

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

Xiaolong Cui¹, Tariq Alturkestani², Esteban Meneses³, Taieb Znati¹, Rami Melhem¹

¹Computer Science, University of Pittsburgh, USA

²King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

³School of Computing, Costa Rica Institute of Technology, Costa Rica



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	100 K-1 M
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

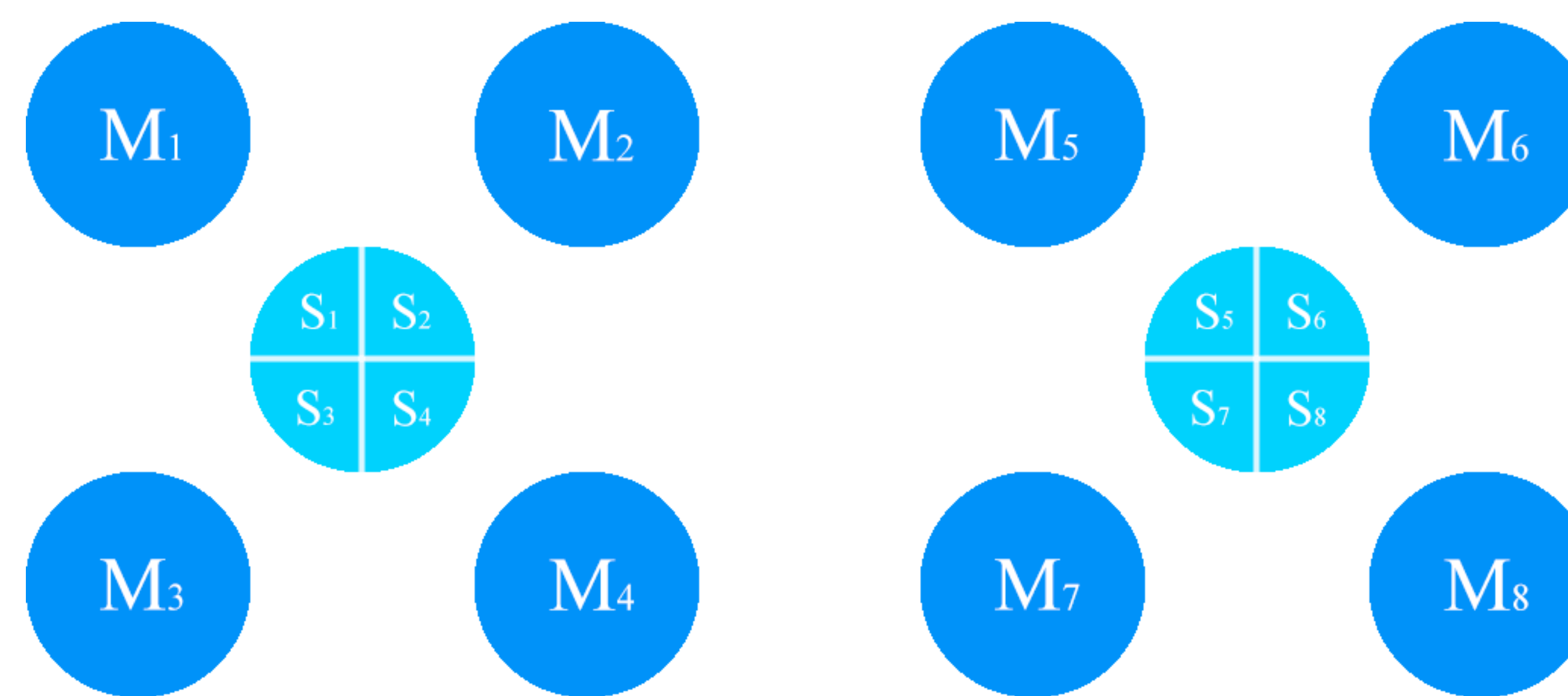
System level **failure rate** will increase by orders of magnitude

Power/energy consumption will dominate CAPEX

Low efficiency + high cost

Shadow Collocation

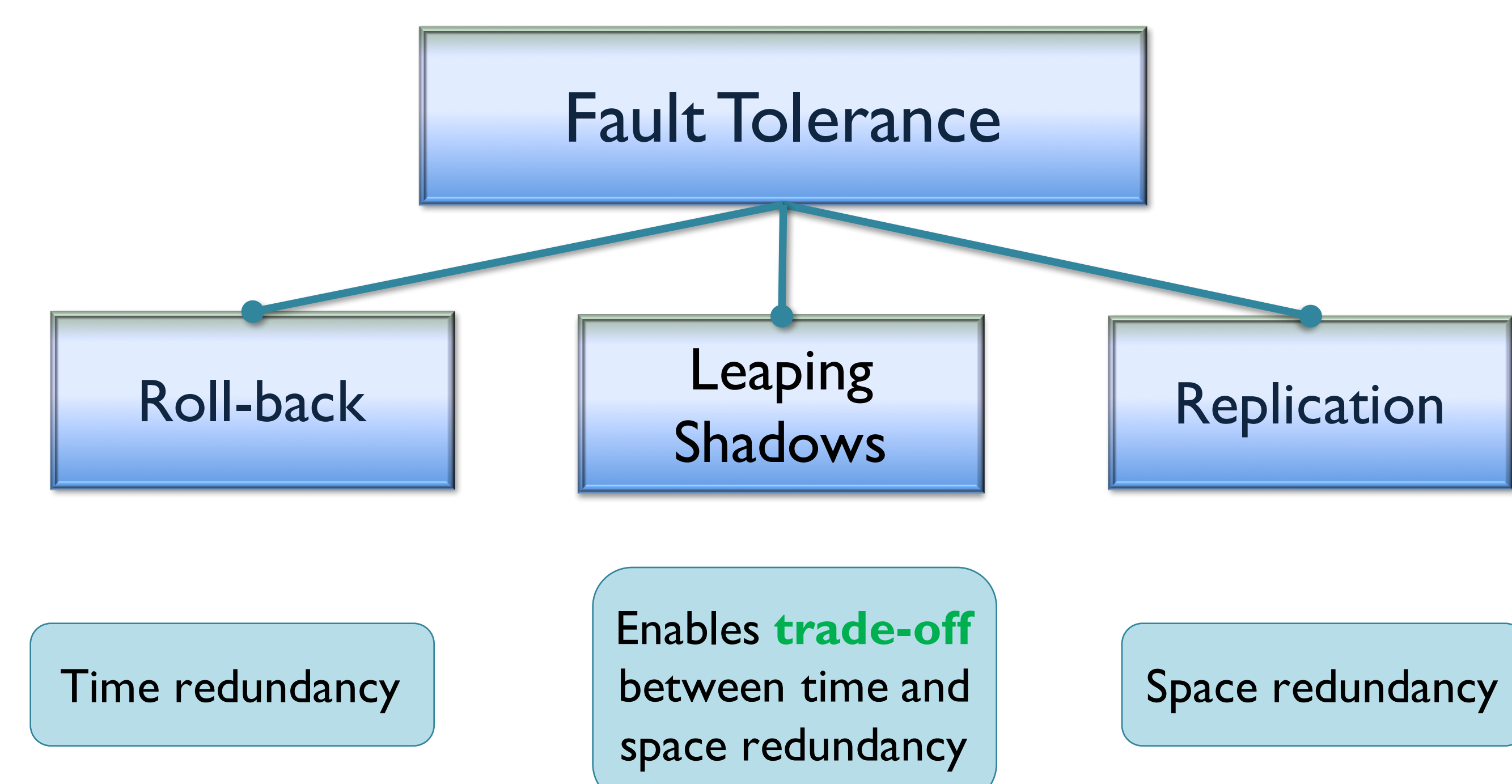
- ❖ Collocate multiple shadow processes on each node
- ❖ Reduces shadow processes' execution rate
- ❖ Reduces **hardware** and **power** requirement



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



Shadow Leaping

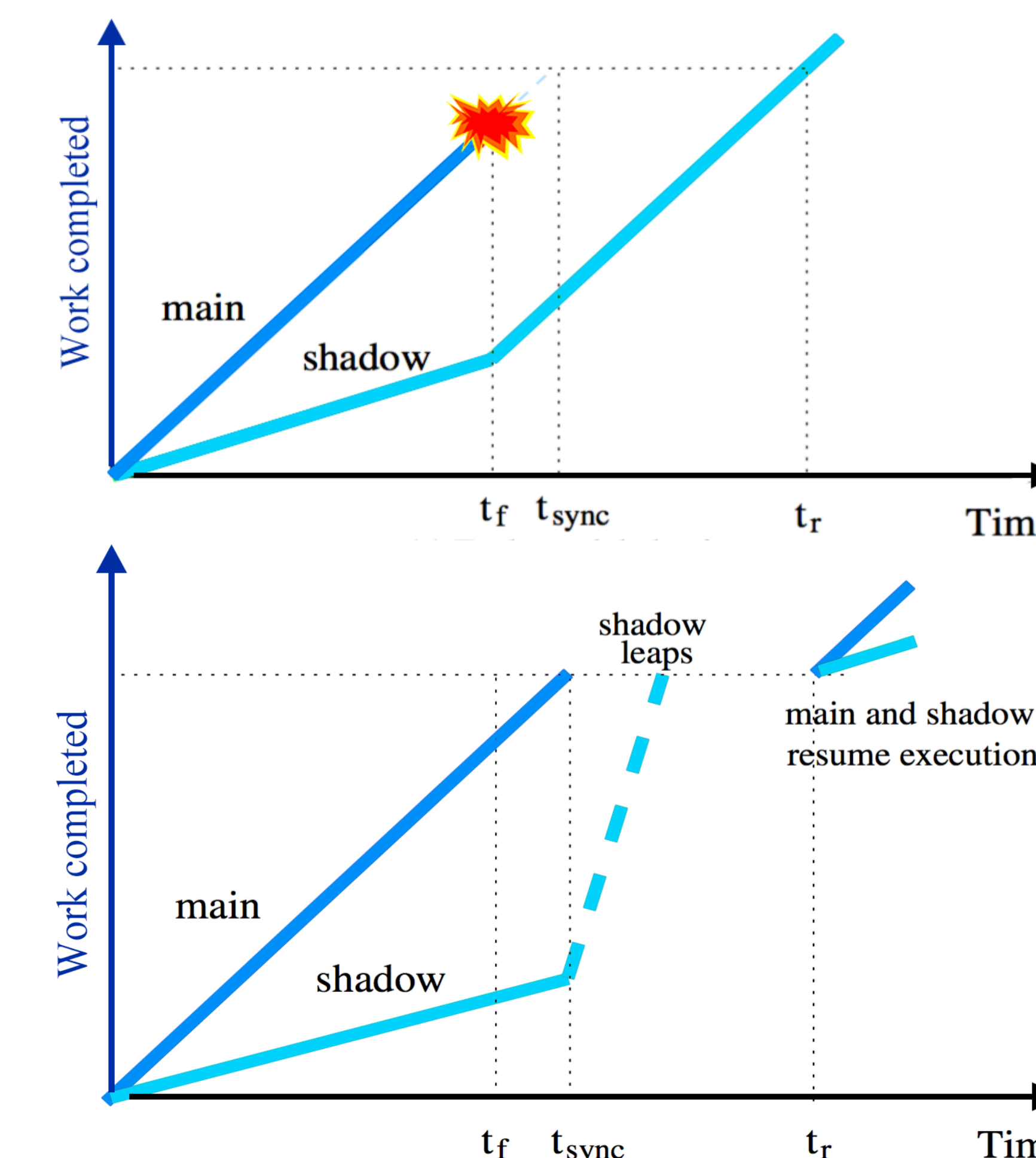
- ❖ The lagging shadow processes can benefit from the faster execution of the main processes
- ❖ Sync states from the main processes to the shadows

Overlap leaping with recovery

Forward progress with **minimized power**

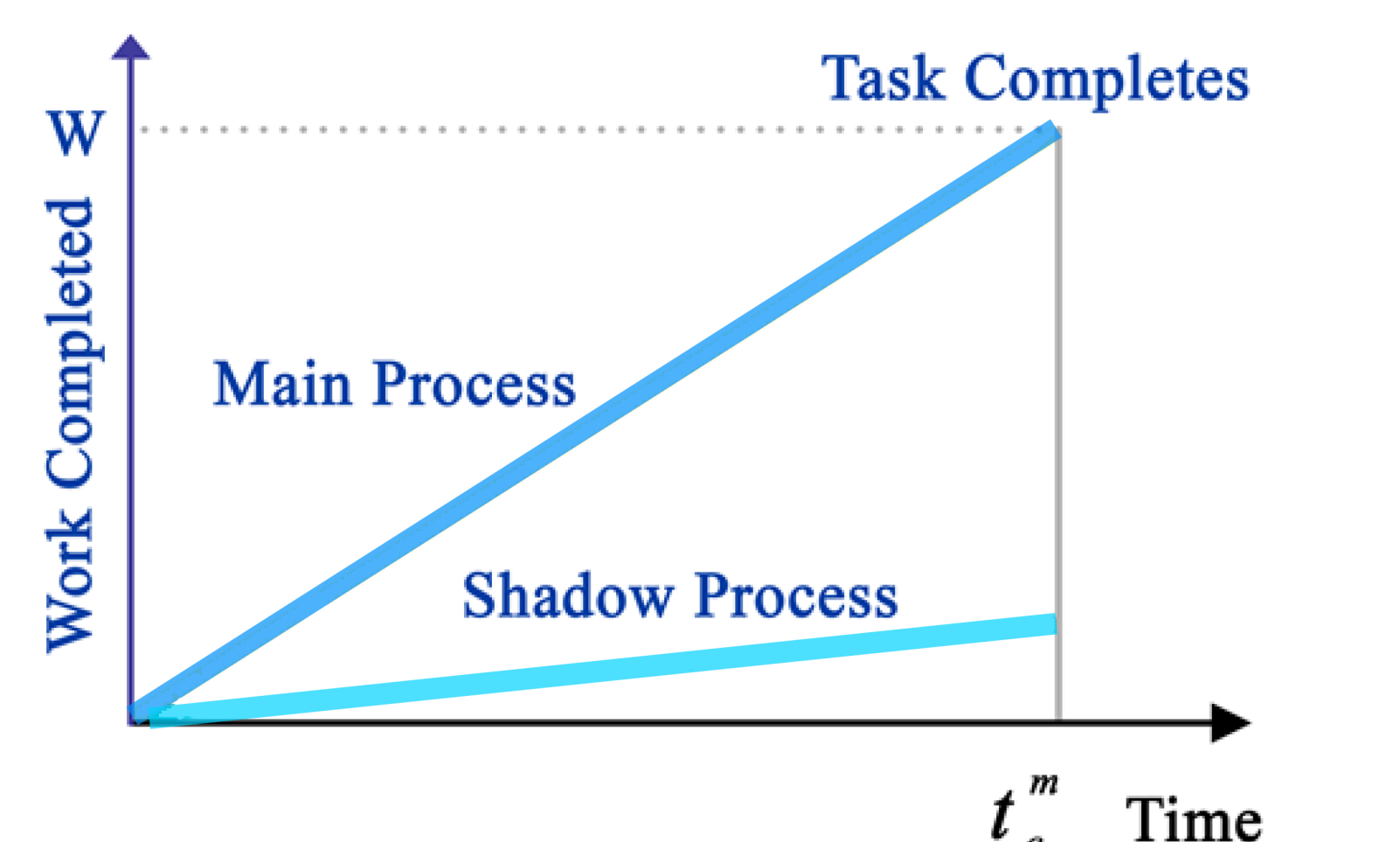
Reduces main and shadow **distance**

Minimizes subsequent **recovery time**



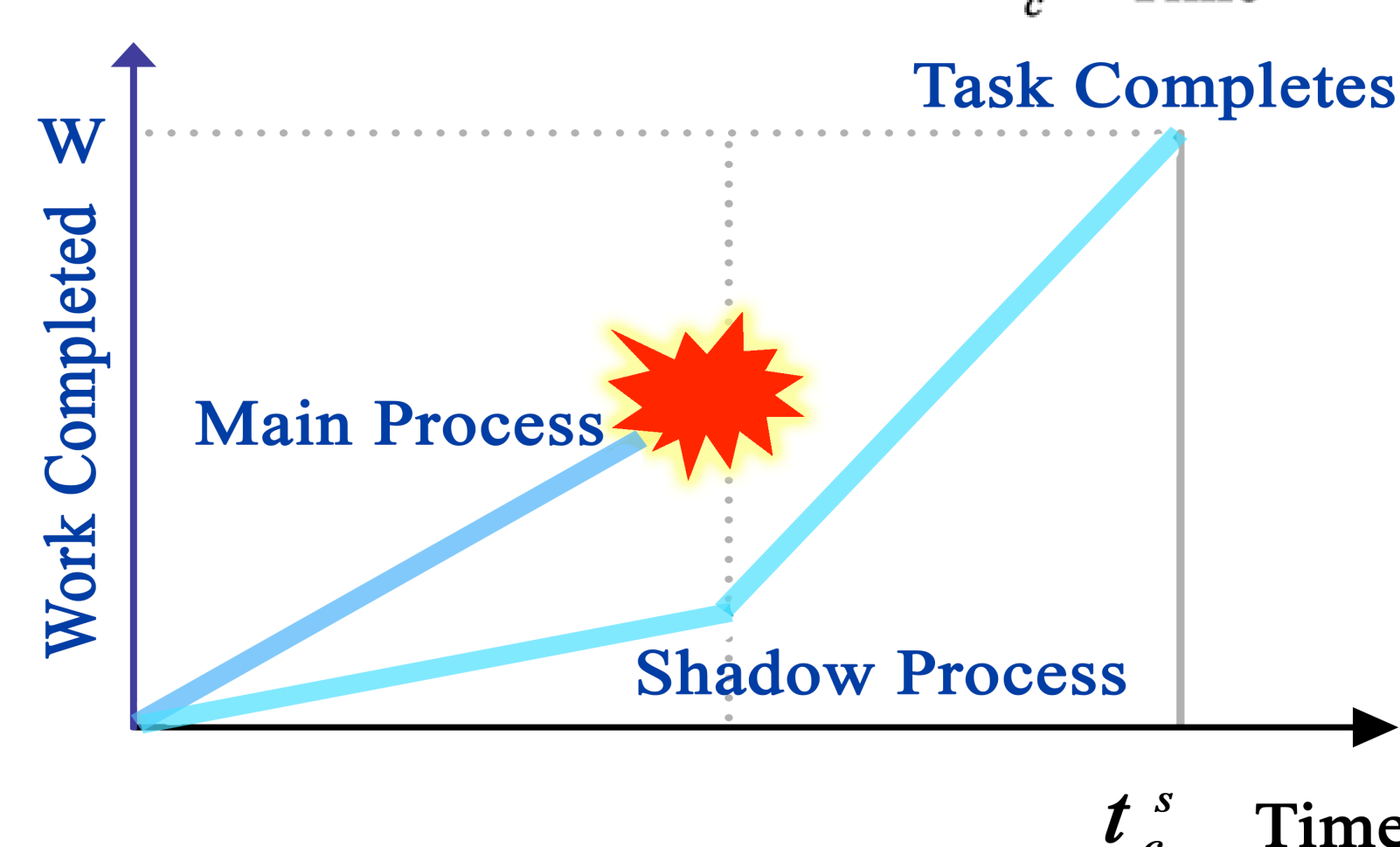
Lazy Shadowing

- ❖ 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



Reduced execution rate

Power/Energy **savings**

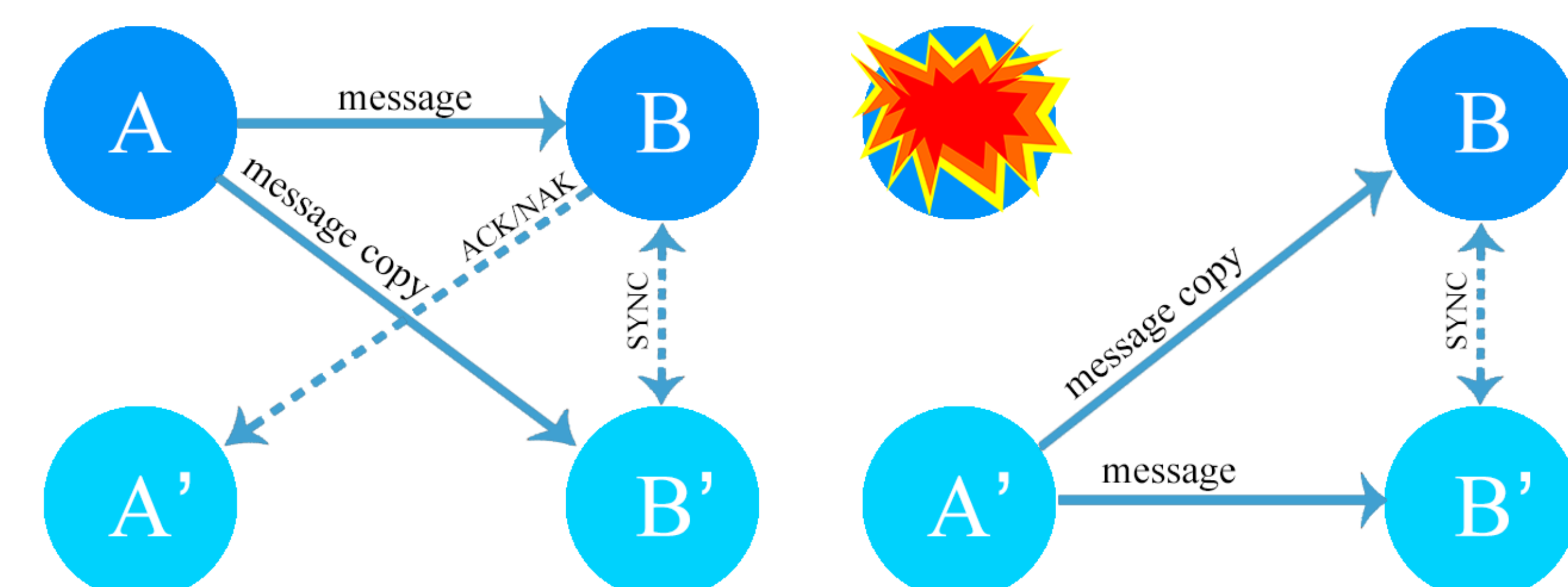


Dynamic increase of rate upon **failure**

Delay **minimized**

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



- ❖ 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