Leaping Shadows: Adaptive and Power-aware Resilience for Extreme-scale systems

Xiaolong Cui, Taieb Znati, Rami Melhem
Computer Science, University of Pittsburgh

Introduction

❖ System scale keeps growing for both HPC and Cloud.

|	TODAY	FUTURE
Performance	20 Pf/s	1 Ef/s
Power	8 MW	20 MW
System size	18,688	100k-1m
Memory	0.7 PB	32-64 PB
Network BW	1.5 GB/s	100-1000 GB/s
I/O Capacity	15 PB	300-1000 PB

System level **failure rate** will increase by orders of magnitude

Power/energy consumption will dominate CAPEX

Low efficiency + **high cost**

Shadow Collocation

❖ Collocate multiple shadow processes on each node
  ❖ Reduces shadow processes' execution rate
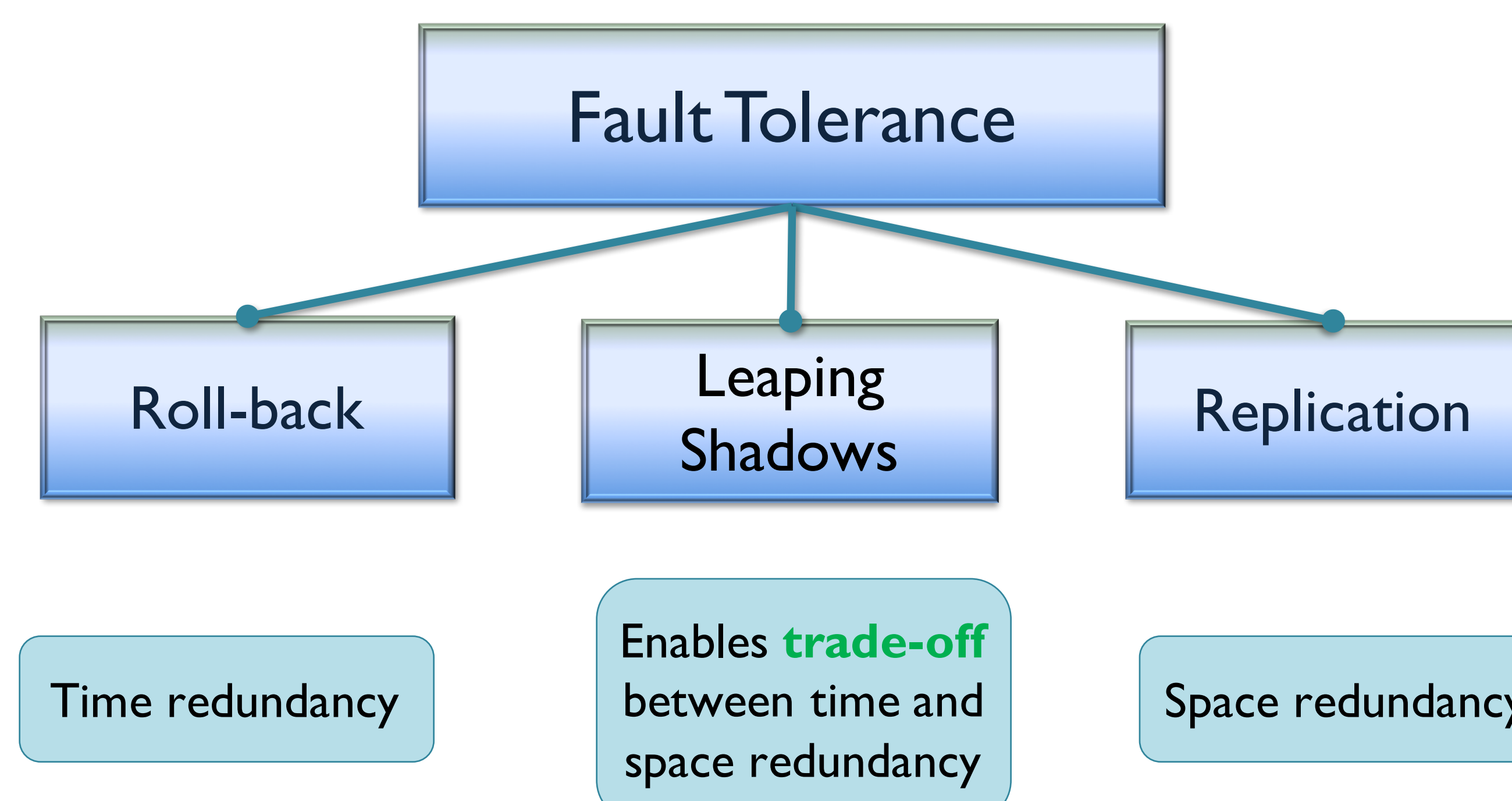 ❖ Reduces **hardware** and **power** requirement

M₁ M₂ M₅ M₆

S₁ S₂ / S₃ S₄   S₅ S₆ / S₇ S₈

M₃ M₄ M₇ M₈
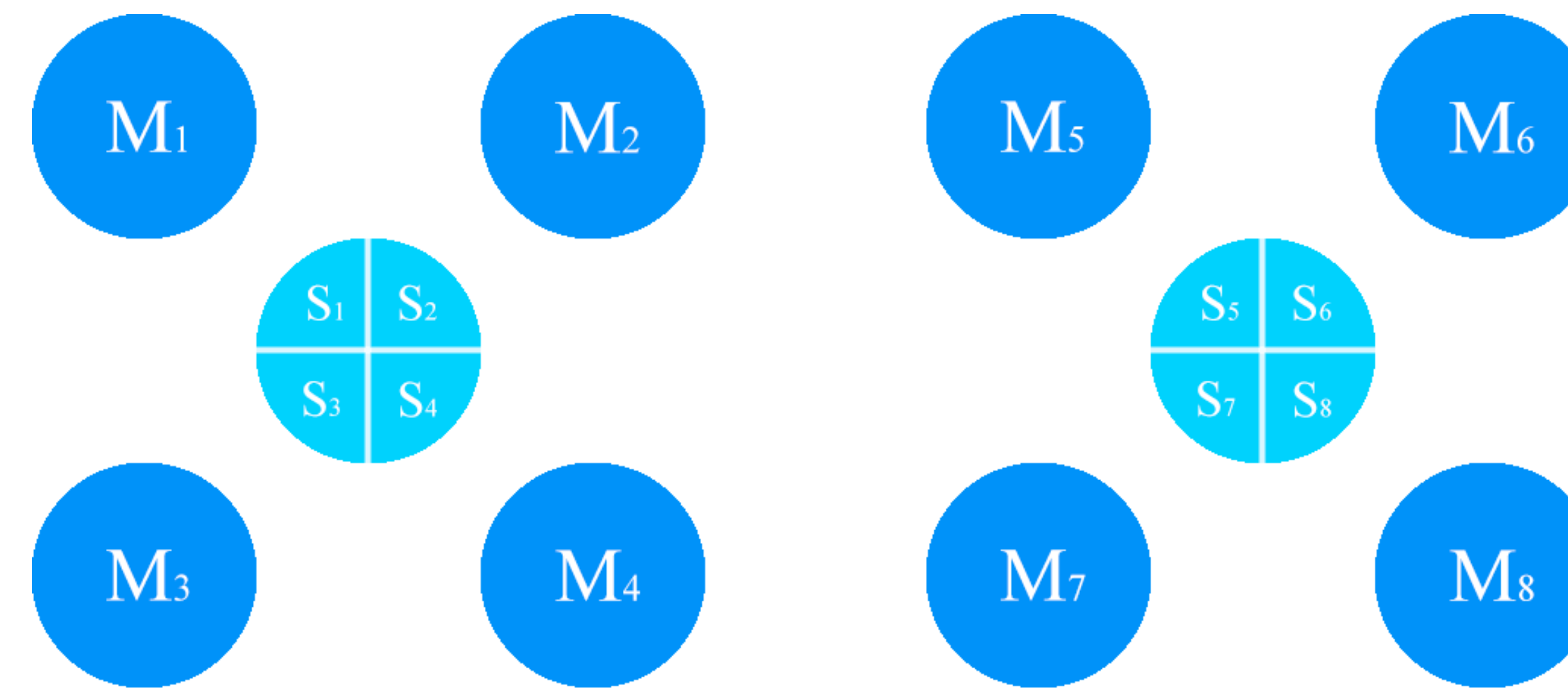
Collocation ratio controls the number of shadows collocated

Shadow Leaping

◆ The lagging shadow processes can benefit from the faster execution of the main processes
◆ Sync states from the main processes to the shadows

Overlap shadow leaping with failure recovery
↓
Forward progress with **minimized energy**
↓
Reduces main and shadow **distance**
↓
Minimizes subsequent **recovery time**

Work completed — main — shadow — t_f t_{sync} t_r — Time

Work completed — shadow leaps — main and shadow resume execution — main — shadow — t_f t_{sync} t_r — Time

Lazy Shadowing

❖ Each process is associated with a "**shadow**"
❖ Shadow processes initially execute at reduced rate
❖ Upon failure of a main process, its shadow process increases execution rate to recover and complete task

Work Completed (W) — Task Completes — Main Process — Shadow Process — t_c^m Time

Work Completed (W) — Task Completes — Main Process — Shadow Process — t_c^s Time

Reduced execution rate
↓
Power/Energy **savings**

Dynamic increase of rate upon **failure**
↓
Delay **minimized**

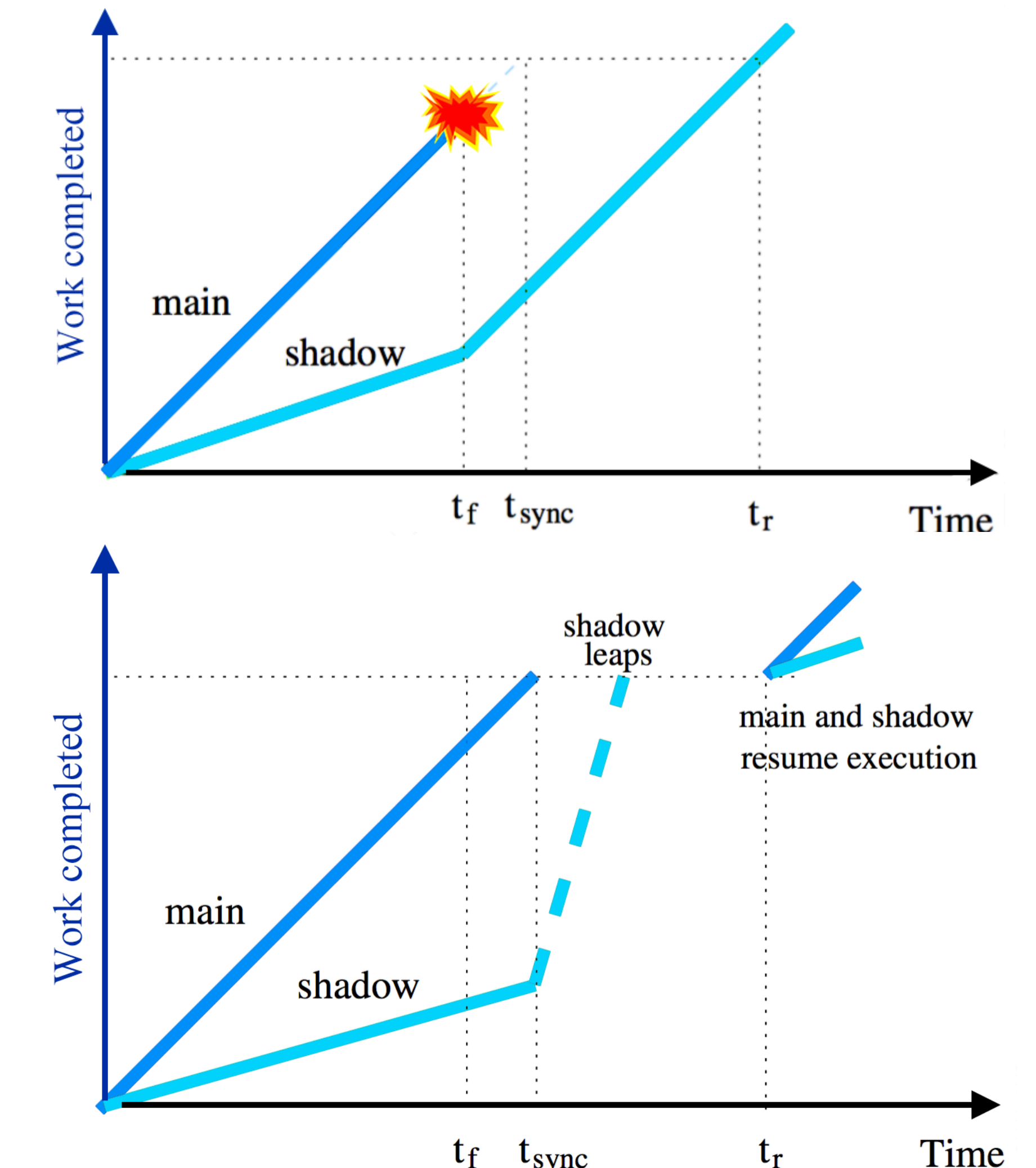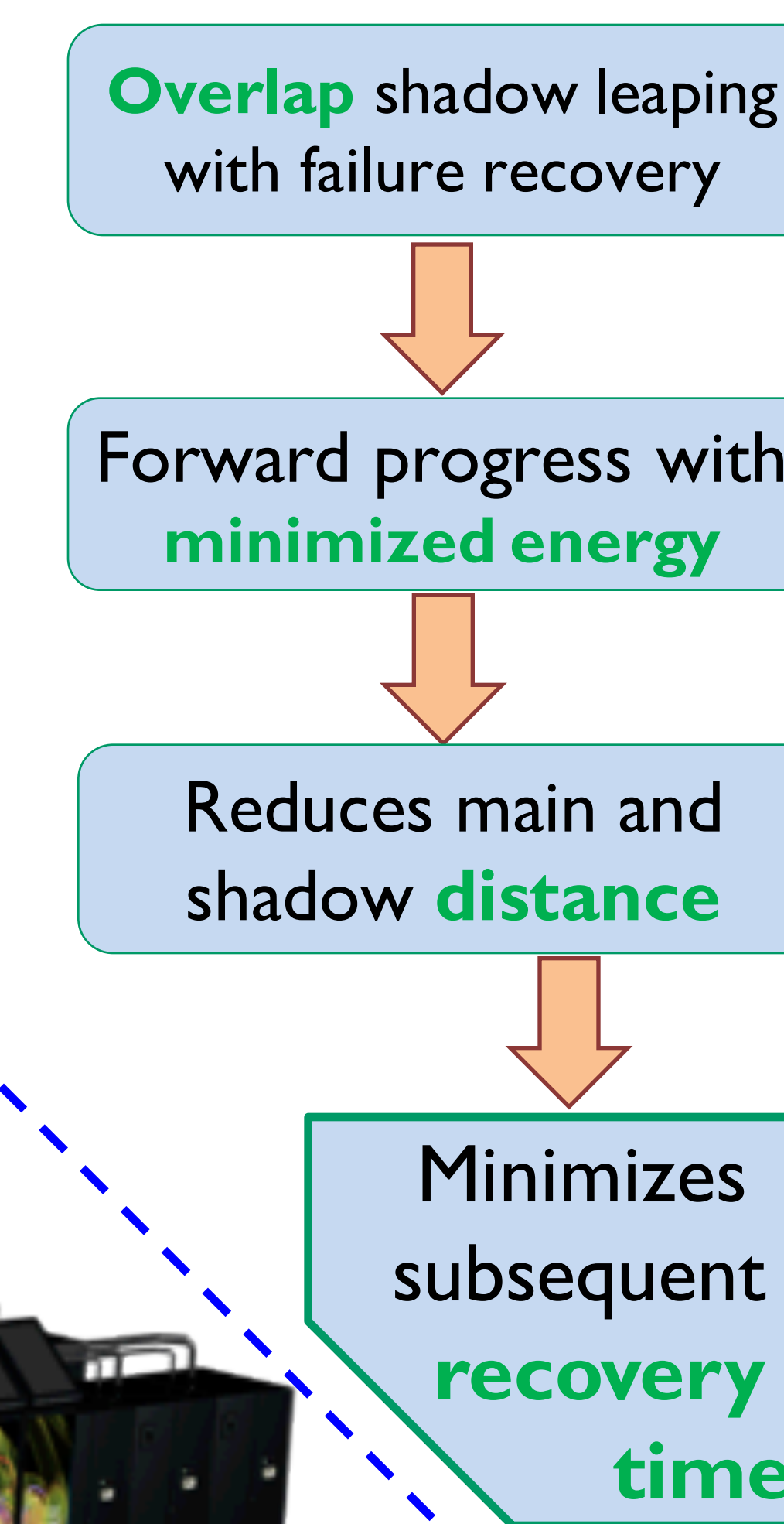## World's #1 Open Science Supercomputer

Flagship accelerated computing system | 200-cabinet Cray XK7 supercomputer
18,688 nodes (AMD 16-core Opteron + NVIDIA Tesla K20 GPU)
CPUs/GPUs working together – GPU accelerates | 20+ Petaflops
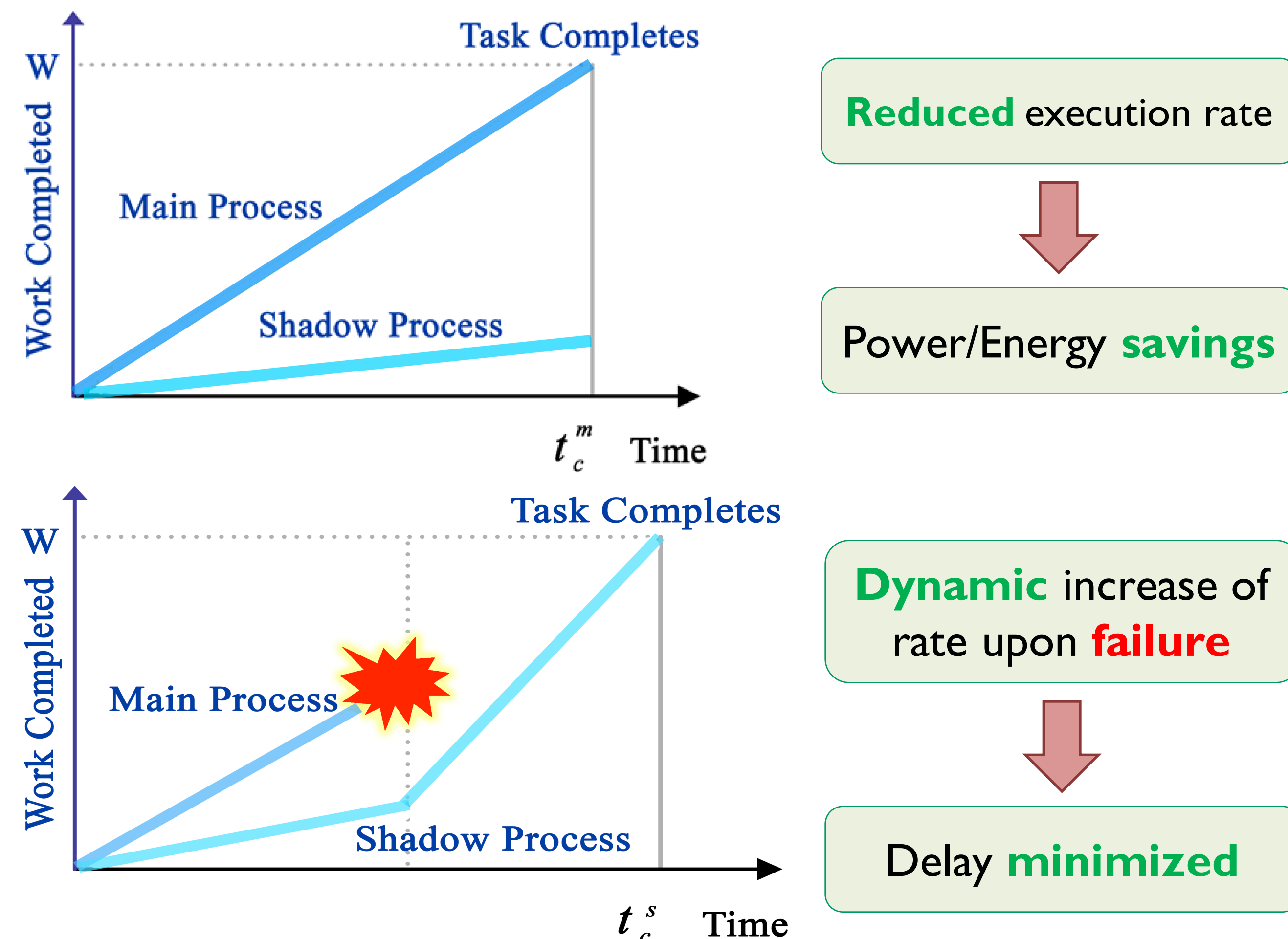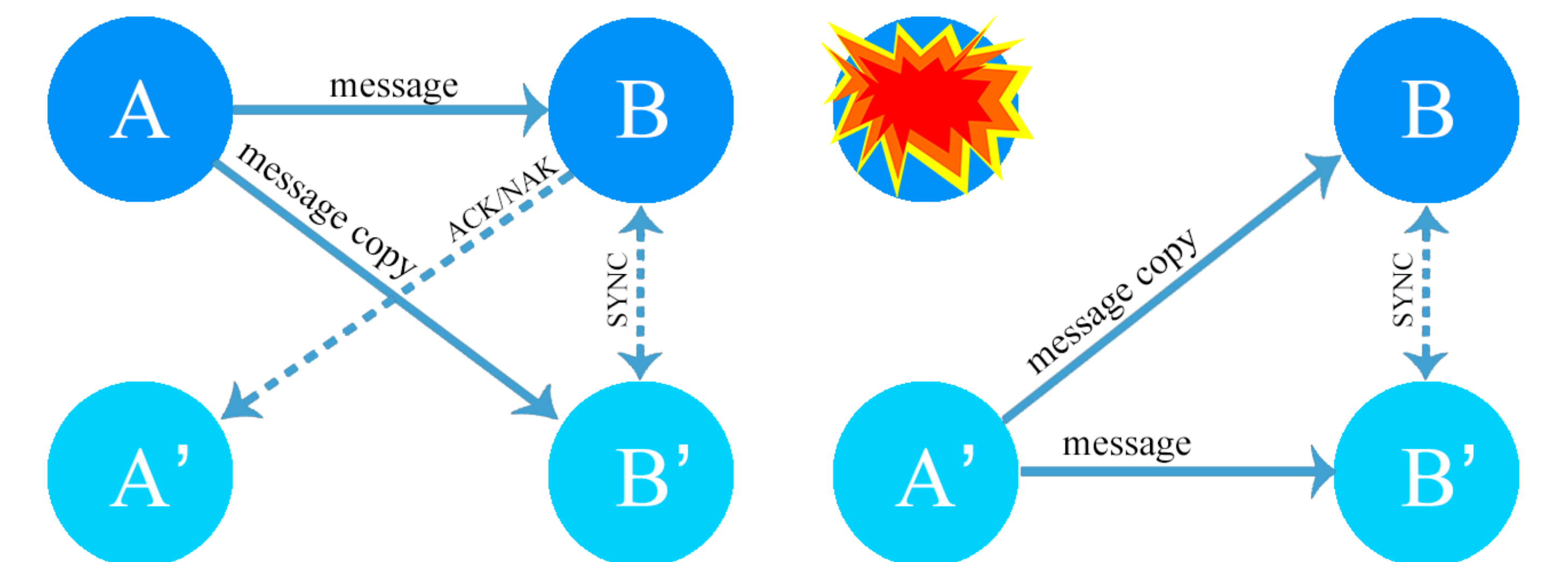
The Fault Tolerance Spectrum

Fault Tolerance
├─ Roll-back
├─ Leaping Shadows
└─ Replication

Time redundancy | Enables **trade-off** between time and space redundancy | Space redundancy

MPI Implementation

❖ A library (lsMPI) is inserted between application and MPI that transparently supports Leaping Shadows
❖ Failure detection is delegated to User Level Fault Mitigation

A — message → B   message copy / ACK/NAK / SYNC   A' ⟶ B'

A' — message → B   message copy / SYNC   A' — message → B'

❖ Main process duplicates each message
❖ ACK/NAK is used to guarantee consistent promotion of a shadow process in the case of a failure
❖ Main process is responsible for resolving non-determinism, such as MPI_ANY_SOURCE receive, MPI_Wtime( )
❖ Collectives use lsMPI internal point-to-point communications

CONTACT: Xiaolong Cui xic51@pitt.edu