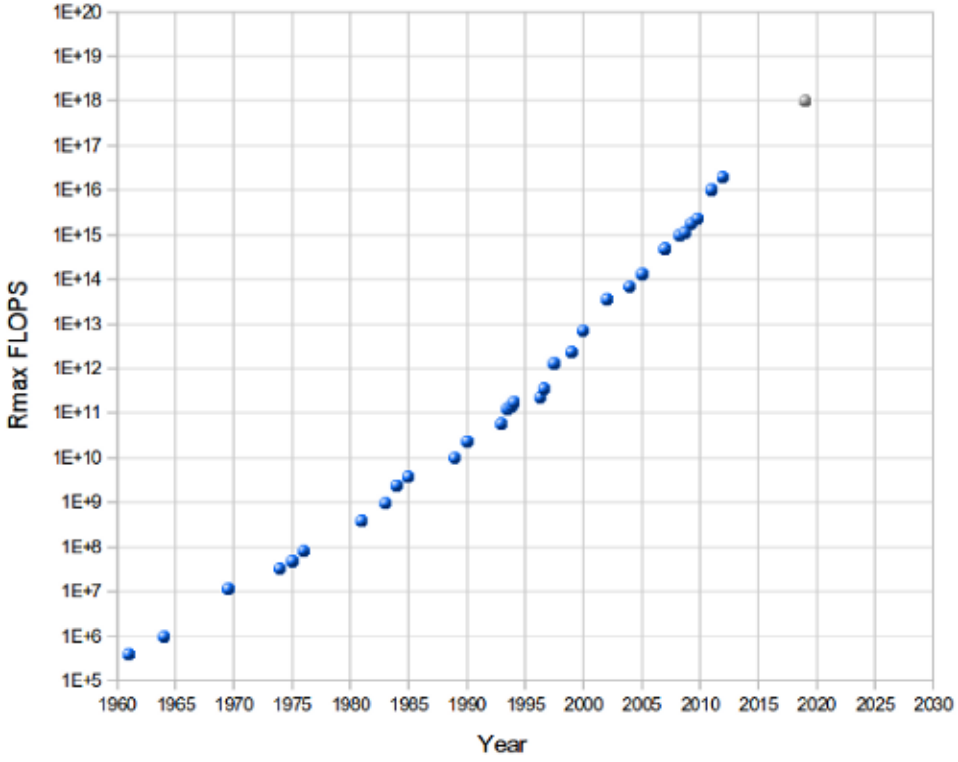
Scalable Self-Healing Algorithms for High Performance Scientific Computing

Zack Tillotson
November 2010

Motivation

- The size of HPC systems is outgrowing traditional checkpointing strategies
- Disk bandwidth
- Multiple failures
- =Scalable



Self Healing Framework

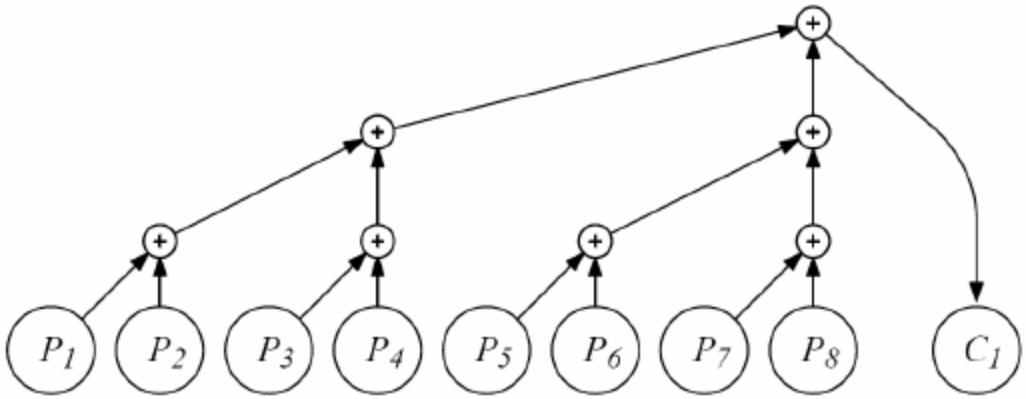
- 1. FT-MPI
- 2. Diskless Checkpointing
- 3. Pipelining
- 4. Weighted Checksums

FT-MPI

- MPI + Fault Tolerance
- How does it work?
  - Survives failures
  - Failure actions:
    - Processors
    - Messages
- Fast
  - Scalable

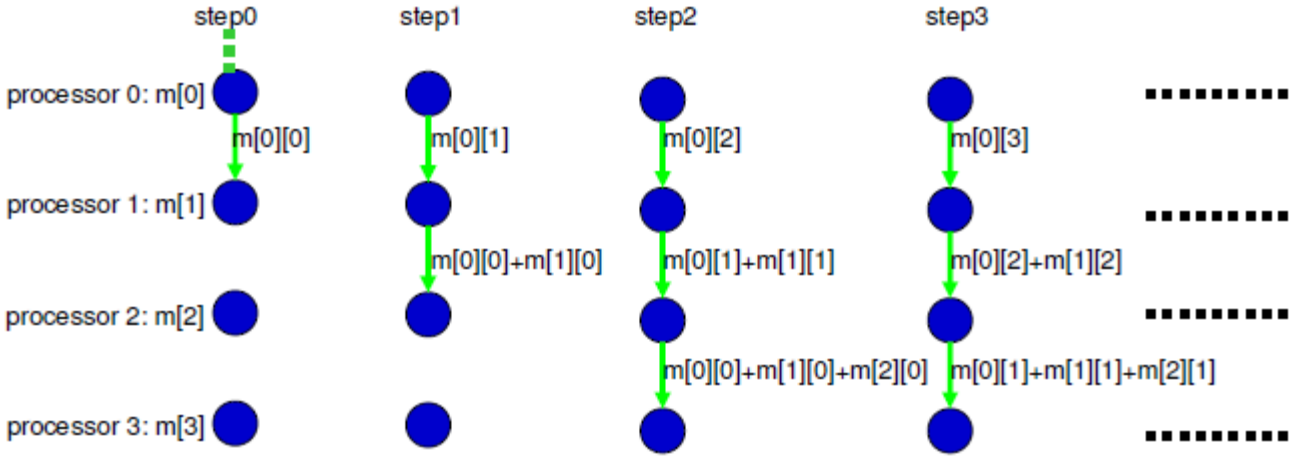
Diskless Checkpointing

- Application level/System level
  - Synchronization points
  - Size
  - Heterogeneous environment
- Bit stream vs floating point number
  - Galois field vs Floating point arithmetic
  - Heterogeneous recov
- Parity/Checksums
  - Log speed
- Scalable



Pipelining

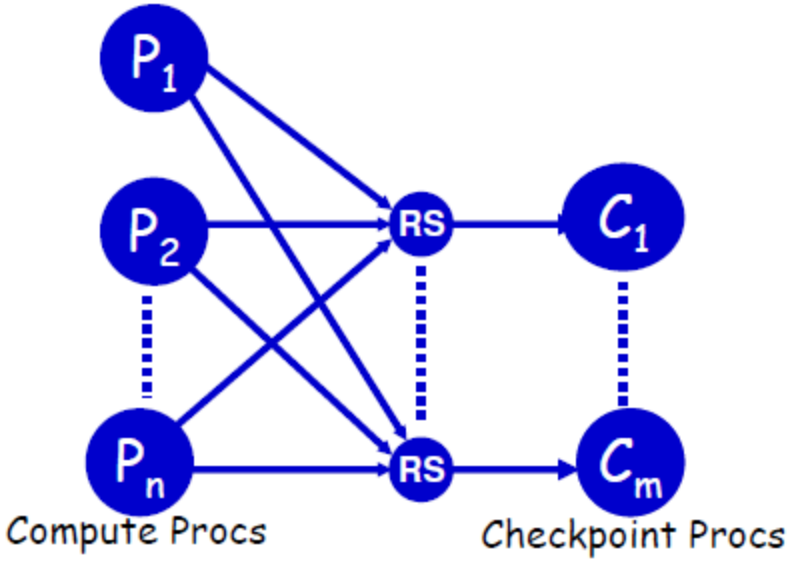
- Segment message
- Simultaneously send and receive



- Checkpoint in p + s + 2
- Scalable

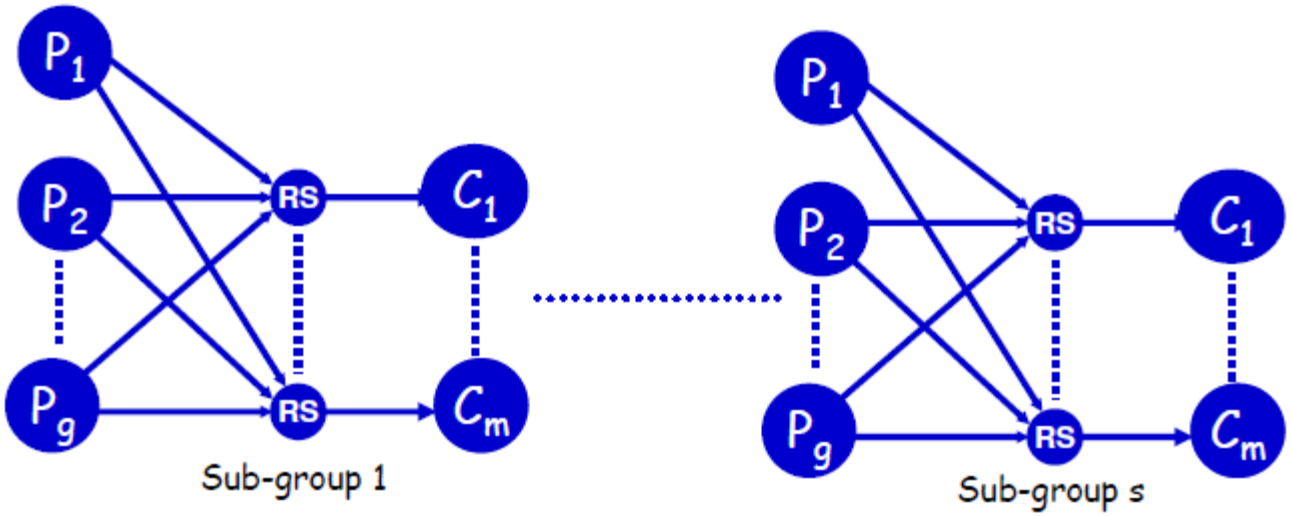
Weighted Checksum

- Reed-Solomon encoding
  - Uses matrices which have special properties
  - E.g. Vandermonde, Cauchy, Gaussian Random
  - Use gaussian random matrices with floating point operations
    - Round off errors
- Basic weighted checksum
  - To handle k errors, k checkpoint processors
  - Linear time



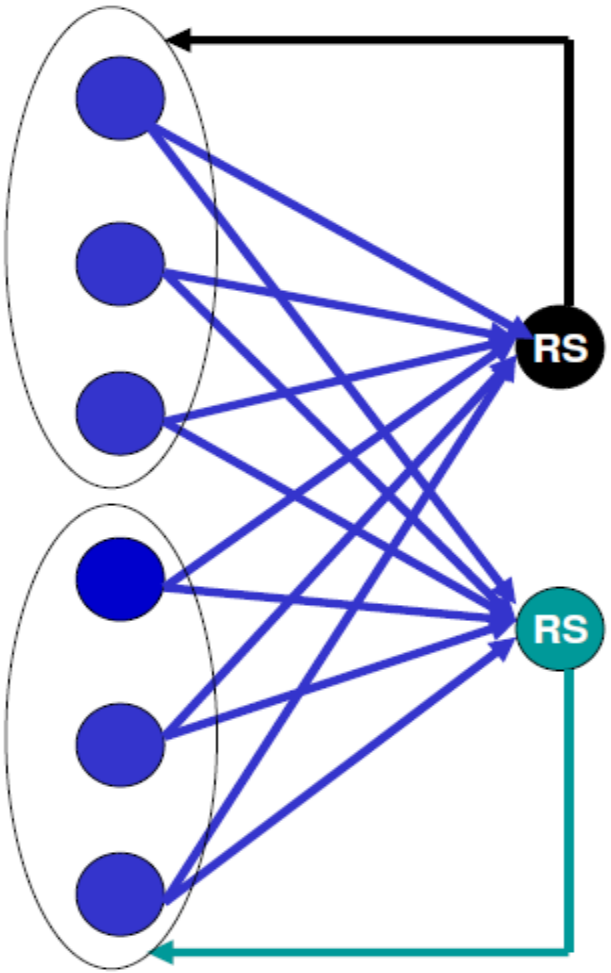
Weighted Checksum

- 1 Dimensional Weighted Checksum
  - Split processors into groups of constant size
  - To handle k errors, k checkpoint processors per group
  - Constant time with regard to total number of processors



Weighted Checksum

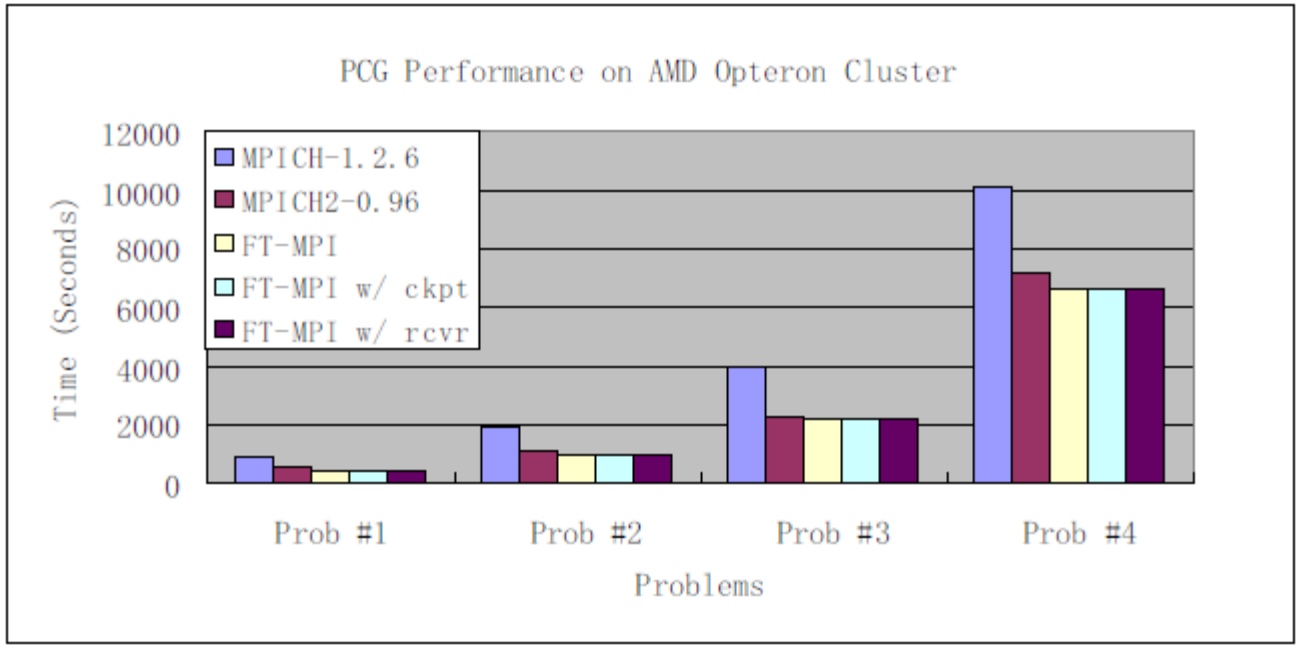
- Localized Weighted Checksum
  - To handle k errors, partition processors into groups of k(k+1) size
  - Each processor encodes with k+1 other processors
  - Any k processors going down can be recovered
  - No dedicated checkpointing processors



Performance

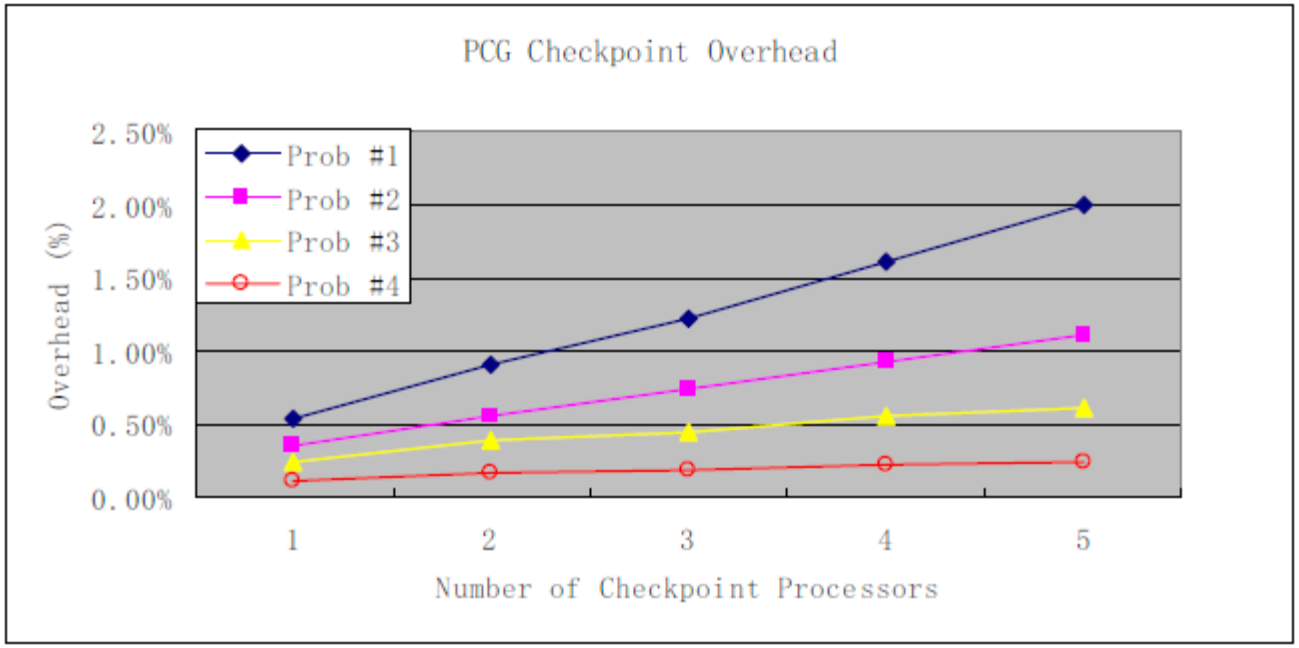
- Preconditioned Conjugate Gradient (PCG) program
  - Basic Weighted Checksum Scheme
  - Small checkpoints
  - FT-MPI
    - Processor rebuild mode
    - Message drop mode

Performance: FT-MPI



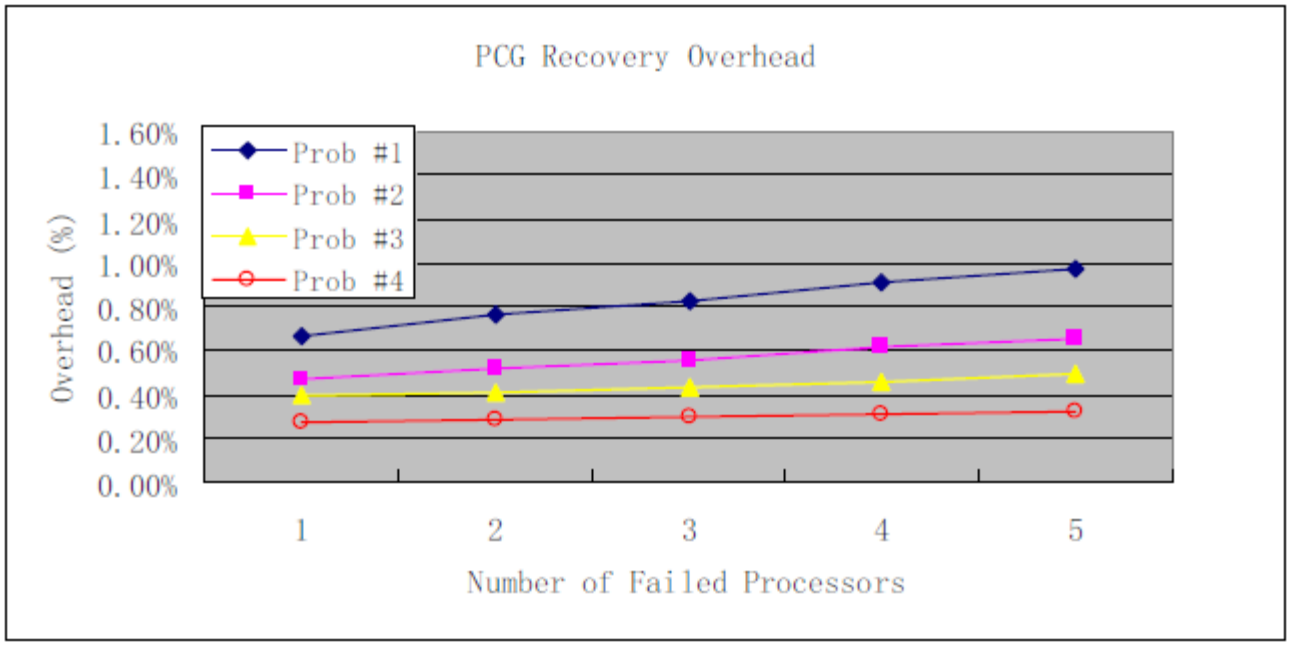
Performance: Pipelining

- No faults, all increase in time is due to overhead of computing checkpoints



Performance: Pipelining

- Failure of processors simulated each run

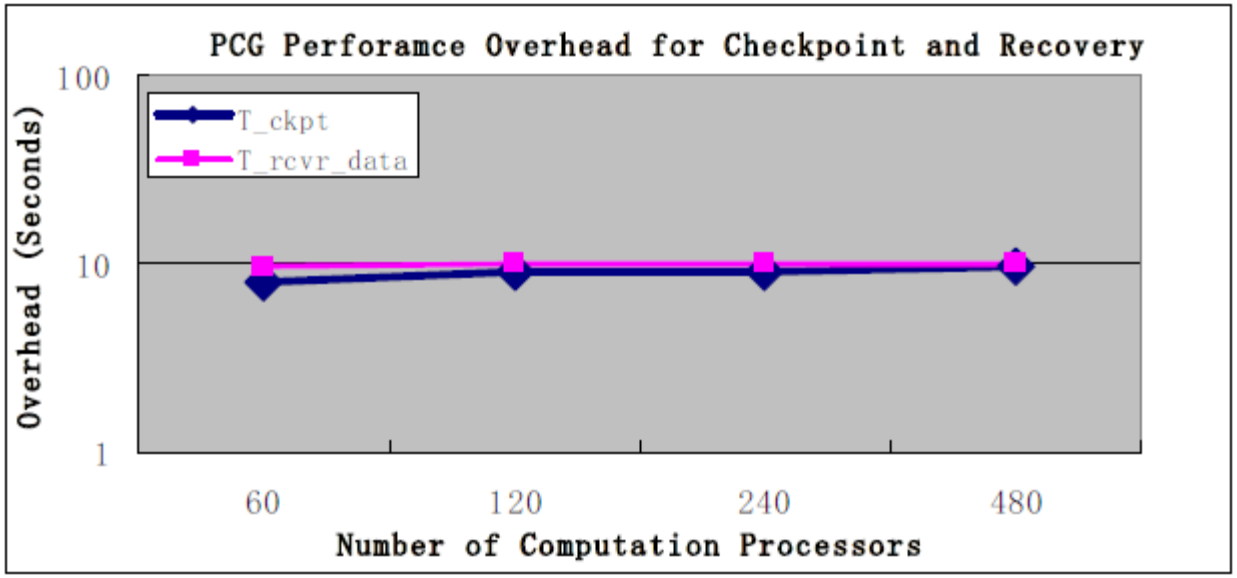


Performance: Round Off Errors

- Round off error due to recovery

Residual	Prob #1	Prob #2	Prob #3	Prob #4
0 proc	3.050e-6	2.696e-6	3.071e-6	3.944e-6
1 proc	2.711e-6	4.500e-6	3.362e-6	4.472e-6
2 proc	2.973e-6	3.088e-6	2.731e-6	2.767e-6
3 proc	3.036e-6	3.213e-6	2.864e-6	3.585e-6
4 proc	3.438e-6	4.970e-6	2.732e-6	4.002e-6
5 proc	3.035e-6	4.082e-6	2.704e-6	4.238e-6

Performance: Scalability



Limitations

- Checkpointing
  - Checkpoint size
  - Different checkpoint schemes
- General failures
  - Add disk based checkpointing
- Application level system

Questions

Zack Tillotson
November 2010