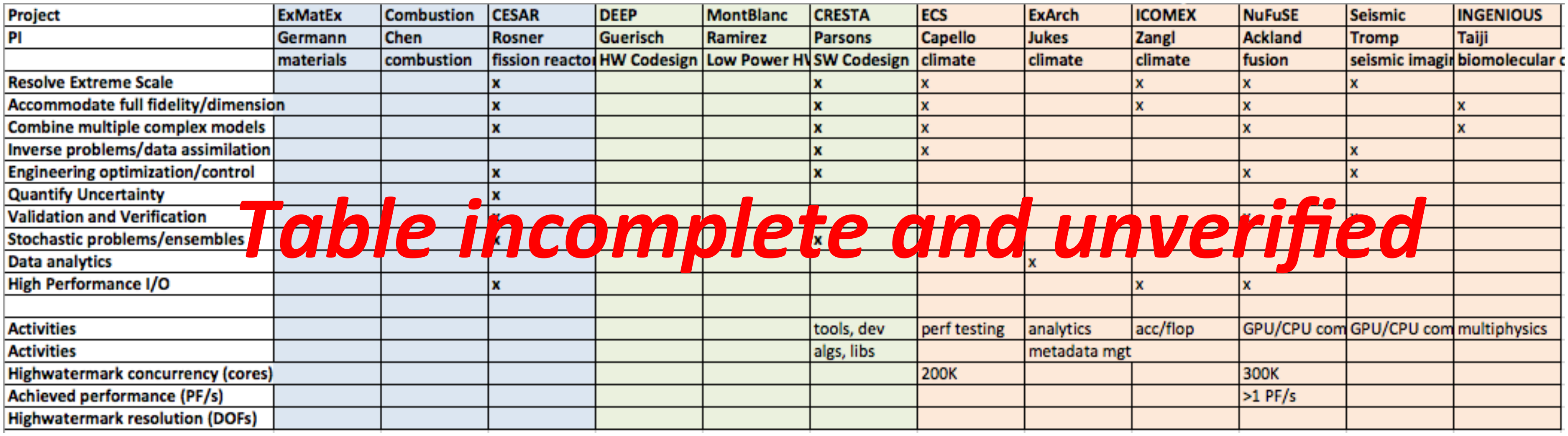
Where are apps *en route* to exascale?

IESP #8
Kobe, Japan

How will the G-8 and other operating and planned co-design projects explore new architectural space?

- **3 US DOE co-design projects**
- **3 EU co-design projects**
- **6 G-8 exascale projects**
- **to be joined by Chinese and Japanese co-design projects**



Project	ExMatEx	Combustion	CESAR	DEEP	MontBlanc	CRESTA	ECS	ExArch	ICOMEX	NuFuSE	Seismic	INGENIOUS
PI	Germann	Chen	Rosner	Guerisch	Ramirez	Parsons	Capello	Jukes	Zangl	Ackland	Tromp	Taiji
	materials	combustion	fission reactor	HW Codesign	Low Power HV	SW Codesign	climate	climate	climate	fusion	seismic imagir	biomolecular c
Resolve Extreme Scale			x			x	x		x	x	x	
Accommodate full fidelity/dimension			x			x	x		x	x		x
Combine multiple complex models			x			x	x			x		x
Inverse problems/data assimilation						x	x				x	
Engineering optimization/control			x			x				x	x	
Quantify Uncertainty			x									
Validation and Verification			x							x	x	
Stochastic problems/ensembles			x			x						
Data analytics								x				
High Performance I/O			x						x	x		
Activities						tools, dev	perf testing	analytics	acc/flop	GPU/CPU com	GPU/CPU com	multiphysics
Activities						algs, libs		metadata mgt				
Highwatermark concurrency (cores)							200K			300K		
Achieved performance (PF/s)										>1 PF/s		
Highwatermark resolution (DOFs)												

Table incomplete and unverified

Exascale payoffs for apps

- **Resolve extreme scales**
- **Accommodate high fidelity physics in full dimensions**
- **Combine multiple complex models**
- **Solve an inverse problem, or perform data assimilation**
- **Perform optimization or control**
- **Quantify uncertainty**
- **Validate and verify**
- **Solve stochastic problems**
- **Analyze data**
- **Combine 3rd and 4th paradigms**

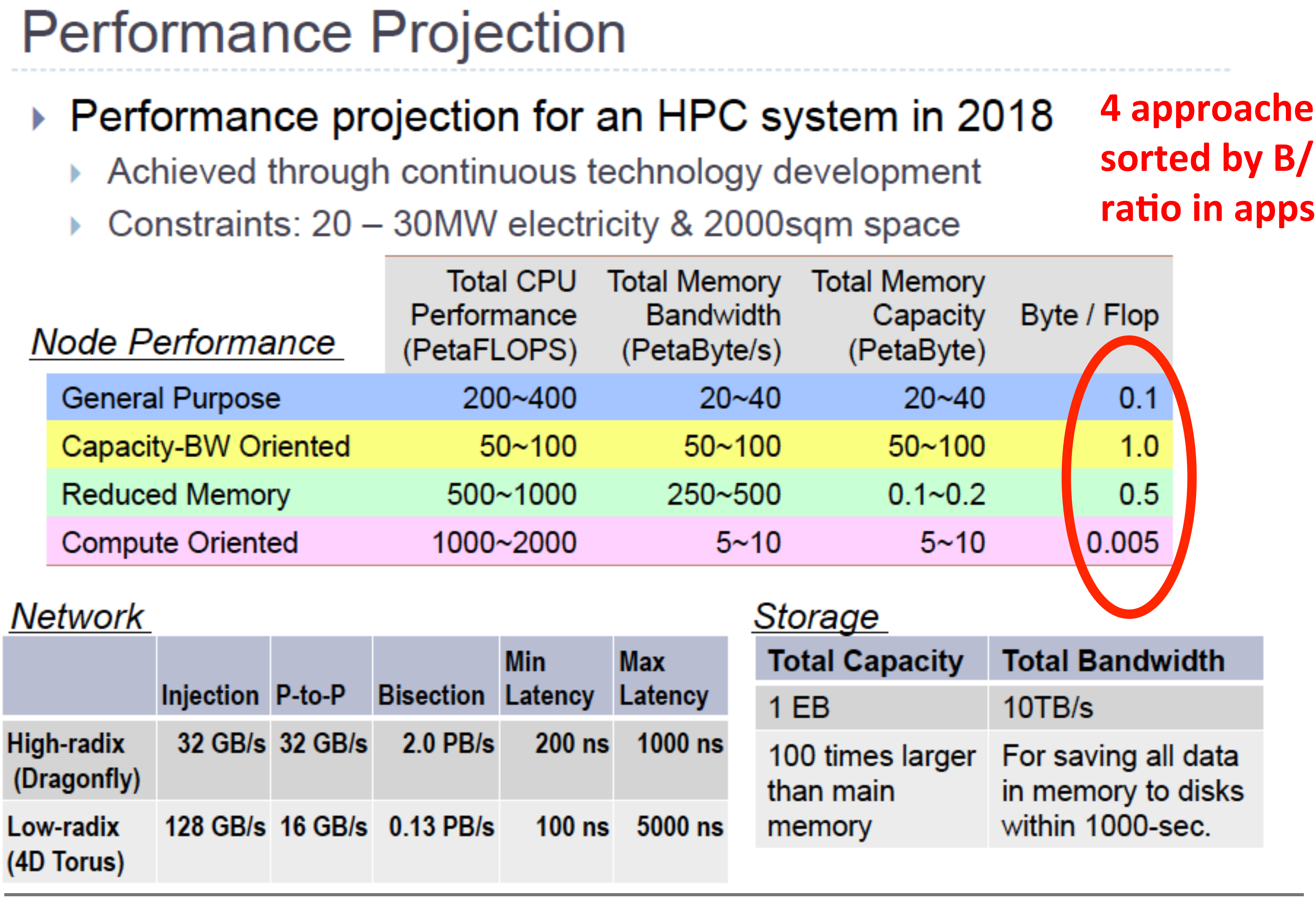
Current co-design and G-8 projects

- Primarily (not exclusively) apps using traditional formulations (e.g., PDEs) and exploring what is possible with emerging HW/SW environments (e.g., GPUs, MPI+X)
- Primarily apps achieving 1 PF/s for the first time
- Encouraging scientific progress from improved resolution, fidelity, and relaxation of assumptions
    - e.g., trillion mesh points or particles, full finite-rate chemical kinetics, fully coupled etc.
- Interesting synergies across the portfolio of disciplines
- Interesting feedback on architectural proportions

Where are apps *en route* to exascale?

- **Perspective: in 1996, Thomas started the Petaflops series of meetings (apps, archs, algs, etc.)**
  - **Hardware and programming model were pitched between COTS and revolutionary (e.g., HTMT)**
  - **In 2008, we achieved 1 PF/s without changing the programming model (bulk synchronous SPMD)**
- **Subsequently, in 2008, Jack *et al.* started this IESP series of meetings**
  - **Hardware "swimlanes" vary around the "Meganode/ Kilocore/GigaHertz" default, with various accelerators**
  - **By 2020, will we achieve 1 EF/s without changing the programming model?**
  - **Probably not! (Need to follow where solvers are leading – more shortly)**

What feedback can apps provide to archs?



Performance Projection

- Performance projection for an HPC system in 2018
  - Achieved through continuous technology development
  - Constraints: 20 – 30MW electricity & 2000sqm space

4 approaches sorted by B/F ratio in apps

Node Performance

	Total CPU Performance (PetaFLOPS)	Total Memory Bandwidth (PetaByte/s)	Total Memory Capacity (PetaByte)	Byte / Flop
General Purpose	200~400	20~40	20~40	0.1
Capacity-BW Oriented	50~100	50~100	50~100	1.0
Reduced Memory	500~1000	250~500	0.1~0.2	0.5
Compute Oriented	1000~2000	5~10	5~10	0.005

Network

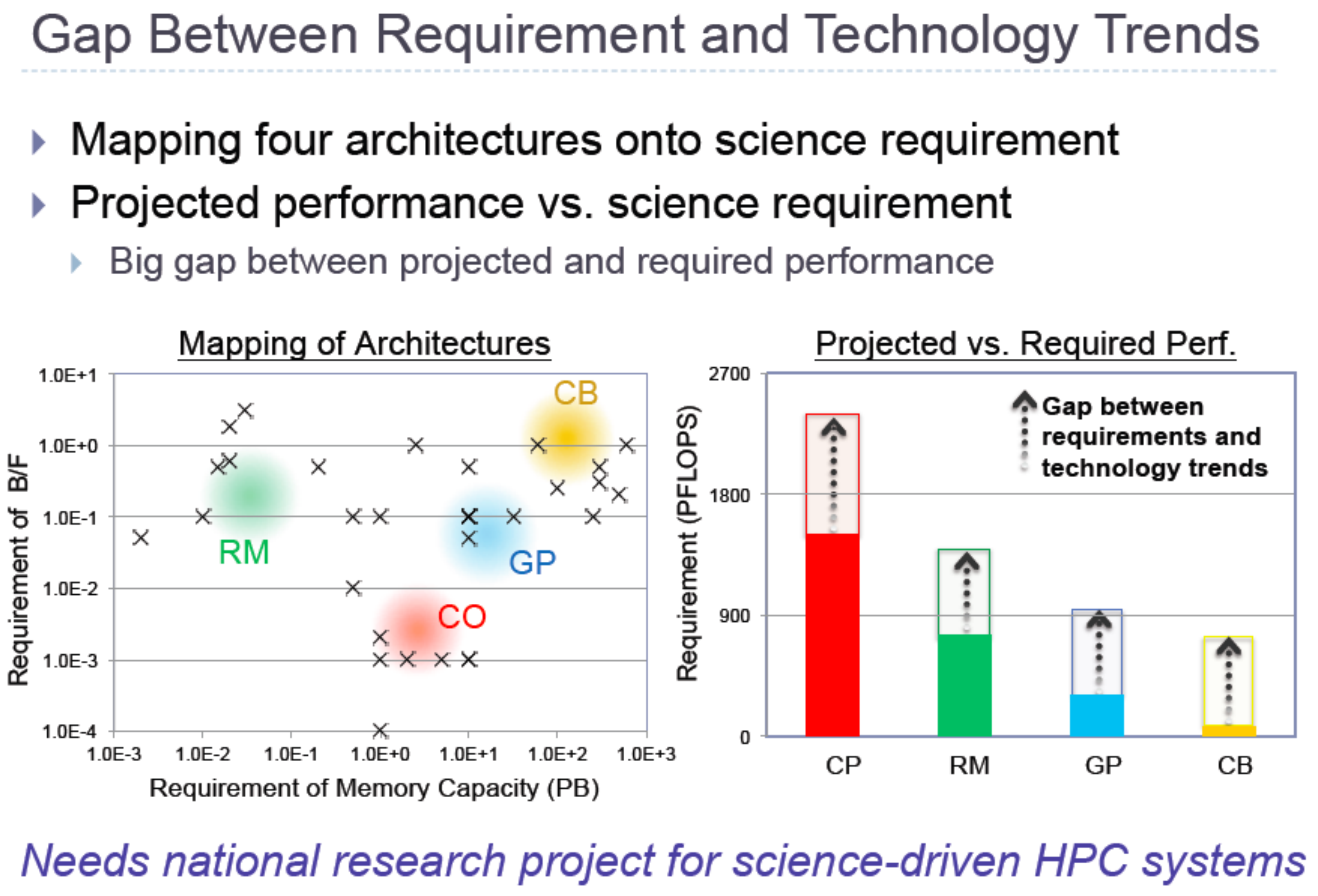
	Injection	P-to-P	Bisection	Min Latency	Max Latency
High-radix (Dragonfly)	32 GB/s	32 GB/s	2.0 PB/s	200 ns	1000 ns
Low-radix (4D Torus)	128 GB/s	16 GB/s	0.13 PB/s	100 ns	5000 ns

Storage

Total Capacity	Total Bandwidth
1 EB	10TB/s
100 times larger than main memory	For saving all data in memory to disks within 1000-sec.

c/o M. Kondo (IESP-Kobe)

What feedback can apps provide to archs?



Gap Between Requirement and Technology Trends

- Mapping four architectures onto science requirement
- Projected performance vs. science requirement
  - Big gap between projected and required performance

Mapping of Architectures

Projected vs. Required Perf.

Needs national research project for science-driven HPC systems

c/o M. Kondo (IESP-Kobe)

Will the field support this many archs?

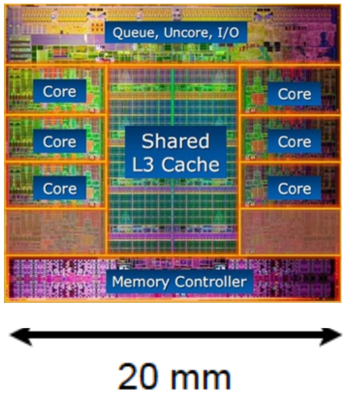
- Remains to be seen how heterogeneous the architectural space will be
- Ratios of bytes transferred to flops executed is an interesting metric by which to evaluate demand, but does not by itself address scheduling of the respective operations
- Little said so far at this meeting about programming models to accommodate exascale architectural stresses
- Little promised in current co-design and exascale projects to address new algorithms and new formulations

Reminder: 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
```



(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

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 or clocks speeds varied based on compute schedules and based on cooling capacity
- Non-reproducible memory hierarchy policies under sharing will cause node performance to vary
- Result: per-core and per-node performance rates will be unreliable complicating load balance and synchrony

“Physics will converge our programming models” (Lamb)

- **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**

Apps programming model convergence

- **Billions of dollars of scientific software hang in the balance while better formulations and algorithms evolve to span the architectural gap**
- **Historically, in the 40 apps surveyed by Beckman for the San Francisco IESP meeting, explicit message passing is effectively hidden from most developers by well developed libraries**
- **Similarly, it will eventually be easier to program hybrid, heterogeneous architectures than to do user-managed data placement**
- **This is asymptotic… in the meantime…**

Applications Agenda

- **New hardware-tolerant 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* enhancing applications to obtain all of the exciting payoffs that exascale is meant to exploit**

A community working model

- Mathematical and modeling ideas should be welcomed from many sources, including nontraditional
- It is important to show pay-offs on skeleton applications before these ideas *can*, *should*, or *will* be adopted
- Applications communities have detailed roadmaps between now and 2020 apart from adapting to emerging architectures
- Expertise from a small number of progressive computational mathematicians and computer scientists will have to be packaged for wider adoption, since not every applications community can find or afford all the computational experts it otherwise needs

Algorithmic Paths Forward

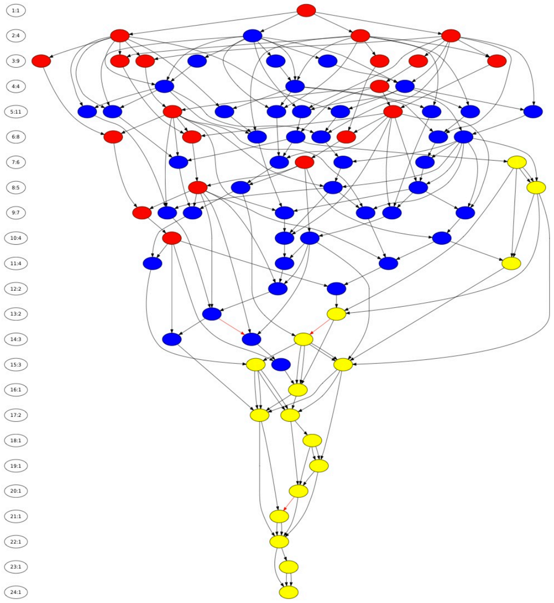
- **Requirements (?) of algorithm and software developers**
  - New depth of knowledge about what the hardware is doing
  - New exercise of control over what the hardware is doing
  - Performance models that are "good enough" to inform decisions
  - More levels of abstraction in representing ideas and coding for multiple architectures
  - Better understanding of numerical error propagation to exploit reduced precision
- **Modes of adaptation to emerging architectures (incl. some opposing tendencies)**
  - Concentrate locality for efficient access *vs.* relax locality for load-balancing flexibility
  - Aggregate to reduce overhead *vs.* disaggregate to hide latency
  - redundant or extra work to avoid communication or synchronization
  - Catch and conditionally tolerate errors in user space
  - Checkpoint in user space
  - Apply machine learning to optimize ordering and layout

Algorithmic Paths Forward

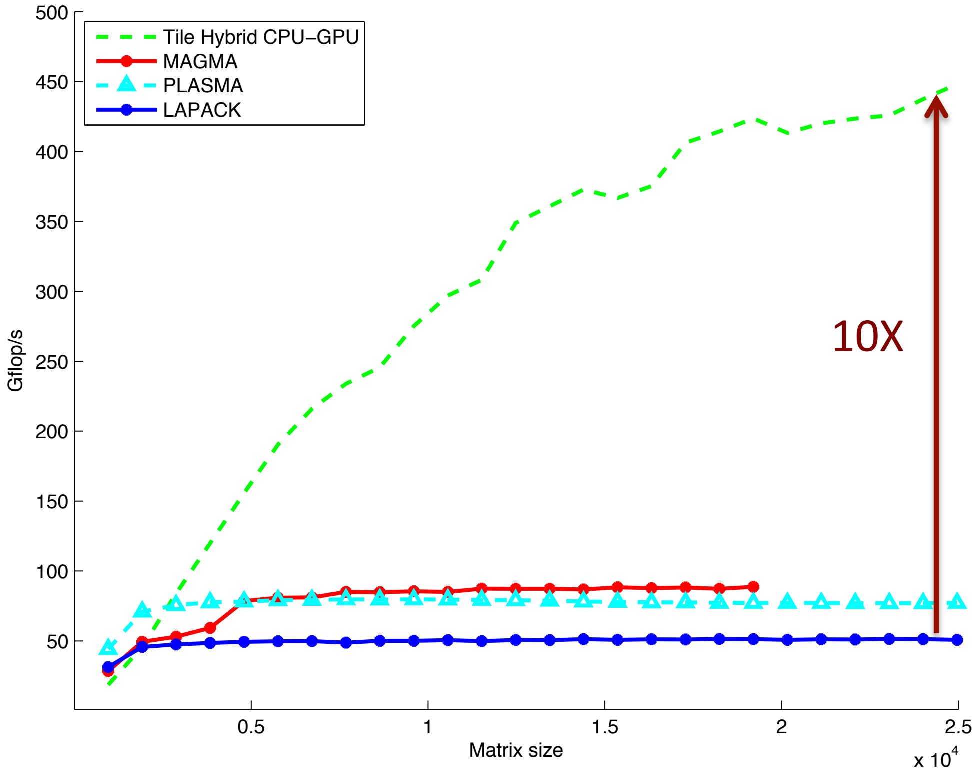
- **Solvers, those traditional bottlenecks, can lead the way, just as they did onto shared memory and distributed memory parallelism**
- **Next two slides mention two traditional types of solvers that have been "liberated" to tolerate less rigid scheduling**
- **Each generates a sequence of small tasks that can be scheduled either in the body of loops arranged sequentially within subroutines *or* at the nodes of a directed acyclic graph (DAG)**

Example of DAG-based linear solver

High Performance Cholesky-based Matrix Inversion
on Multicore with Hardware Accelerators



Dynamic Runtime System



c/o H. Ltaief *et al.* HiPC’11 (Bangalore)

Example asynchronous nonlinear solver

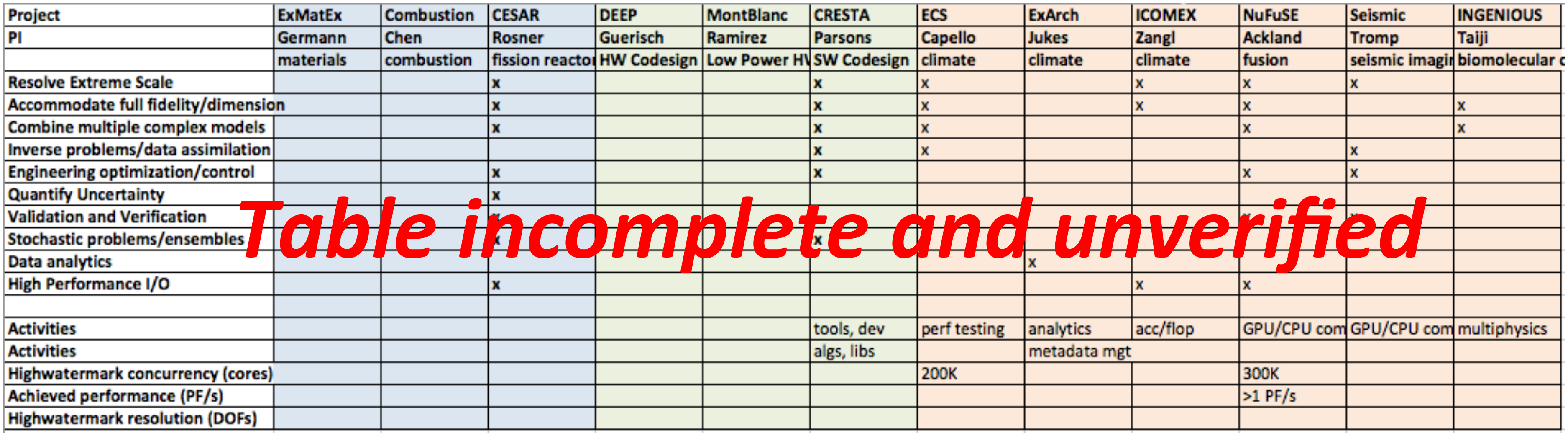
- **Nonlinear Schwarz replaces a Newton method for a global nonlinear system …**
  - **which computes a global distributed Jacobian matrix and synchronizes globally in both the Newton step and in solving the global linear system for the Newton**
- **With a set of local problems on subsets of the global nonlinear system**
  - **Each local nonlinear problem has only local synchronization**
  - **All of the linear systems for local Newton updates have only local synchronization**
  - **There is still global synchronization in a number of steps hopefully much fewer than required in the original Newton method**
- **What was a "feature" is now a major advantage**

Important frontier, under-explored

- **Characterizing the 4th paradigm requirements for extreme computing as well as many 3rd paradigm requirements**
- **For the 3rd paradigm**
  - **“optimal” complexity (work scales linearly or log-linearly in data size)**
  - **“optimal” weak scaling methods (up to a point)**
- **What are the complexities for typical 4th paradigm problems? What, even, are the parameters?**
- **Combination of 3rd and 4th paradigm in data assimilation is better characterized than 4th alone**
  - **Since simulation tends to dominate in existing examples**

Co-design: it's time to inventory and evaluate

- **3 US DOE co-design projects**
- **3 EU co-design projects**
- **6 G-8 exascale projects**
- **to be joined by Chinese and Japanese co-design projects**



Project	ExMatEx	Combustion	CESAR	DEEP	MontBlanc	CRESTA	ECS	ExArch	ICOMEX	NuFuSE	Seismic	INGENIOUS
PI	Germann	Chen	Rosner	Guerisch	Ramirez	Parsons	Capello	Jukes	Zangl	Ackland	Tromp	Taiji
	materials	combustion	fission reacto	HW Codesign	Low Power H\	SW Codesign	climate	climate	climate	fusion	seismic imagi	biomolecular c
Resolve Extreme Scale			x			x	x		x	x	x	
Accommodate full fidelity/dimension			x			x	x		x	x		x
Combine multiple complex models			x			x	x			x		x
Inverse problems/data assimilation						x	x				x	
Engineering optimization/control			x			x				x	x	
Quantify Uncertainty			x									
Validation and Verification			x							x	x	
Stochastic problems/ensembles			x			x						
Data analytics								x				
High Performance I/O			x						x	x		
Activities						tools, dev	perf testing	analytics	acc/flop	GPU/CPU com	GPU/CPU com	multiphysics
Activities						algs, libs		metadata mgt				
Highwatermark concurrency (cores)							200K			300K		
Achieved performance (PF/s)										>1 PF/s		
Highwatermark resolution (DOFs)												

Table incomplete and unverified

Co-design program strategies

- **Application domains may be quite different based upon the number of different issues to be addressed (*e.g.* climate *vs.* chemistry)**
- **Vendors are interested in a common interface with CDC's (at least initially)**
- **Is there a «best» way to proceed? Series of specific (national, international) (application, transverse) CDC's, relationships between projects, others, … ?**
- **Need to start with a (small) number of initiatives to demonstrate feasibility and management practices**
- **Whatever the organization, critical need for exchanging expertise between domains**
- **Evaluate the existing CDC's (EESI-2)**