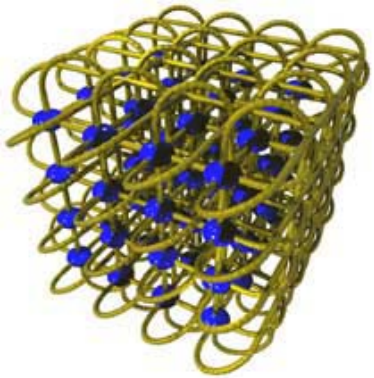
Februrary, 2003



Diskless Checkpointing on Super-scale Architectures

Applied to the Fast Fourier Transform

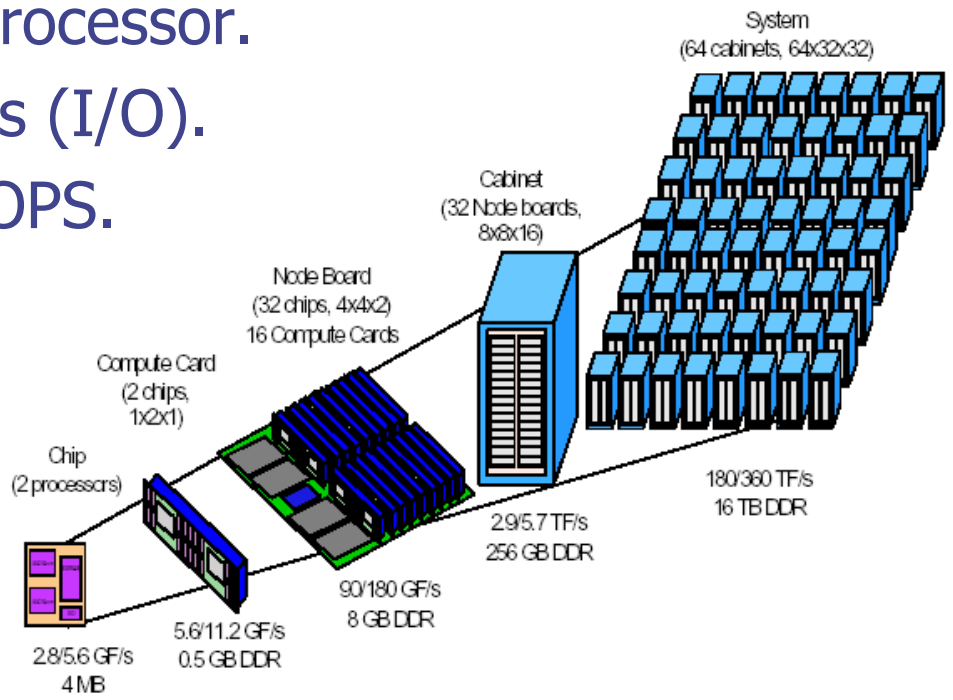
Christian Engelmann, Al Geist
Oak Ridge National Laboratory

Super-scale Architectures

- Current tera-scale supercomputers have up to 10,000 processors.
- Next generation peta-scale systems will have 100,000 processors and more.
- Such machines may easily scale up to 1,000,000 processors in the next decade.
- IBM currently builds the BlueGene\L at Lawrence Livermore National Laboratory.

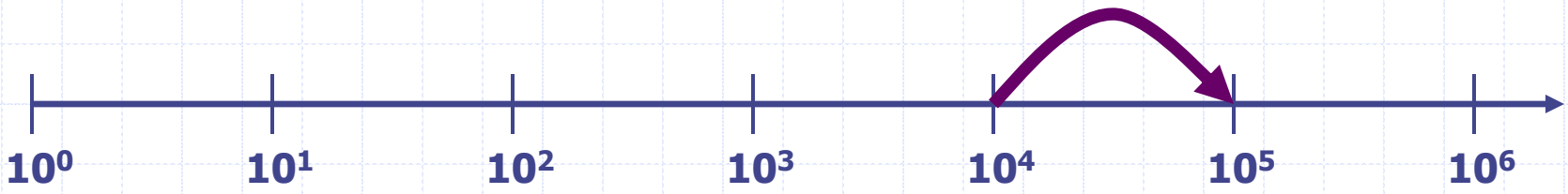
IBM BlueGene\L at LLNL

- Up to 64K diskless nodes with 2 processors per node.
- Only 256MB RAM per processor.
- Additional service nodes (I/O).
- Estimated 360 Tera FLOPS.
- Over 150k processors.
- Global tree network.
- 3-D torus network.
- Gigabit Ethernet.
- Operational in 2005.



Scalability Issues

- How to make use of 100,000 processors?
- System scale jumps by a magnitude.
- Current algorithms do not scale well on existing 10,000-processor systems.
- Next generation peta-scale systems are useless if efficiency drops by a magnitude.



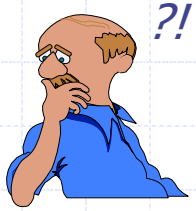
Fault-tolerance Issues

- How to survive on 100,000 processors?
- Failure rate grows with the system size.
- Mean time between failures may be a few hours or just a few minutes.
- Current solutions for fault-tolerance rely on checkpoint/restart mechanisms.
- Checkpointing 100,000 processors to central stable storage is not feasible anymore.

ORNL/IBM Collaboration

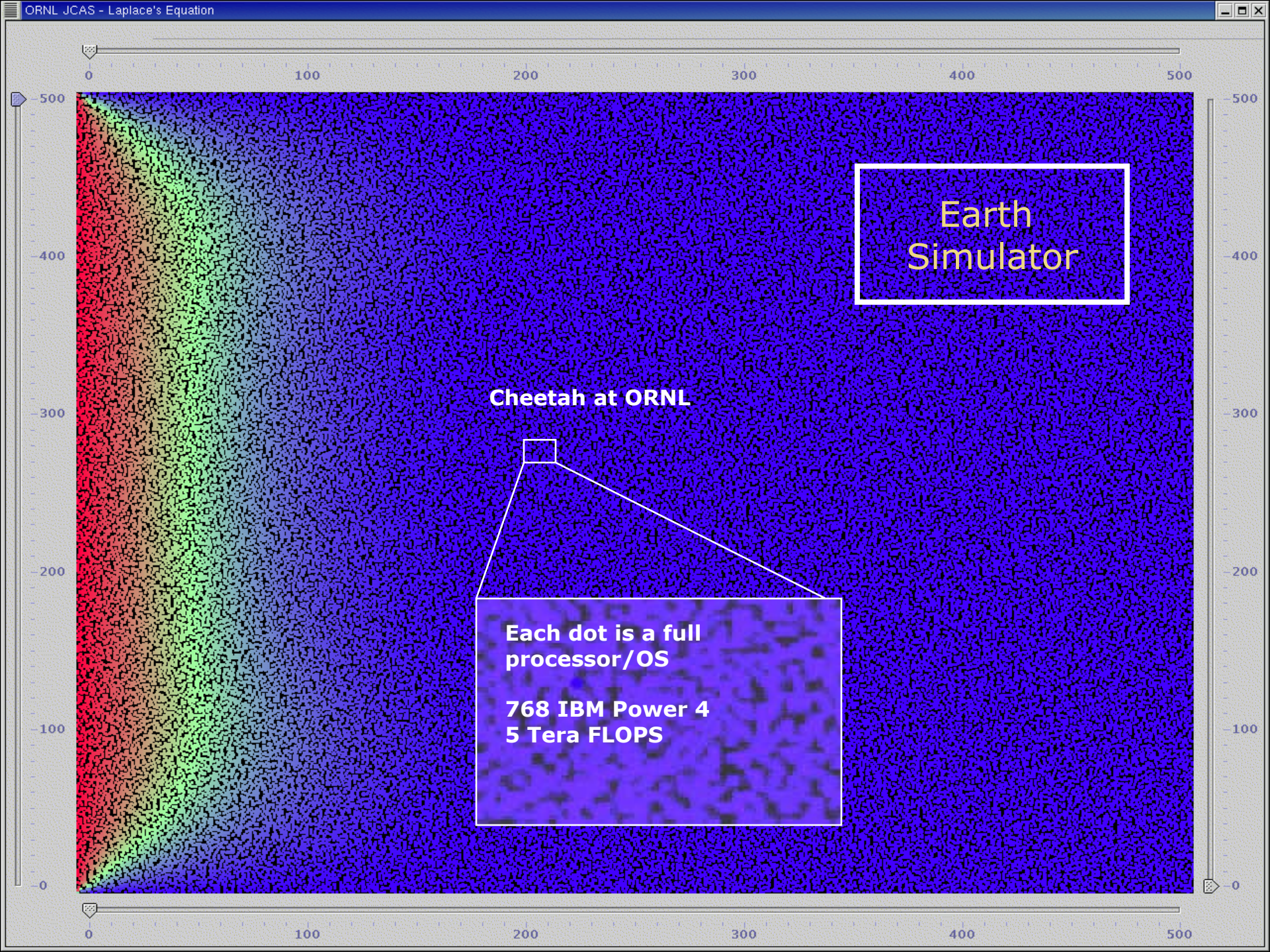
- Development of biology and material science applications for super-scale systems.
- Exploration of super-scalable algorithms.
  - Natural fault-tolerance.
  - Scale invariance.
- Focus on test and demonstration tool.

- Get scientists to think about scalability and fault-tolerance in super-scale systems!



Cellular Architecture Simulator

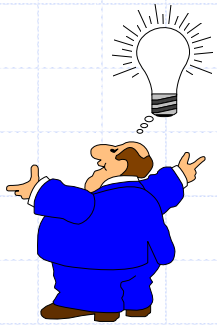
- Developed at ORNL in Java with native C and Fortran application support using JNI.
- Runs as standalone or distributed application.
- Lightweight framework simulates up to 1,000,000 processes on 9 real processors.
- Standard and experimental networks:
  - Multi-dimensional mesh/torus.
  - Nearest/Random neighbors.
- Message driven simulation is not in real-time.
- Primitive fault-tolerant MPI support.


ORNL JCAS - Laplace's Equation
0
100
200
300
400
500
Earth Simulator
Cheetah at ORNL
Each dot is a full processor/OS
768 IBM Power 4
5 Tera FLOPS

Super-scalable Fault-tolerance

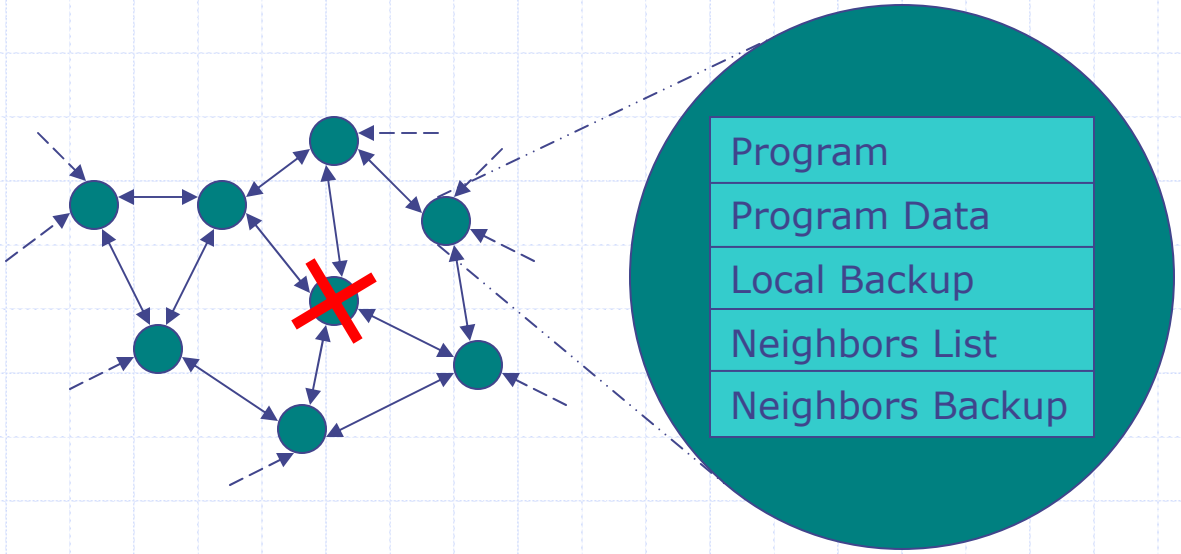
- For non-naturally fault tolerant algorithms.
- Does it makes sense to restart all 100,000 processors because one failed?
- The mean time between failures is likely to be a few hours or just a few minutes.
- Traditional centralized checkpointing is limited by bandwidth (bottleneck).

➢ The failure rate is going to outrun the recovery and the checkpointing rate.



Diskless Checkpointing

- Decentralized peer-to-peer checkpointing.
- Processors hold backups of neighbors.
- Local checkpoint and restart algorithm.
- Coordination of local checkpoints.



Diskless Checkpointing

- In case of a failure:
  - Rollback to local memory backup if necessary.
  - Restart from remote memory backup.
- Encoding semantics, such as RAID, trade off storage size vs. degree of fault tolerance.
- Very infrequent checkpointing to central stable storage (disk/tape).
- Checkpoint and application processes may be the same or different.
- Possible OS support via library/service.

Choosing Neighbors

- Physically near neighbors:
  - Low latency, fast backup and recovery.
- Physically far neighbors:
  - Recoverable multiprocessor node failures.
- Random neighbors:
  - Medium latency and bandwidth.
  - Acceptable backup and recovery time.
- Optimum: Pseudorandom neighbors based on system communication infrastructure.

Backup Coordination

- All peer-to-peer checkpoints need to be consistent with the global application state.
- Includes local states and in-flight messages.
- No backup coordination for checkpoints with no communication since the last one or start.
- Coordination techniques:
  - Global synchronization.
  - Local synchronization.

Global Synchronization

- Global application snapshot (e.g. barrier) at stable global application state.
- Synchronous backup of all local states.
- Synchronizes complete application.
- Preferred method for communication intensive applications.
- Easy to implement.

Local Synchronization

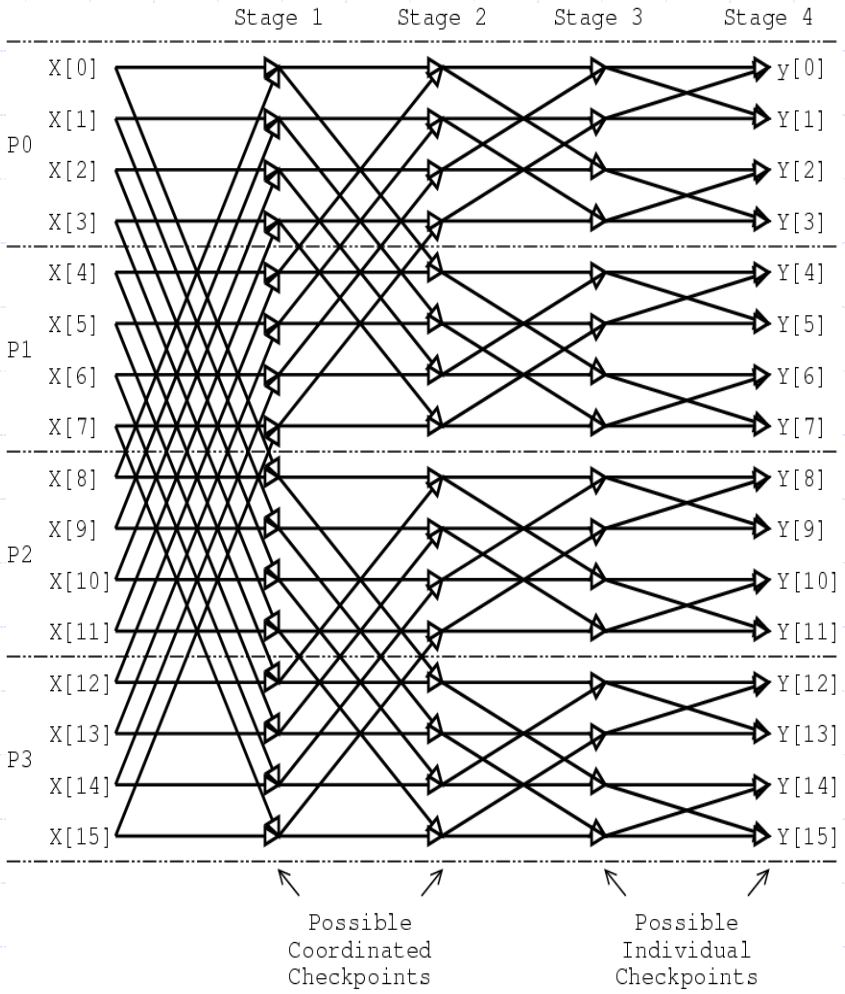
- Asynchronous backup of local state and in-flight messages (extensive message logging).
- Acknowledgements for messages to keep accurate records of in-flight messages.
- Additional local group communication.
- Different methods to retrieve missed messages from neighbors (replay/lookup).
- Preferred method for less communication intensive applications.
- More complicated to implement.

Application to FFT

- Distributed and transposed FFT:
  - Not naturally fault-tolerant.
  - Every process is important.
  - Not scale invariant.
  - Mixture of local and global communication.
  - Well known algorithm behavior.
- Other Fourier transform algorithms may be naturally fault-tolerant or scale better.
- They are not considered here.

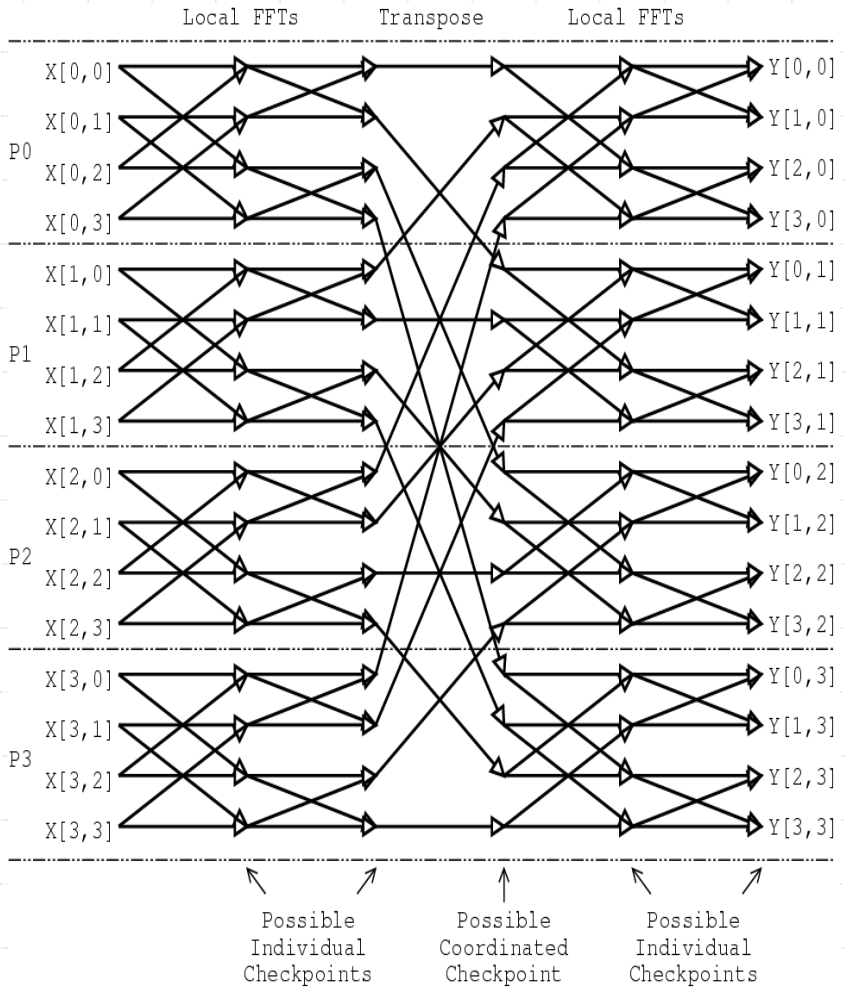
How to checkpoint DFFT?

- Individual checkpoints with no synchronization.
- Coordinated checkpoints with global sync. due to heavy message load.
- Number of coordinated checkpoints depends on coefficients/processor.



How to checkpoint TFFT?

- Individual checkpoints with no synchronization.
- Coordinated checkpoints with local sync. due to light message load.
- Coordinated checkpoints only after transpose.
- More efficient than DFFT.



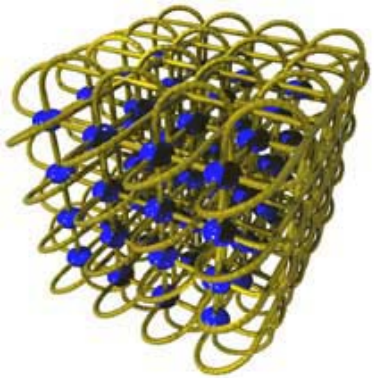
Observations

- Diskless peer-to-peer checkpointing on super-scale architectures is possible.
- Synchronization methods have different strengths and weaknesses.
- Timing, latency and bandwidth data impossible to obtain from simulator.
- Real-time tests with different applications are needed for further discussion.
- Final real-world implementation requires super-scalable FT-MPI or PVM.

Conclusions

- Super-scale systems with 100,000 and more processors become reality very soon.
- Diskless peer-to-peer checkpointing provides an alternative to natural fault-tolerance.
- A lot of research still needs to be done.

Februrary, 2003



Diskless Checkpointing on Super-scale Architectures

Applied to the Fast Fourier Transform

Christian Engelmann, Al Geist
Oak Ridge National Laboratory