Motivation: Replication Cost vs. Guarantees

- **Primary-backup**: f+1 replicas to tolerate f crash failures
- **Paxos**: 2f+1 replicas to tolerate asynchronous events and f crash failures
  - **Cheap Paxos**: f+1 replicas, sacrifice availability
  - **Gnothi**: f+1 replicas, sacrifice generality
  - ......
- **BFT**: 3f+1 replicas to tolerate arbitrary failures

Can we reduce replication cost without sacrificing other properties?

**YES**

Solution: Adaptive Recovery

Complete recovery in a timely manner and make a balance between recovery and executing new requests

Why necessary?
- Increasing recovery time results in higher probability of data loss
- Long delayed flexible tasks (e.g. garbage collection) will eventually block the system

Method: compare \( \frac{\text{Fetched data}}{\text{Total data}} \) and \( \frac{\text{elapsed Time}}{\text{deadline}} \); if the former is smaller, increases recovery speed

Solution: Combine on-demand instantiation and lazy recovery

A. On-demand Instantiation
   - Activate a subset of replicas first and activate backup ones when active ones fail
   - **Low cost**
   - **Simple combination**
   - **Key Idea**: Separating agreement recovery and execution recovery
   - **Key observation**: they do not need to know prior requests
   - Solution: when an agreement node fails, ask a blank one to join instantly
   - **Execution nodes** run the application’s logic to execute tasks
   - **Key observation**: critical tasks may require fewer replicas than flexible tasks
   - Solution: activate sufficient replicas for critical tasks; use lazy recovery for flexible tasks

B. Lazy Recovery
   - Keep processing requests while recovering a backup replica in the background during recovery
   - **High availability**

Evaluation

We apply the proposed techniques to Paxos (2f+1) and build ThriftyPaxos (f+1)

A. **Throughput**: fewer replicas lead to fewer messages and higher throughput

Throughput of writing to replicated RemoteHashMap

\(~80\%\) improvement in TPS

B. **Availability**: ThriftyPaxos achieves same availability as Paxos during failure recovery

Deadline = 100 seconds
Complete recovery in 92 seconds

Deadline = 300 seconds
Complete recovery in 265 seconds