Joint Laboratory for Petascale Computation

INRIA NCSA

Panel: Overcoming the fear of the unknown

Franck Cappello

INRIA and UIUC

CCGSC Ashville 2010

Is there a fear of Exascale?

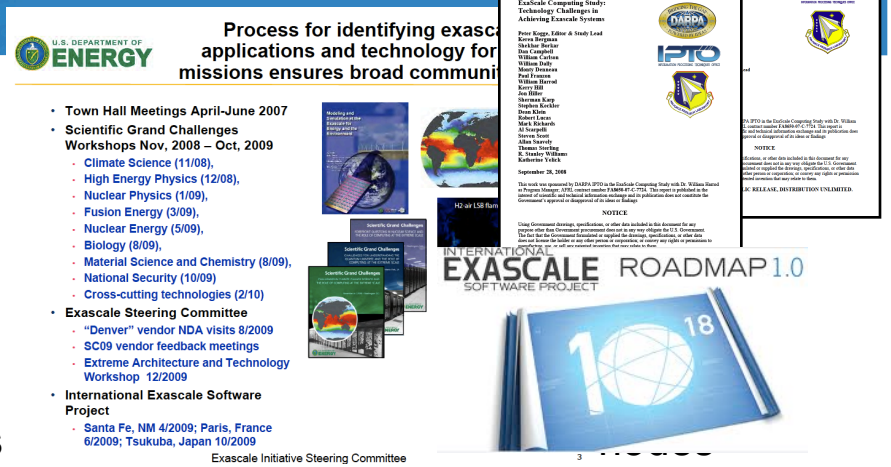
First: some have no fear

"Right now there is nothing we see that would prevent us from getting to Exascale — in just the sense of a system. The problems identified are all solvable. … Sure, we face some pretty big challenges like power and memory issues — but they are all solvable."

In « Exascale: The Beginning of the Great HPC Disruption », Exascale Report, August 2010

Some reasons to be less optimistic
Before looking to
“How can we overcome the fear?"

Main source of fear



Distance between projections and the current situation + remaining time (2018-2020)

1) Hardware

1)1B cores: 10K core/node * 1Ghz* 100K nodes

2)Less memory per core (<<1GB per core) because of the cost ½ of $200M

3)Memory bandwidth is critical (highest energy cost is movement of data offchip):

→200W 10TFfops/chip OK but 0.02 byte/flop/s (BW → 0.5, top machines → 0.1 to 0.3)

4)Network power consumption is critical:

→ topology choice based on power (mesh topologies have power advantages)

5)More soft errors: need more ECC, Parity, etc. But contradictory with power limits?

6)O(1day) ≥ MTBF ≥ O(1h)

2) System software

-programming/runtime env., debugging tools, fault tolerance support, etc.

3) Applications

•What applications will be able to exploit even 10% of peak Exascale perf. with

+Strong Scaling (lower memory per core)

+Mesh topology (fat tree with full bisection bandwidth too expensive)

+0.02 Bytes / Flop (0.2 if we are lucky, from Exascale study reports)

+Application MTTI of 1 hour (pessimistic assumption)

Another source of fear

Some key factors seem not explored

Energy is a major concern but Resilience (fault tolerance) too.

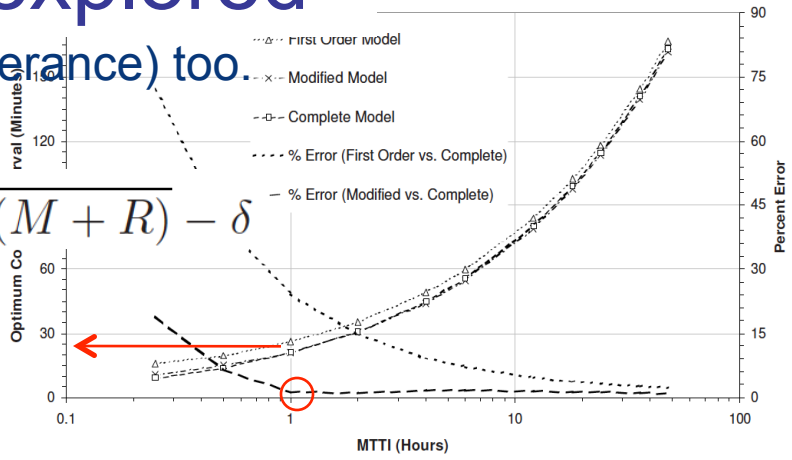
How they interrelate → ECC+parity AND Ckpt/Rst

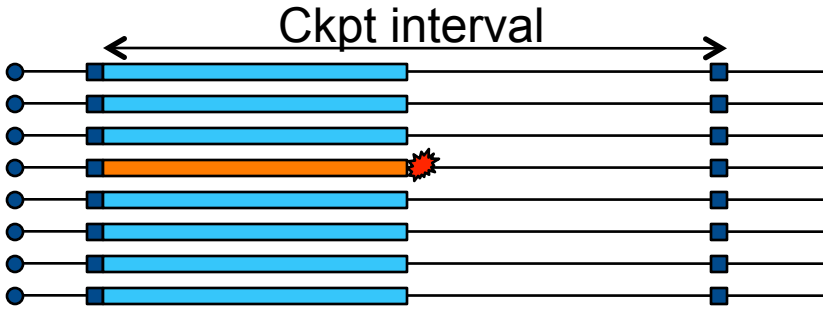
Simple calculation of the ckpt interval (Dally's modified model from Young)

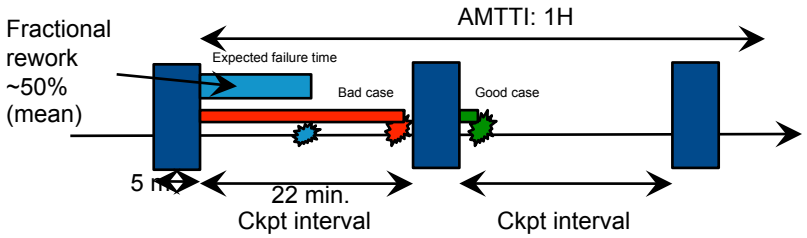
δ =5 minutes, R=10 minutes, AMTTI=1h.

Ckpt interval ~ 22 minutes.

$$\tau = \sqrt{2\delta(M+R)} - \delta$$







If we keep global restart, 11+10 → 21 minutes over 1h, will be for rework

And 10 minutes will be used for checkpointing…

→ ½ of the 20MW (or more) just for fault tolerance, even if 1 process fails!

→ With AMTTI=10h, δ =R=30m, 1/3 of 20MW (or more) for fault tolerance.

→ Explore partial restart and methods to avoid restart when possible

Yet another source of fear

Lack of information - cooperation

For Fault tolerance, we need more precisions on:

- How many non detected soft errors per day (hours)?
- How well vendors intent to improve errors detection and correction for soft errors? more protection mechanisms → more power consumption

  → how much this is in contradiction with reducing the power consumption?
- How errors and faults propagate and turn to failures (fault propagation graphs are mostly unknown or undocumented, in particular for soft errors)?
- Root causes (a cause is considered as a root cause if by fixing it, it prevents the failure to happen again, then real root cause may manifest by another path)
- What is the proportion of software errors/hardware errors (they should be managed differently, but existing analysis diverge in their conclusions)
- Many event monitoring and logging systems filter « normal events », so event analysis only has part of the story. Characterizing « normal behavior » may help detecting deviations before errors are reported
- Etc.

→ More meetings with hardware developers are needed

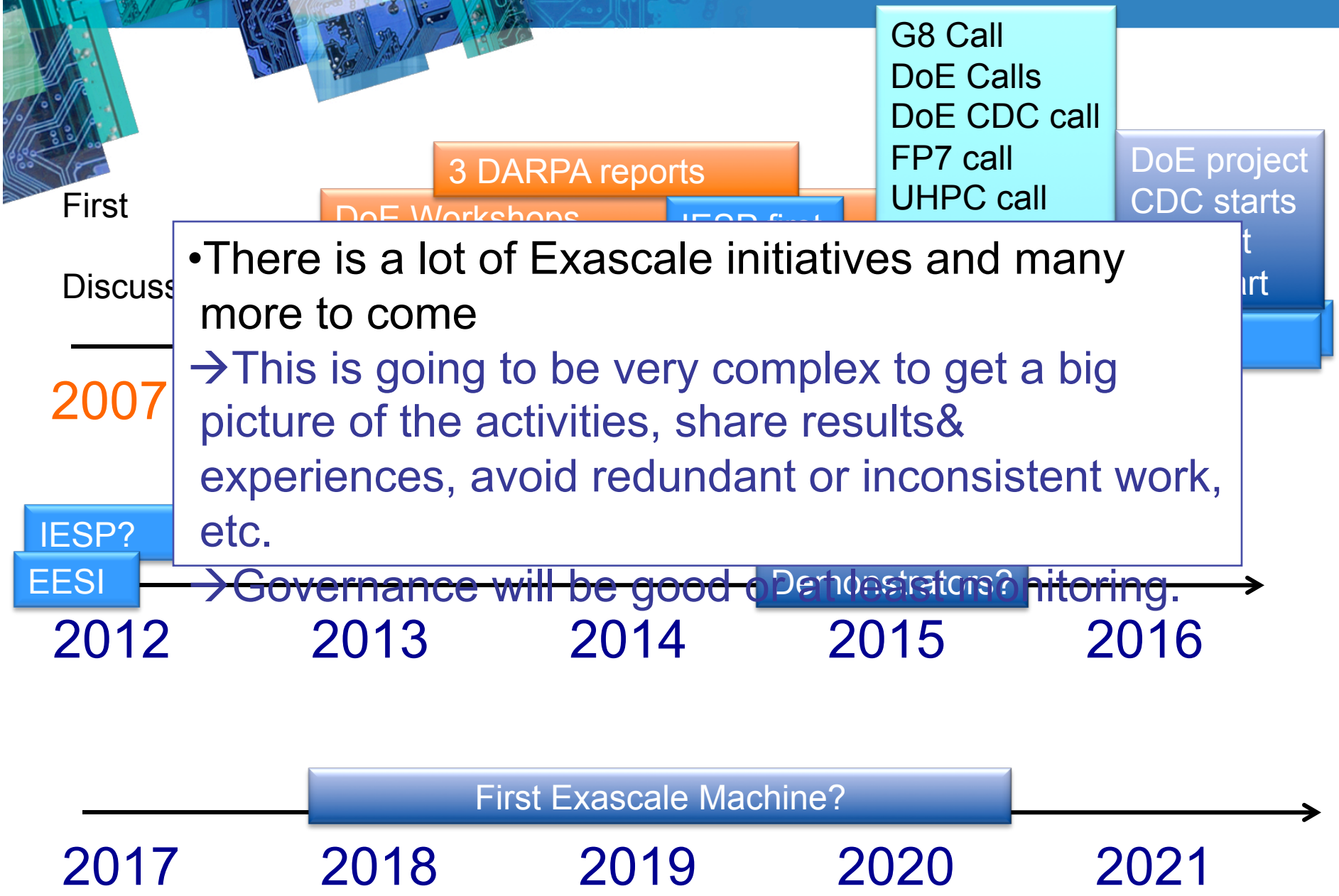
→ We desperately need access to recent error/event log files

How can we overcome the fear?

(Reasons to be less pessimistic)

The community is developing plans and research at different levels

Level 1: A lot of research initiatives



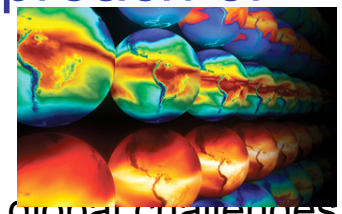
- There is a lot of Exascale initiatives and many more to come

→This is going to be very complex to get a big picture of the activities, share results& experiences, avoid redundant or inconsistent work, etc.

→Governance will be good or at least monitoring.

Level2: Making the problem less complex

1) Several initiatives are following a co-design approach or follow application driven CS research



G8

- Collaborations between experts in research areas related to global challenges and developers of future exascale platforms

DoE Exascale co-design centers (July 2010)

- A center = a PI + 1 application + Strong ties to DoE + Interdisciplinary team
- Combine expertise of vendors, hardware architects, system software developers, domain scientists, computer scientists, and applied mathematicians.

DARPA UHPC (Ubiquitous High Performance Computing)

- Goals: new system-wide technology approaches specifically including hardware and software co-design

EESI (European Exascale Software Initiative)

- Application driven (Grand-Challenges)

2) Several initiatives consider Ensemble computations

Ex: Climate simulations

- Large ensembles of 1000's of configurations will be needed to resolve the effects of natural variability in the climate system and quantify uncertainty

Level3: Promising technical solutions to complex issues are emerging

Joint Laboratory for Petascale Computation

INRIA NCSA

Example: addressing the global restart problem

Coordinated checkpoint → global restart
Partial restart → message logging
Problem with message logging: high overhead + memory occupation

Proposed solution → partial restart:

- Establish virtual clusters on the system
- Coordinated checkpointing inside clusters
- Message logging between clusters

→Not enough: cluster should replay deterministically
→Either application is deterministic OR need to force deter.

Applications	Communication determinism	Applications	Communication determinism
ScaLAPACK SUMMA	Deterministic	NAS SP	Send-deterministic
NAS LU	Deterministic	NAS CG	Deterministic
NAS MG	Deterministic	NAS BT	Send-Deterministic
NAS FT	Deterministic	NAS EP	Deterministic
NAS DT	Deterministic	Nbody	Deterministic
USQCD CPS++ (MPI)	Send-Deterministic	USQCD MILK	Send-Deterministic
NERSC-6 CAM	Send-Deterministic	NERSC-6 GTC	Deterministic
NERSC-6 IMPACT	Send-Deterministic	NERSC-6 MAESTRO	Send-Deterministic
NERSC-6 PARATEC	Deterministic	SpecFEM3D	Deterministic
Jacobby	Deterministic	SQ sphot	Send-deterministic
SQ UMT	Send-Deterministic	SQ IOR	Deterministic
SQ IRS	Send-Deterministic	SQ lammps	Send-Deterministic
Ray2Mesh MW	Non-Deterministic	Ray2Mesh QD	Send-Deterministic
Master Worker	Non-Deterministic	Divide and Conquer	Send-Deterministic [1]

Table 2: Communication determinism in parallel applications

→Study of communication determinism in HPC apps [Cappello, Guermouche, Snir 2010] → Most HPC applications have deterministic or send-deterministic Coms. Pat.
→No need to force communication determinism
→ Proposed solution is promising

Other promising technical solutions: Performance Modeling of Architectures and Applications, Dedicated core(s) for coms, I/O and other OS services, communication avoiding algorithms, etc.

Some remarks

How do we move past qualitative statements to quantitative predictions (and real progresses)?

- Some issues have been raised repetitively, but there is no or little progress on them: access to event/errors log, file syst. logs., batch scheduler logs, etc.

→ We need to address this problem for better predictions and better tools for FT mechanisms, scheduling, power management, performance modeling, etc.

- Some of us "assumes" that the software community will develop part of the Exascale system software stack

BUT most of the community does not have contact with hardware developers

→We need to establish more link between HW and SW developers

Some consequences

Energy is a major concern but Resilience (fault tolerance) too.

How they interrelate → ECC+parity AND Ckpt/Rst

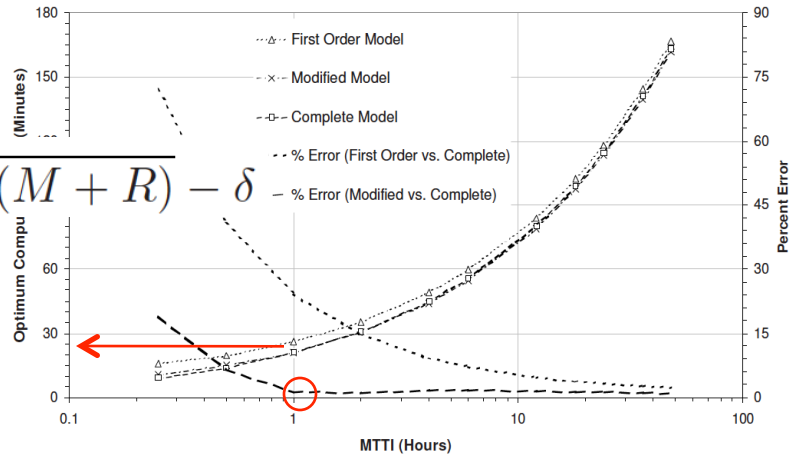
Here is an example about Chpt/Rst (efficiency):

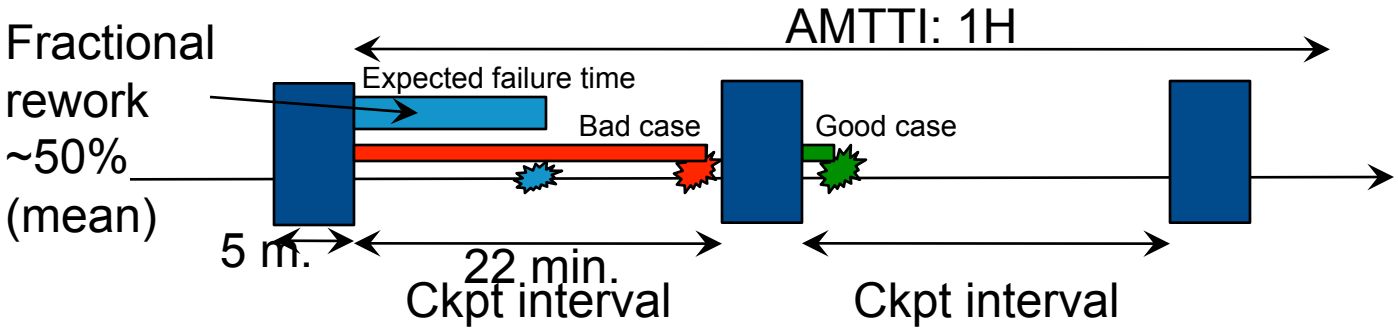
Simple calculation of the ckpt interval (Dally's modified model from Young)

δ =5 minutes, R=10 minutes, AMTTI=1h.

Ckpt interval ~ 22 minutes.

$$\tau = \sqrt{2\delta(M+R)} - \delta$$





If we keep global restart, 11+10 → 21 minutes over 1h, will be for rework

And 10 minutes will be used for checkpointing…

→ ½ of the 20MW (or more) just for fault tolerance, even if 1 process fails!

→ With AMTTI=10h, δ =R=30m, 1/3 of 20MW (or more) for fault tolerance.