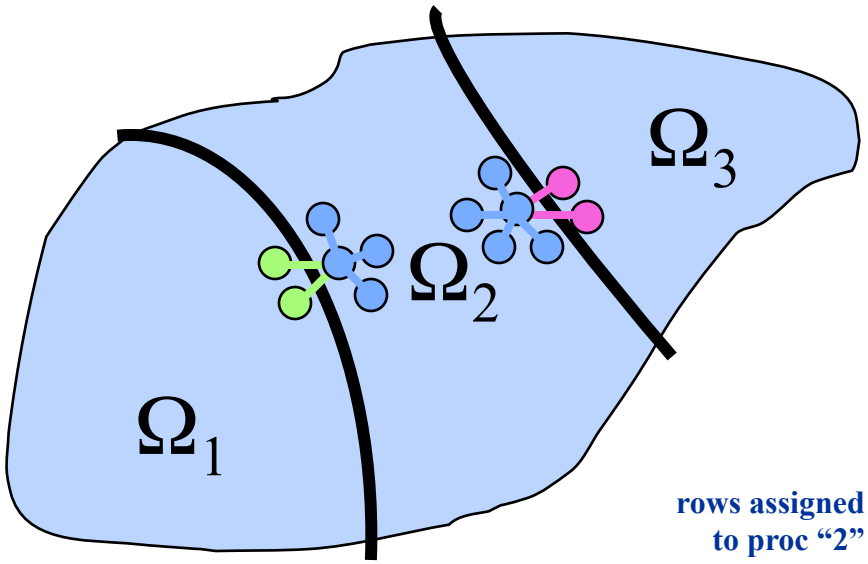


NB: ratio of local to global communication must be studied before accepting this asymptotic estimate that neglects local; wide halos are increasingly common

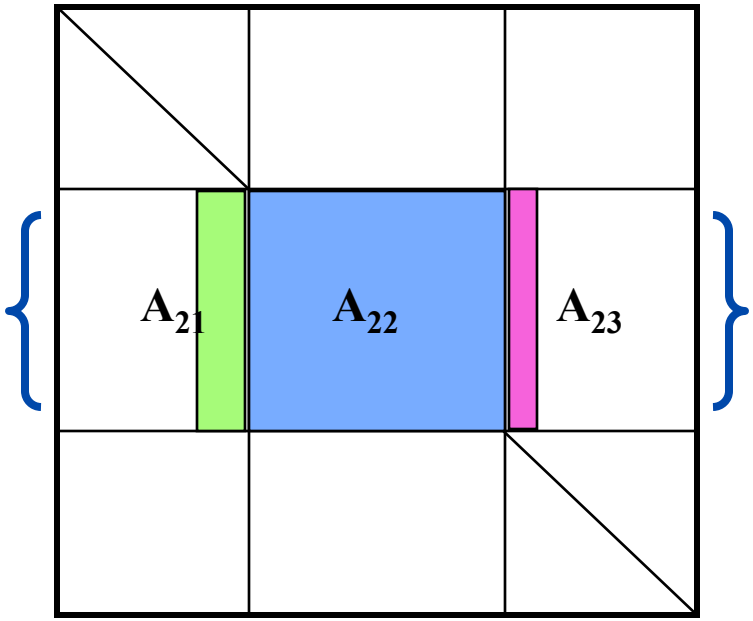
- **P can grow linearly with N, and running time increases "only" logarithmically – *as good as weak scaling can be!***
- **Problems: (1) assumes perfect synchronization,**

(2) log of a billion may be "large"

SPMD parallelism w/ domain decomposition: *an endangered species?*

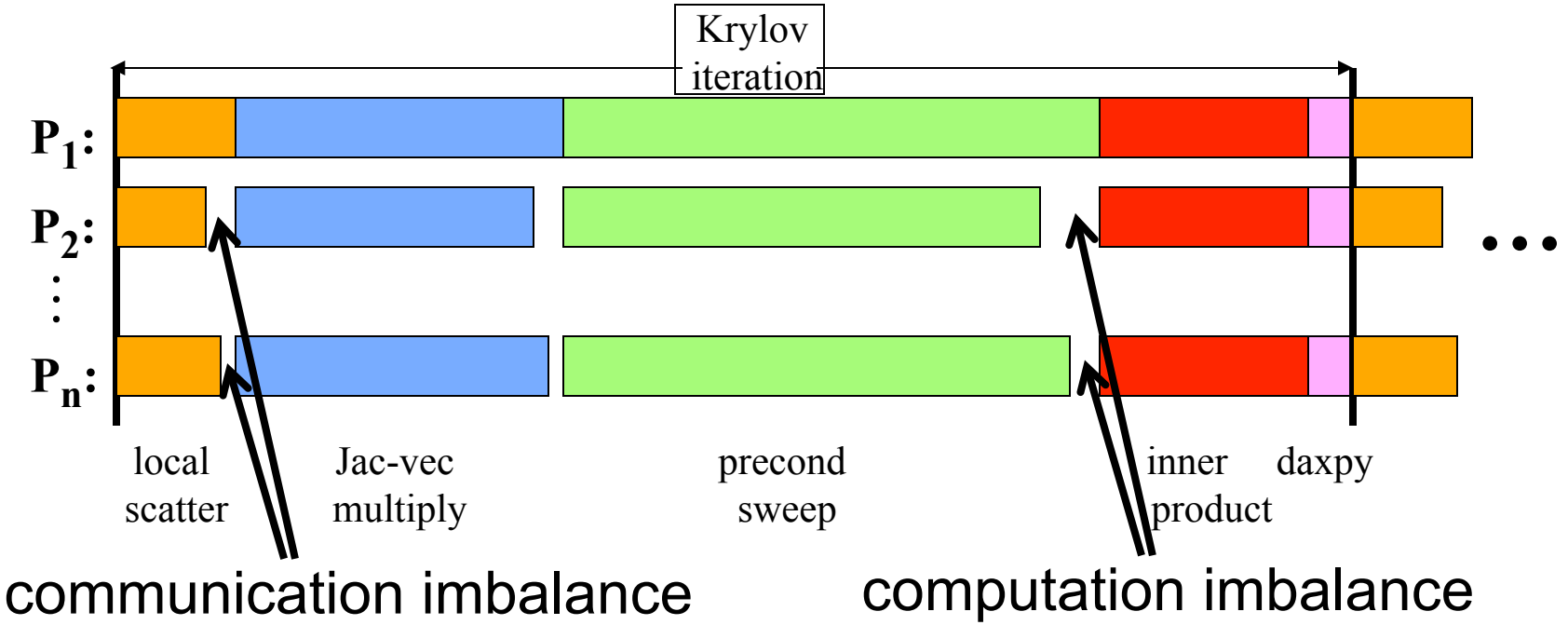


rows assigned to proc "2"



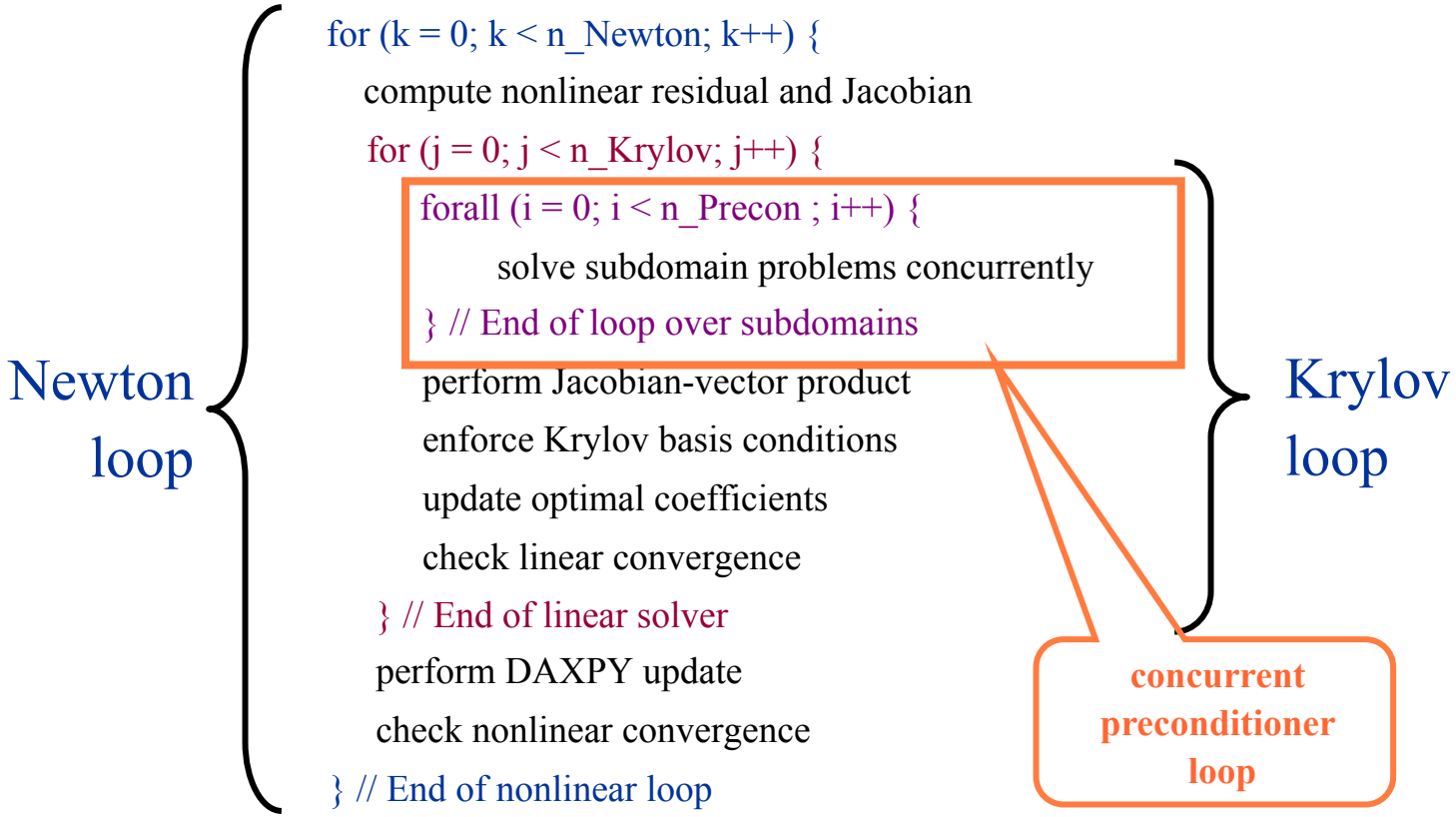
Partitioning of the grid induces block structure on the system matrix (Jacobian)

Workhorse innards: e.g., Krylov-Schwarz, a bulk synchronous implicit solver



Idle time due to load imbalance becomes a challenge at, say, one billion threads, when *one* thread can hold up *all* of the rest at a synchronization point

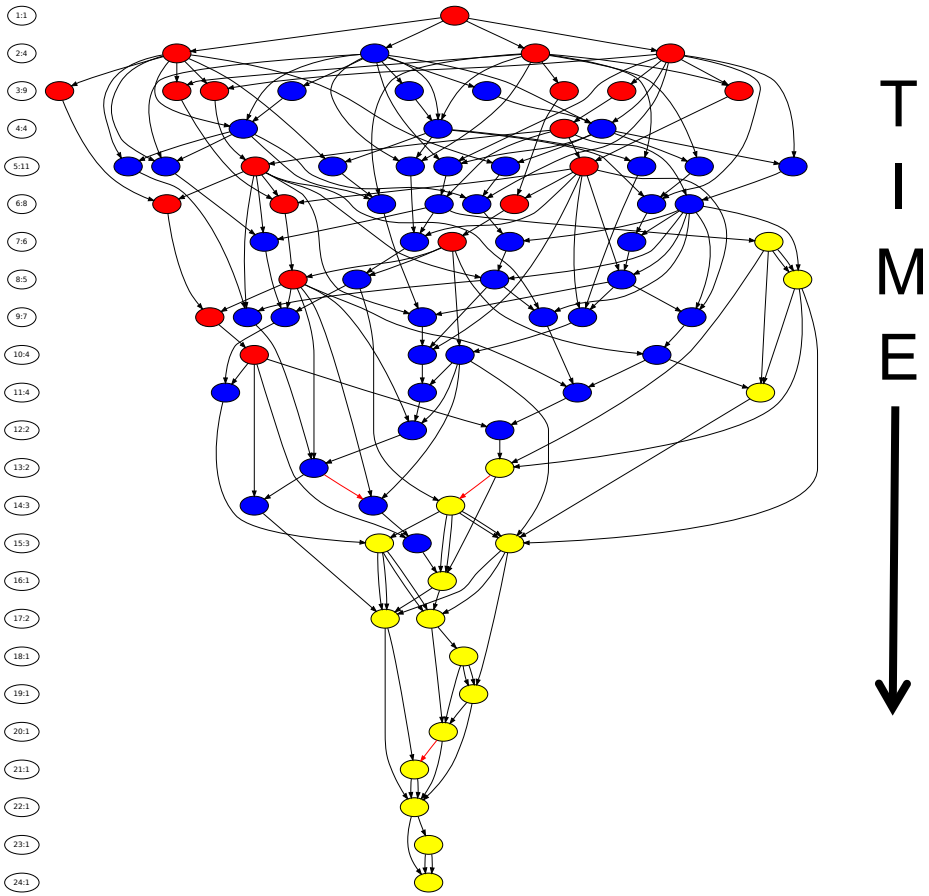
Our programming idiom is nested loops, e.g., for Newton-Krylov-Schwarz



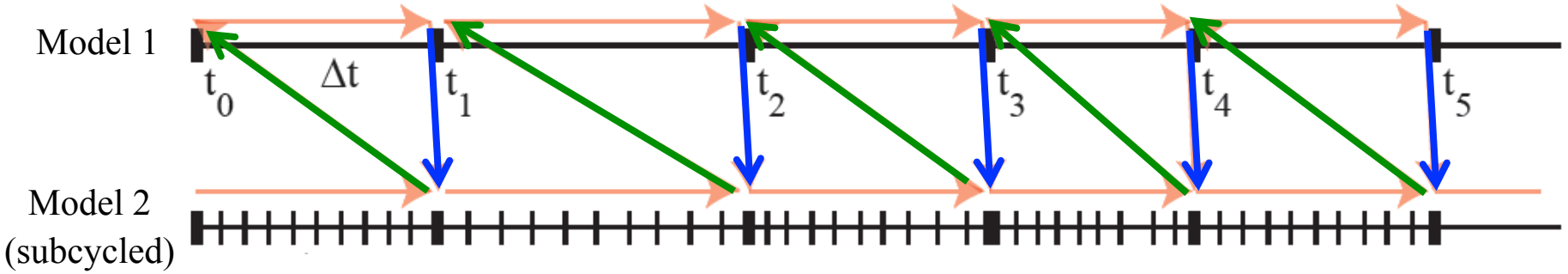
Outer loops (not shown): continuation, implicit timestepping, optimization

These loops, with their artifactual orderings, need to be replaced with DAGs

- Diagram shows a dataflow ordering of the steps of a 4×4 symmetric generalized eigensolver
- Nodes are tasks, color-coded by three types, and edges are data dependencies
- Original algorithm completed each task before starting the next (subroutine boundaries) and each task took longer (overconstrained by loop-based operation order)



Multiphysics w/ legacy codes: *an endangered species?*



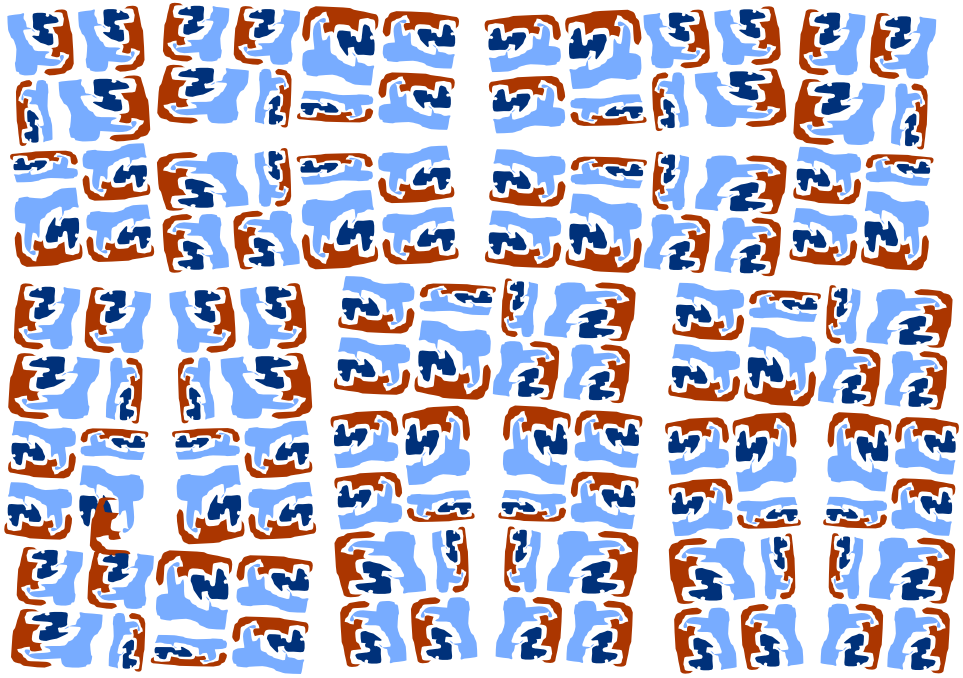
- **Many multiphysics codes operate like this, where the models may occupy the same domain in the bulk (e.g., reactive transport) or communicate at interfaces (e.g., ocean-atmosphere)***
- **The data transfer cost represented by the blue and green arrows may be much higher than the computation cost of the models, even apart from first-order operator splitting error and possible instability**

*see **ANL MCS-TM 321** from DOE ICiS workshop (Keyes, *et al.*, 2011)

Many research frontiers have the algebraic and software structure of multiphysics

- **Exascale is motivated by these:**
  - **uncertainty quantification, inverse problems, optimization, immersive visualization and steering**
- **These may carry auxiliary data structures to/from which blackbox model data is passed and they act like just another "physics" to the hardware**
  - **pdfs, Lagrange multipliers, etc.**
- **Today's separately designed blackbox algorithms for these may not live well on exascale hardware: co-design may be required to reduce data motion**

Multiphysics layouts must invade blackboxes



- Each application must first be ported to extreme scale (distributed, hierarchical memory)
- Then applications may need to be interlaced at the data structure level to minimize copying and allow work stealing at synchronization points

Bad news/good news (1)

- **Users will have to control data motion**
  - **carries the highest energy cost in the exascale computational environment**
- **Users will finally get the privilege of controlling the vertical data motion**
  - **horizontal data motion under control of users under *Pax MPI*, already**
  - **but vertical replication into caches and registers was (until now with GPUs) scheduled and laid out by hardware and runtime systems, mostly invisibly to users**

Bad news/good news (2)

- **"Optimal" formulations and algorithms may lead to poorly proportioned computations for exascale hardware resource balances**
  - **today's "optimal" methods presume flops are expensive and memory and memory bandwidth are cheap**
- **Architecture may lure users into more arithmetically intensive formulations (e.g., fast multipole, lattice Boltzmann, rather than mainly PDEs)**
  - **tomorrow's optimal methods will (by definition) evolve to conserve what is expensive**

Bad news/good news (3)

- **Hardware nonuniformity may force abandonment of the Bulk Synchronous Programming (BSP) paradigm**
  - **it will be impossible for the user to control load balance sufficiently to make it work**
- **Hardware and algorithmic nonuniformity will be indistinguishable at the performance level**
  - **good solutions for the dynamically load balancing in systems space will apply to user space, freeing users**

Bad news/good news (4)

- **Default use of high precision may come to an end, as wasteful of storage and bandwidth**
  - **we will have to compute and communicate "deltas" between states rather than the full state quantities, as we did when double precision was expensive (e.g., iterative correction in linear algebra)**
  - **a combining network node will have to remember not just the last address, but also the last values, and send just the deltas**
- **Equidistributing errors properly while minimizing resource use will lead to innovative error analyses in numerical analysis**

Bad news/good news (5)

- **Fully deterministic algorithms may simply come to be regarded as too synchronization-vulnerable**
  - **Rather than wait for data, we may need to infer it and move on**
- **A rich numerical analysis of algorithms that make use of statistically inferred "missing" quantities may emerge**
  - **Machine learning, sensitivity analysis, estimates from approximate Green's functions are among the techniques that will be called up to analyze or minimize the numerical perils of operating without delayed data**

Philosophy of an algorithmicist

- **Applications are *given* (as function of time)**
- **Architectures are *given* (as function of time)**
- **Algorithms and software *must be adapted or created* to bridge to hostile architectures for the sake of the complex applications**
  - **as important as ever today, with transformation of Moore's Law from speed-based to concurrency-based, due to power considerations**
  - **scalability still important, but new memory-bandwidth stresses arise when on-chip memories are shared**
  - **greatest challenge is lack of performance robustness of individual cores, which can spoil load balance**
- **Knowledge of algorithmic capabilities can usefully influence**
  - **the way applications are formulated**
  - **the way architectures are constructed**
- **Knowledge of application and architectural opportunities can usefully influence algorithmic development**

How will PDE computations adapt?

- **Programming model will still be message-passing on the outside (due to large legacy code base), adapted to multicore processors beneath a relaxed synchronization MPI-like interface**
- **Load-balanced blocks, scheduled today with nested loop structures will be separated into critical and non-critical parts**
- **Critical parts will be scheduled with directed acyclic graphs (DAGs)**
- **Noncritical parts will be made available for work-stealing in economically sized chunks**

Adaptation to asynchronous programming styles

- **To take full advantage of such asynchronous algorithms, we need to develop greater expressiveness in scientific programming**
  - **create separate threads for logically separate tasks, whose priority is a function of algorithmic state, not unlike the way a time-sharing OS works**
  - **join priority threads in a directed acyclic graph (DAG), a task graph showing the flow of input dependencies; fill idleness with noncritical work or steal work**
- **Steps in this direction**
  - **Asynchronous Dynamic Load Balancing (ADLB) [Lusk (Argonne), 2009]**
  - **Asynchronous Execution System [Steinmacher-Burrow (IBM), 2008]**

Evolution of Newton-Krylov-Schwarz: breaking the synchrony stronghold

- Can write code in styles that do not require artifactual synchronization
- Critical path of a nonlinear implicit PDE solve is essentially

  … lin_solve, bound_step, update; lin_solve, bound_step, update …
- However, we often insert into this path things that could be done less synchronously, because we have limited language expressiveness
  - Jacobian and preconditioner refresh
  - convergence testing
  - algorithmic parameter adaptation
  - I/O, compression
  - visualization, data mining

Some major challenges that stay the same in peta to exa

- Poor scaling of collectives due to internode latency
- Poor performance of SpMV due to shared intranode bandwidth
- Poor scaling of coarse grids due to insufficient concurrency
- Lack of reproducibility due to floating point noncommutativity and algorithmic adaptivity (including autotuning) in efficient production mode
- Difficulty of understanding and controlling vertical memory transfers

Sources of nonuniformity

- **System**
  - **manufacturing, OS jitter, TLB/cache performance variations, network contention, dynamic power management, soft errors, hard component failures, software-mediated resiliency, etc.**
- **Algorithmic**
  - **physics at gridcell/particle scale (e.g., table lookup, equation of state, external forcing), discretization adaptivity, solver adaptivity, precision adaptivity, etc.**
- **Effects are similar when it comes to waiting at synchronization points**
- **Possible solutions for system nonuniformity will improve programmability, too**

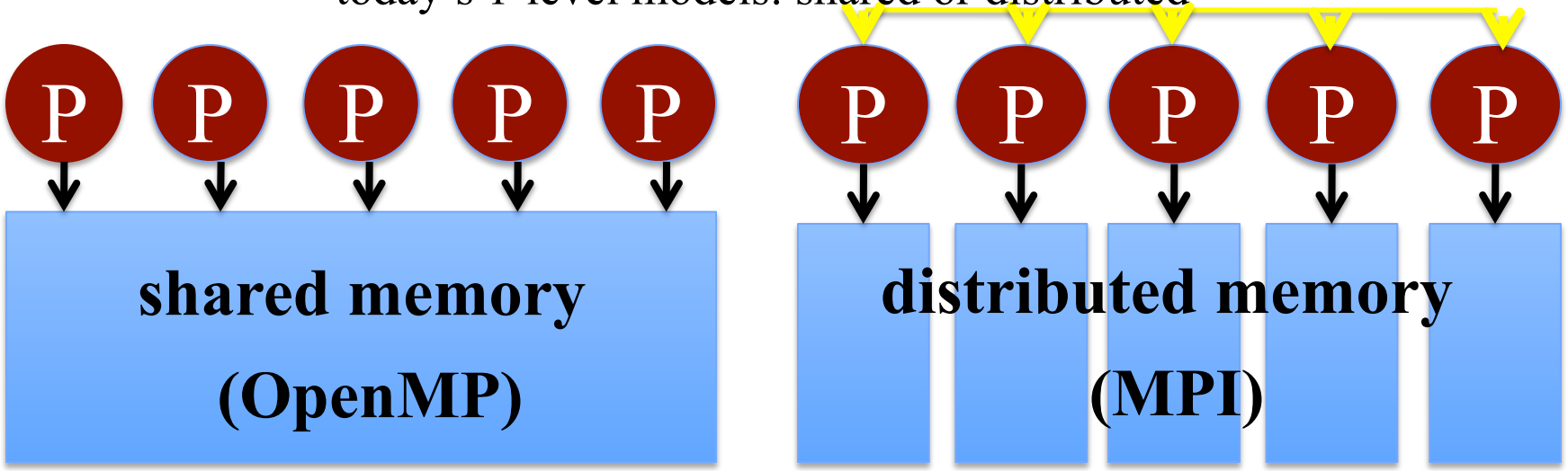
Programming practice

- **Prior to possessing exascale hardware, users can prepare themselves by exploring new programming models**
  - **on manycore and heterogeneous nodes**
- **Attention to locality and reuse is valuable at all scales**
  - **will produce performance paybacks today *and* in the future**
  - **domains of coherence will be variable and hierarchical**
- **New algorithms and data structures can be explored under the assumption that flop/s are cheap and moving data is expensive**
- ***Independent tasks that have complementary resource requirements can be interleaved in time in independently allocated spaces***

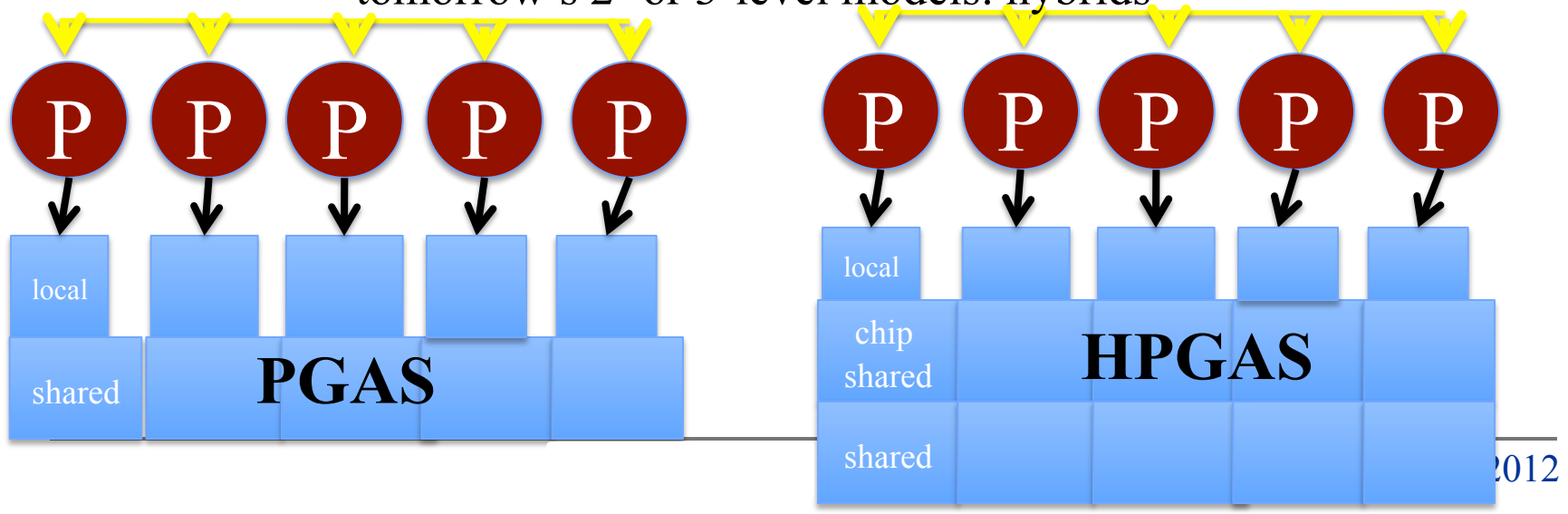
Evolution of parallel programming models:

strong scaling within a node with nonuniform coherency domains

today's 1-level models: shared or distributed



tomorrow's 2- or 3-level models: hybrids



Hybrid programming models *not enough*

- **Tools for monitoring the availability and predicted performance of resources within an architecture-adaptive and application-adaptive are improving**
- **However, even perfect knowledge of resource capabilities at every moment and perfect load balancers will not rescue billion-thread SPMD implementations of PDE simulations, etc.**
  - **cost of rebalancing frequently is too large**
  - **Amdahl penalty of failing to rebalance is fatal**

Peta to exa for algorithms

- **Things we need to do for exascale will help us at petascale and terascale**
  - **Reducing memory requirements and memory traffic**
  - **Exploiting hybrid and less synchronous parallel programming models**
  - **Co-design of hardware and software (for, e.g., power management)**
- **Though it inveighs against the CS aesthetic of "separation of concerns", and involves more issues, co-design requires similar attitude and aptitude as in, say, MPI programming today**
  - **Applications programmers have "bit the bullet" and designed excellent MPI-based codes, by using quality libraries designed and ported by specialists**
  - **Hopefully, we will be able to isolate applications programmers from many of the hardware and software architectural details, just as we do today from message-passing details**

Peta to exa

- **Billion-way parallelism of GigaHertz cores will not significantly expand today's million-way flat parallelism at the node level**
  - **MPI legacy code will still be usable on the "outside" on a million nodes**
  - **Changes will be mainly *within* a node, where we will need to evolve thousand-way parallelism: "MPI+*X*"**
- **Principal challenges from peta to exa are within the node, and the burden is *shared by the marketplace* at all scales of node aggregation**

Path for scaling up applications

- Weak scale applications up to distributed memory limits
  - proportional to number of nodes
- Strong scale applications beyond this
  - proportional to cores per node/memory unit
- Scale the workflow, itself
  - proportional to the number of instances (ensembles)
  - integrated end-to-end simulation
- Co-design process is staged, with any of these types of scaling valuable by themselves
- Big question: does the software for co-design factor? Or is all the inefficiency at the data copies at interfaces between the components after a while?

Algorithmic Priority Research Directions (1)

- **Advanced mathematical methods for scientific understanding in exascale simulations, including *in situ***
  - **uncertainty quantification, intrusive and nonintrusive**
  - **optimization, inverse problems, sensitivity**
  - **analysis and visualization**
  - **validation and verification**

Algorithmic Priority Research Directions (2)

- **Exascale algorithms that expose and exploit multiple levels of parallelism**
  - **communication-reducing algorithms**
  - **synchronization-reducing algorithms**
  - **intensity-enhancing algorithms**
    - **increasing order cuts two ways here – smaller storage and more reuse (good for memory), but also larger workingsets (bad for registers)**
  - **fault resilient algorithms**
- **Algorithmic support for multiphysics, multiscale methods**
  - **relax the overspecified SPMD and BSP paradigms when joining multiple different codes**
  - **analyze stability of coupling**

Algorithmic Priority Research Directions (3)

- **Exascale algorithms for constructing and adapting discrete objects**
  - **these typically deal with unpredictable, dynamic structures and workloads and have few flops to hide**

A “hot topic” at ICERM, January 2012

Upcoming events

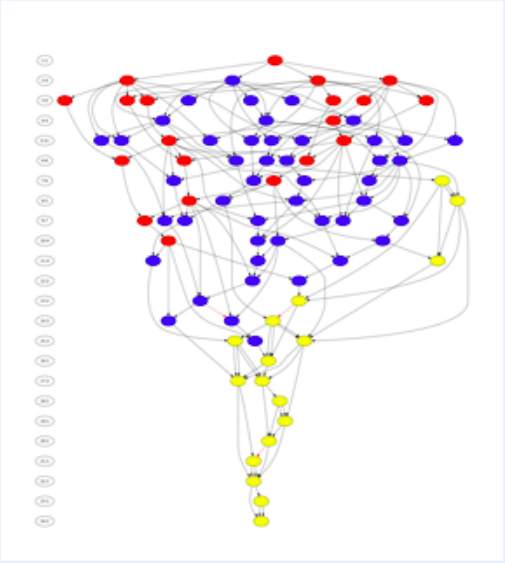
ICERM Topical Workshop on "Synchronization-reducing and Communication-reducing Algorithms and Programming models for Large-scale Simulations" (January 9-13, 2012)

Organizing Committee

- David Keyes (KAUST and Columbia University)
- Matthew Knepley (The University of Chicago)
- Katherine Yelick (University of California at Berkeley and NERSC)

Description

Twin motivations for this interdisciplinary workshop are the necessities of taking scientific simulations beyond their contemporary high-water marks of concurrency and of porting them to execution environments of less scheduling reliability. As concurrency in scientific computing pushes beyond a million threads towards a billion, and as the performance of individual threads becomes less reliable for hardware-related reasons, attention must focus on communication and synchronization bottlenecks in contemporary simulation codes. A fine-grained partial ordering on computational tasks based on the availability of input arguments is imposed by physical causality, but much communication overhead in the form of start-up latency and synchronization delay in popular algorithms isartifactual. Attempts to ameliorate inefficiency due to communication range from increased message aggregation in tightly scheduled algorithms to a fine-grained separation of computational tasks into execution priority ranks, allowing those that are on the critical path to execute whenever their operands are ready, and rebalancing or deferring other tasks for times when tasks on the critical path are waiting, or until the consequences of further deferral are algorithmically detrimental. The evolution of today's simulation codes from the infra-petascale to the ultra-exascale requires importing ideas from other areas of computer science into numerical algorithms, possibly inventing some new ones, and generalizing programming models.



Directed Acyclic Graph for the execution of a symmetric generalized eigensolver for a dense 4x4 matrix [Image courtesy of Hatem Ltaief]

Features important to emerging architectures

- **Exposable concurrency**
- **Storage footprint**
- **Memory locality**
- **Arithmetic intensity**
- **Hierarchical domains and frequency of synchronization**
- **Fusability across interfaces**
- **Opportunity for inspector-executor**
- **Predictability of reference and branch**
- **Controllability of accuracy requirements and FP precision**

Adaptations to emerging architectures

- **Benefits of adaptation to emerging architectures**
  - **Avoidance of hardware limitations**
  - **Mitigation of unavoidable limitations**
- **Modes of adaptation to emerging architectures**
  - **Concentration of locality for communication**
  - **Relaxation of locality for load-balancing**
  - **Aggregation to avoid overhead**
  - **Disaggregation to hide latency**
  - **Redundant or extra work to avoid communication or synchronization**
  - **Catching and tolerating errors in user space**
  - **Applying machine learning to algorithmic optimization**

Desiderata for software developers

- **The usual**
  - **Abstraction of algorithms into libraries**
  - **Criticality of open-source libraries for progress**
  - **Proven practices of scientific software engineering for library promote and curation**
- **The progressive**
  - **Knowledge about what the new hardware is doing**
  - **Controls over what the new hardware is doing**
  - **Performance models that are "good enough" to inform decisions**
  - **More levels of abstraction in representing ideas and coding for multiple architectures**

Required software enabling technologies

Model-related

- Geometric modelers
- Meshers
- Discretizers
- Partitioners
- Solvers / integrators
- Adaptivity systems
- Random no. generators
- Subgridscale physics
- Uncertainty quantification
- Dynamic load balancing
- Graphs and combinatorial algs.
- Compression

Development-related

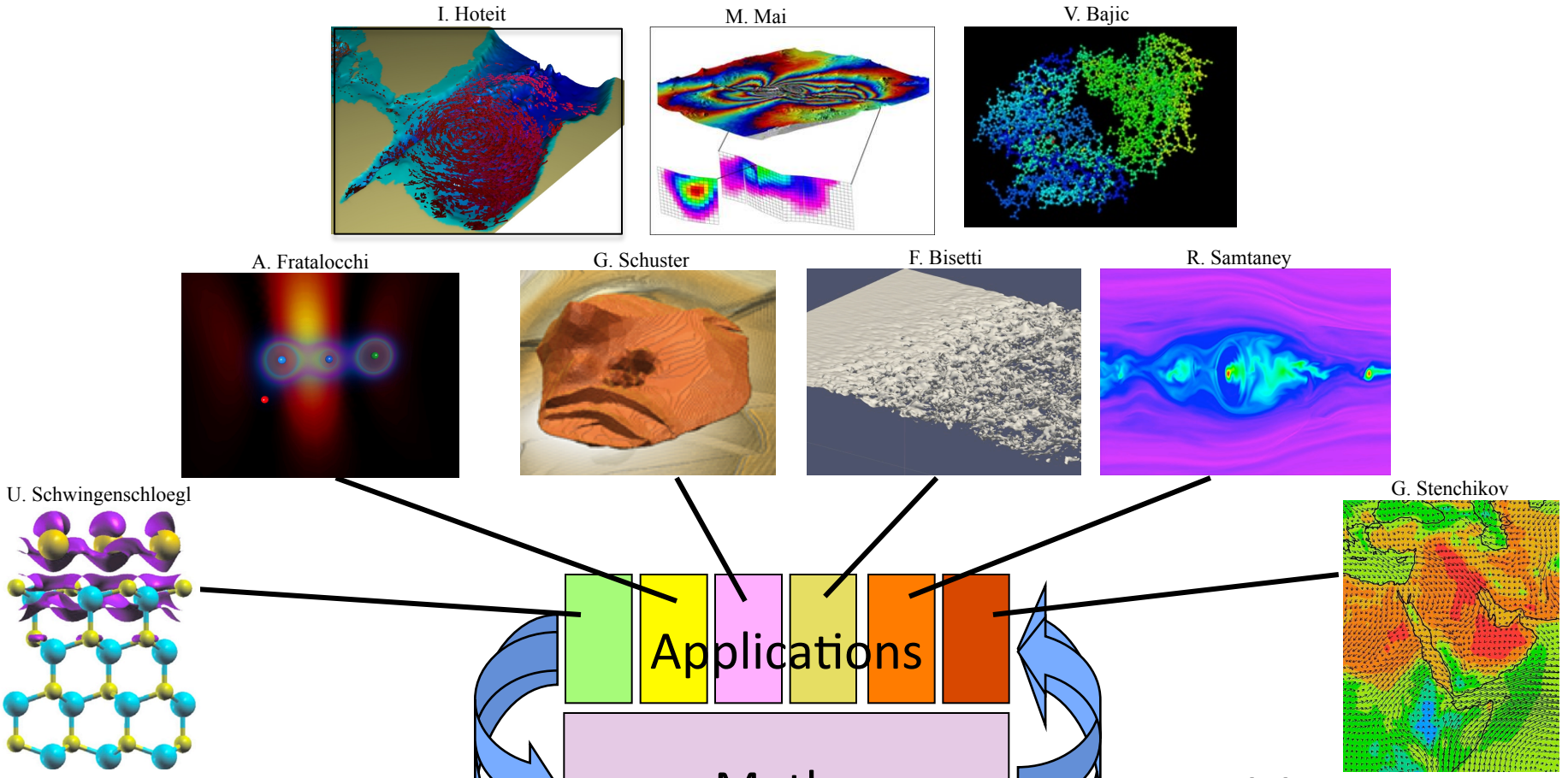
- Configuration systems
- Source-to-source translators
- Compilers
- Simulators
- Messaging systems
- Debuggers
- Profilers

High-end computers come with little of this stuff. Most has to be contributed by the user community

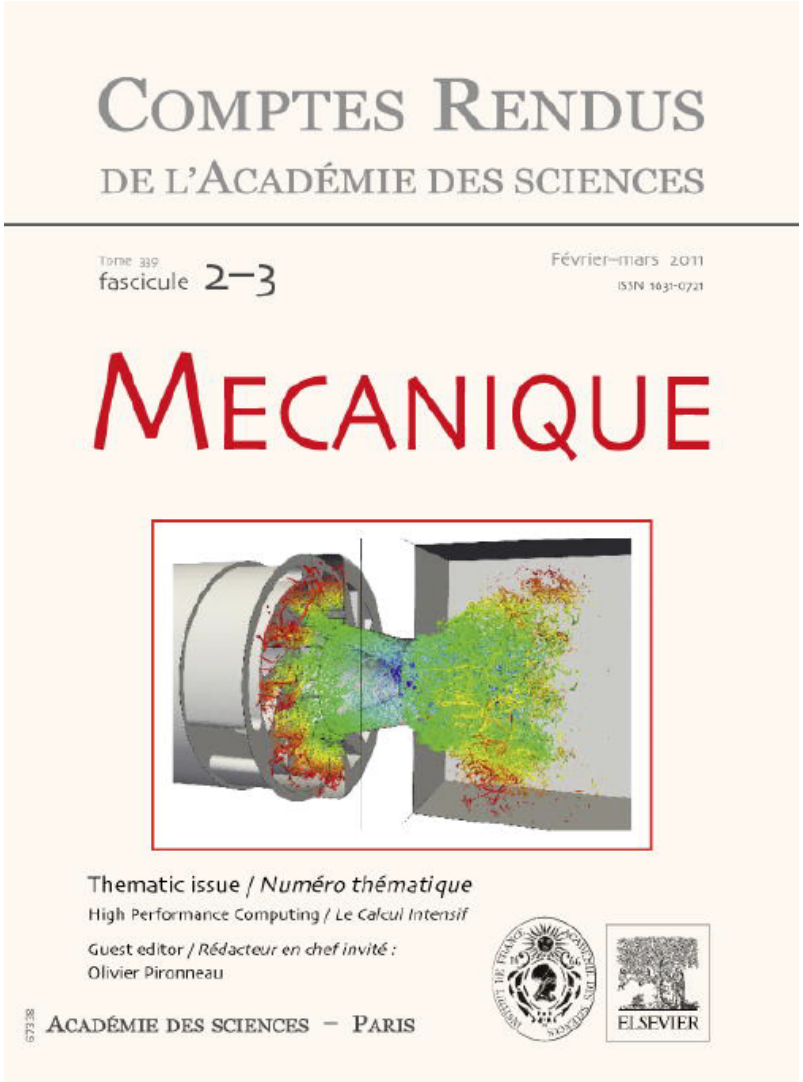
Production-related

- Dynamic resource management
- Dynamic performance optimization
- Authenticators
- I/O systems
- Visualization systems
- Workflow controllers
- Frameworks
- Data miners
- Fault monitoring, reporting, and recovery

"SciDAC philosophy" of software investment



See 2011 special issue of *Comptes Rendus*

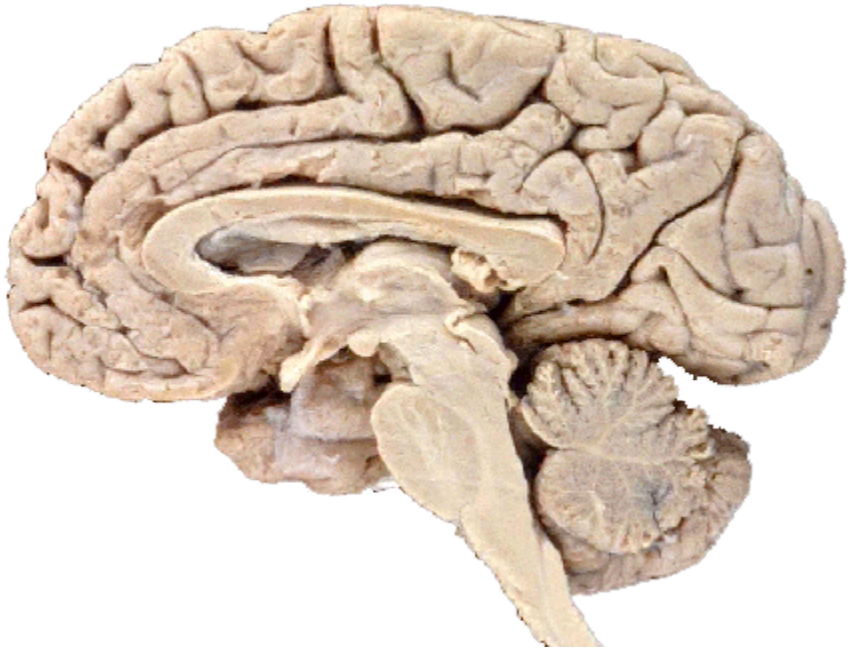


Exaflop/s: The why and the how, D. E. Keyes, *Comptes Rendus de l'Academie des Sciences* **339**, 2011, 70—77.

Kennedy's Challenge, 1962

"We choose to do [these] things, not because they are easy, but because they are hard, *because that goal will serve to organize and measure the best of our energies and skills*, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one which we intend to win..."

Acknowledgment: today's Peta-op/s machines



10^{12} neurons @ 1 KHz = 1 PetaOp/s

1.4 kilograms, 20 Watts


SAUDI ARABIAN