# ECE 259 / CPS 221
# Advanced Computer Architecture II
# (Parallel Computer Architecture)

## Availability



Duke University

# Outline

- **Definition and Motivation**
- **General Principles of Available System Design**
- **Traditional High-Availability Systems**
- **Recent Innovations in Available Systems**

# Definitions

- **Availability = probability that system is working OK**
  - **Function of error rate and time to recover**
- **Fault = physical defect**
  1) **Cosmic particle disrupts charge on SRAM cell**
  2) **Electromigration causes open circuit in switch in interconnect**
- **Error = manifestation of a fault**
  1) **Bit flip in cache**
  2) **Dead switch in interconnection network**
- **Availability is not the same as reliability**
  - **Reliability = probability that system is OK until time T**
  - **Reliability is useful metric for mission critical systems**

# Availability Motivation, part 1

- **Fault rates are increasing**
  - **More faults → more errors → more downtime → less availability**
- **Reason 1: Technology trends**
  - **Smaller transistors (e.g., thinner gate oxides)**
  - **Denser wires**
  - **Less charge on storage elements**
- **Reason 2: Architecture trends**
  - **More components**
  - **More aggressive designs**

# Availability Motivation, part 2

- **We're relying on computers more and more**
  - **Business**
  - **Education**
  - **Government**
- **High availability isn't just for NASA any more**
- **Can't afford to have unavailable services**

# Outline

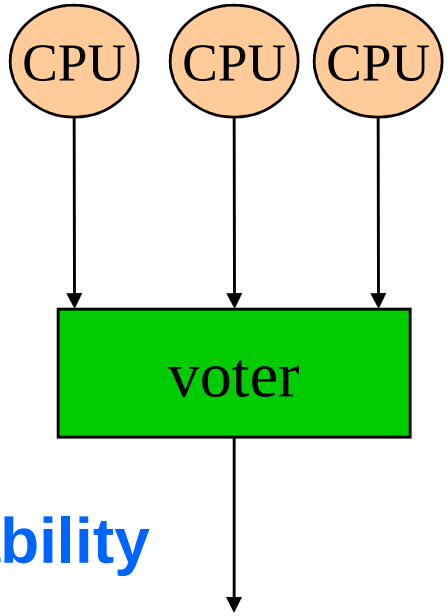
- **Definition and Motivation**
- **General Principles of Available System Design**
- **Traditional High-Availability Systems**
- **Recent Innovations in Available Systems**

# Goals

- System should correct errors as they occur
  - Or at least detect them
  - No incorrect outputs → system provides safety
  - In addition to safety, also want liveness
- Detecting an error and then crashing can be OK
  - Better than letting error propagate and corrupt vital data
- Error types
  - Transient: occurs once and disappears
    - » Bit flip on link due to cosmic ray impact
  - Intermittent: occurs off and on
    - » Bit flip on link due to loose connection
  - Permanent: occurs once and stays
    - » Bit on link stuck at one due to short circuit

# Forward Error Recovery

- **Use redundant hardware to mask faults**
  - **E.g., triple modular redundancy with voter or pair&spare**
- **Commonly used at many levels**
  - **ECC for memory and network links**
  - **RAID for disks**
  - **Redundant processors in mainframes**
- **E.g., IBM mainframes, Stratus**




**Sacrifices cost to achieve availability**

# Backward Error Recovery

- **If fault detected, recover system to pre-fault state**

**A. Periodically stop system and save state**

  - **Fault? Restore pre-fault recovery point checkpoint**

**B. Log all changes to system state**

  - **Fault? Unroll log to undo changes since recovery point**

- **E.g., Sequoia, Synapse N+1, Tandem NonStop**

**Sacrifices performance to achieve availability**

# BER and the Output Commit Problem

- **Problem: we can recover our system when we detect an error, but we can't recover the outside world**
  - **Output commit: can't undo effects of sending bad data out**
  - **Input commit: can't ask outside world to re-send us data**
- **Outside world (I/O)**
  - **Disks, network, printer, etc.**
- **Output commit solution: don't send data to outside world until we know it is error-free (yuck!)**
- **Input commit solution: log data received from outside world and replay it after recovery (not bad)**

# Outline

- **Definition and Motivation**
- **General Principles of Available System Design**
- **Traditional High-Availability Systems**
- **Recent Innovations in Available Systems**

# Traditional Fault Tolerance

- **Availability was most important aspect**
  - **Would sacrifice performance and/or hardware cost**
- **Usually simple implementations of FER and BER**
- **Used to be many companies making fault tolerant computer systems (besides IBM)**
  - **Tandem (bought by Compaq)**
  - **Synapse**
  - **Sequoia**
  - **Stratus**
  - **Sequent (bought by IBM)**
  - **Etc.**

# IBM Mainframes

- **The standard for high availability systems**
- **PRESENTATION**

# Outline

- **Definition and Motivation**
- **General Principles of Available System Design**
- **Traditional High-Availability Systems**
- **Recent Innovations in Available Systems**

# Recent Advances in Uniprocessor Availability

- Availability, but with higher performance & lower cost
- DIVA – dynamic verification of a microprocessor
  - Uses checker core to dynamically verify aggressive core
  - FER approach with little performance cost or hardware cost
- AR-SMT – uses redundant thread to check execution
  - Similar to using redundant processor, but cheaper (esp. for SMT)
  - No recovery mechanism specified, just detection
- But what about multiprocessors?
  - FER is tough, but BER can be made practical

# Dynamic Verification of SC

- **PRESENTATION**

# SafetyNet

- **PRESENTATION**

# Outline

- **Definition and Motivation**
- **General Principles of Available System Design**
- **Traditional High-Availability Systems**
- **Recent Innovations in Available Systems**