Commercial Fault Tolerance：A Tale of Two System

Wendy Bartlett, Member, IEEE Computer Society,
Lisa Spainhower, Member, IEEE

IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 1, NO. 1, JANUARY-MARCH 2004

Outline

- **Introduction**
- **Initial Fault Tolerance Philosophies**
- **Design Principles**
  - **zSeries**
  - **Nontop**
- **Advanced Design**
  - **zSeries**
- **Operating System**
- **Conclusion-Design Trade Offs**

Introduction

- This paper compares the design philosophies and implementations of two computer system – zSeries,Nontop
- Both systems serve commercial business for applications that require very high to continuous availability.

Introduction

- Initial Target Audiences , Ex. ATM , Point Of Sale.

- As businesses became more global and moved to 24 x 7 x forever operations, the demand for continuous operation became common.

Initial Fault Tolerance Philosophies

- Both system focused on providing fault tolerance through duplicate components and paths.

- Hardware module repair or upgrade, can be performed online.

Design Principles - zSeries

- Initially, support subsystems—power, cooling, service processor—were either duplicated.

- Later Fault tolerant enhancements were added with each new generation of CMOS.

Design Principles - zSeries

- L1 , L2 cache ,memory is protected by ECC
- For a permanent failure, a cache line or a memory line delete is performed dynamically.
- The I/O channel adapters perform direct memory access with robust memory protection

Design Principles - Nonstop

- Each processor had its own memory, an I/O bus, and ran its own copy of the operating system.

- If a processor or its I/O bus were to fail, the controllers would switch ownership to their backup paths.

Design Principles - Nonstop

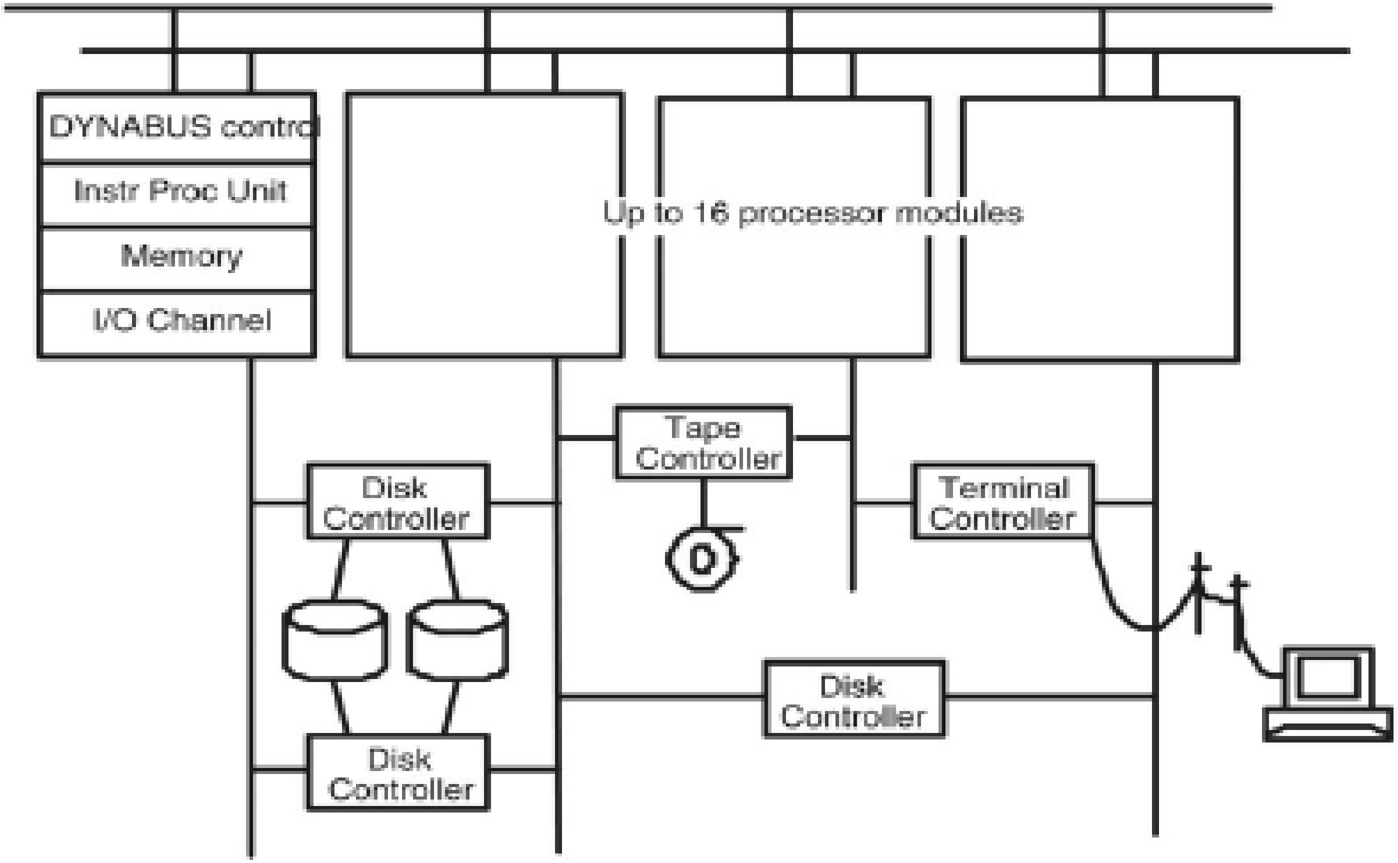
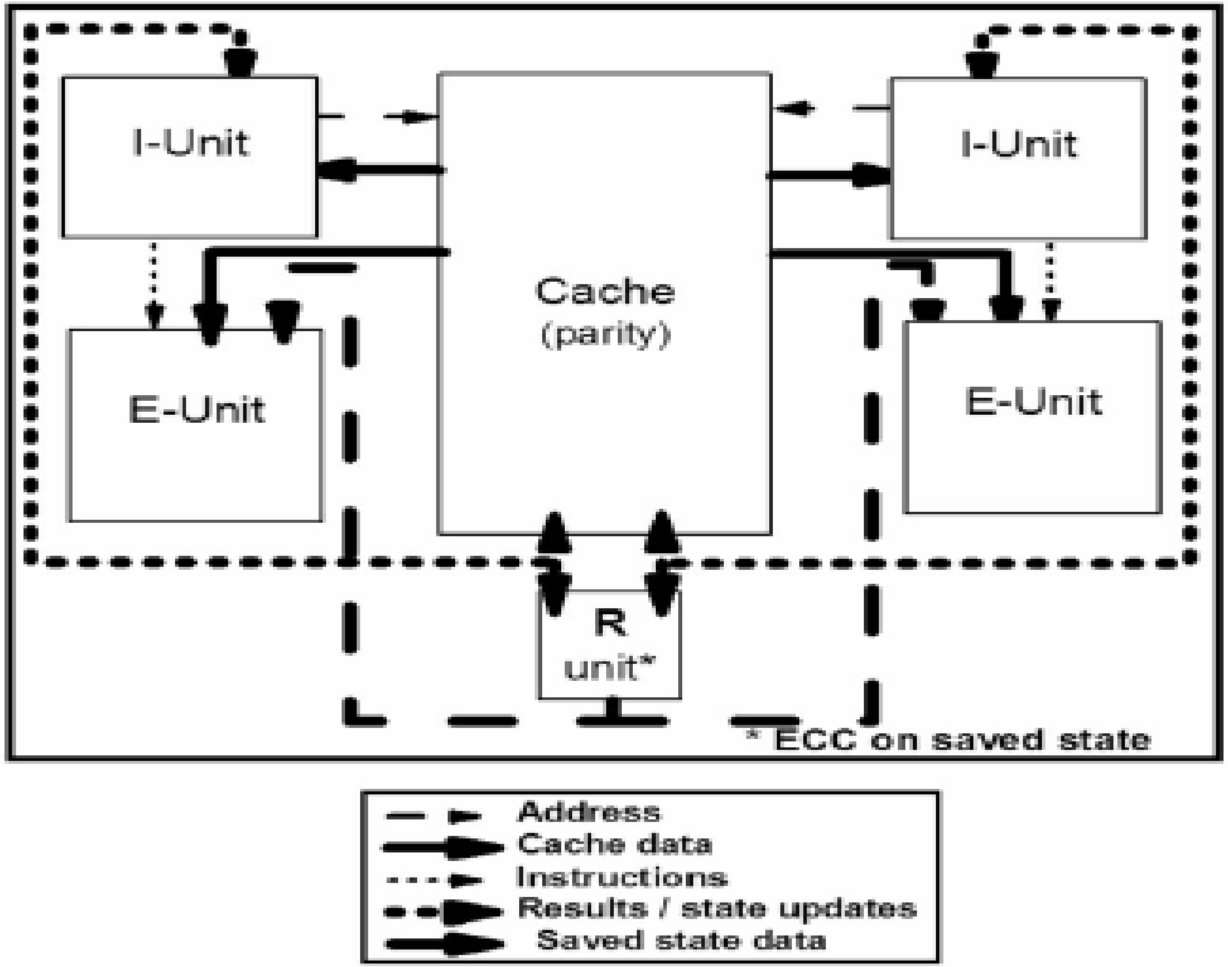


Fig. 2. Original tandem system architecture (1976).

Design Principles - Nonstop

- Software fault tolerance was built into the operating system.
- Each processor preconfigured with a small set of daemon-type processes – Nonstop kernel.
- There is a key software abstraction, Process Pairs.

Advanced Design - zSeries



OS - zSeries

- LPAR allows multiple operating system instances concurrently on one mainframe, OS could be different ,such as VM ,LINUX ,z/OS

- Extended Recovery Facility (XRF), which allowed a backup partition to be created within the same or a different mainframe.

OS – Nonstop

- “transaction processing monitor” was developed to meet this need by handling the distributed and fault-tolerant aspects of the work.
- Other fault tolerant example:
  - Ready list is doubly linked.
  - Disk process will be checkpointed.
  - Disk sector has checksum.

Conclusion-Design Trade Offs

- As a result, other than zSeries, today’s microprocessors leave control and arithmetic unit unchecked.
- However , There are advantages of building in a high level of integrity checking and retry/recovery logic to handle errors.