**Summary**

- Faults are becoming common
- Conventional checkpoint-and-recovery (CPR) is inefficient and not scalable
- Our approach: Globally Precise Restartability
- Uses deterministic execution, or ordered programs
- Order threads (tasks) at synchronization points
- Checkpoint at task boundaries
- Selectively restart only affected tasks
- Performs globally

**Our Approach**

- Globally precise restartable recovery system (GPRS): C++ runtime library
- End-to-end fault recovery: user program (Pthreads, Ordered2) + runtime system + I/O + system calls
- Tracks and releases task checkpoints in order; Restores and restarts faulted tasks

**Key Principles**

- Use deterministic execution, or ordered programs
- Order threads (tasks) at synchronization points
- Checkpoint at task boundaries
- Selectively restart only affected tasks

**Creating Order**

- Round-robin across threads in program text
- Balance-aware ordering to minimize performance loss

**Future Directions**

- Apply principles to larger systems; test design scalability
- Use distributed runtime data structures, hierarchical design
- Combine with other techniques, e.g., containment domain[s]}

---

**Effective execution at no overheads**

**Scalable fault-tolerance with system size**

**Conventional Approach**

- Checkpoint all threads periodically; Recover when necessary

**Faults in Parallel Systems**

- "Local"
  - Affects a single thread
  - Handled locally on a processor
- "Global"
  - Focus of this work
  - Affects multiple threads
  - Cannot be handled locally

**Evaluation**

**Scalable Fault Tolerance in Multiprocessor Systems**

Gagan Gupta (Advisor: Gurindar S. Sohi)

(gagang, sohi)@cs.wisc.edu