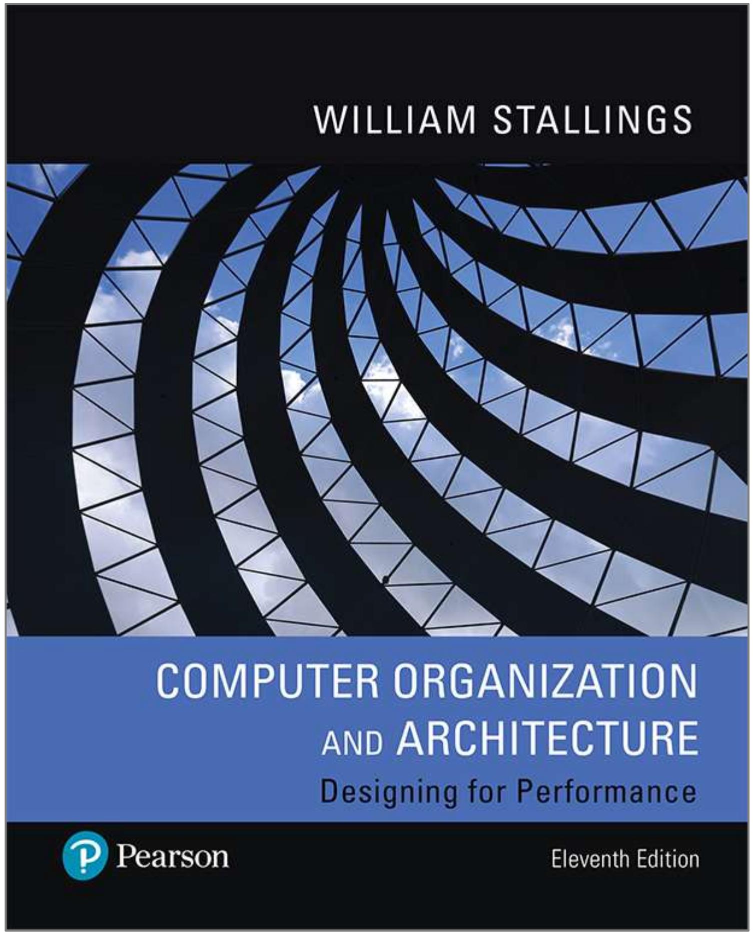
Computer Organization and Architecture
Designing for Performance

11th Edition



Chapter 4

The Memory Hierarchy:
Locality and Performance

Principle of Locality (1 of 2)

- Also referred to as the *locality of reference*
- Reflects the observation that during the course of execution of a program, memory references by the processor tend to cluster
- Locality is based on three assertions:
  - During any interval of time, a program references memory location non-uniformly
  - As a function of time, the probability that a given unit of memory is referenced tends to change slowly
  - The correlation between immediate past and immediate future memory reference patterns is high and tapers off as the time interval increases

Principle of Locality (2 of 2)

- Two forms of locality
  - Temporal locality
    - Refers to the tendency of a program to reference in the near future those units of memory referenced in the recent past
    - Constants, temporary variables, and working stacks are also constructs that lead to this principle
- Spatial locality
  - Spatial locality
    - Refers to the tendency of a program to reference units of memory whose addresses are near one another
    - Also reflects the tendency of a program to access data locations sequentially, such as when processing a table of data

Figure 4.1

Figure 4.1 Moving File Folders Between Smaller, Faster-Access Storage and Larger, Slower-Access Storage

Figure 4.2

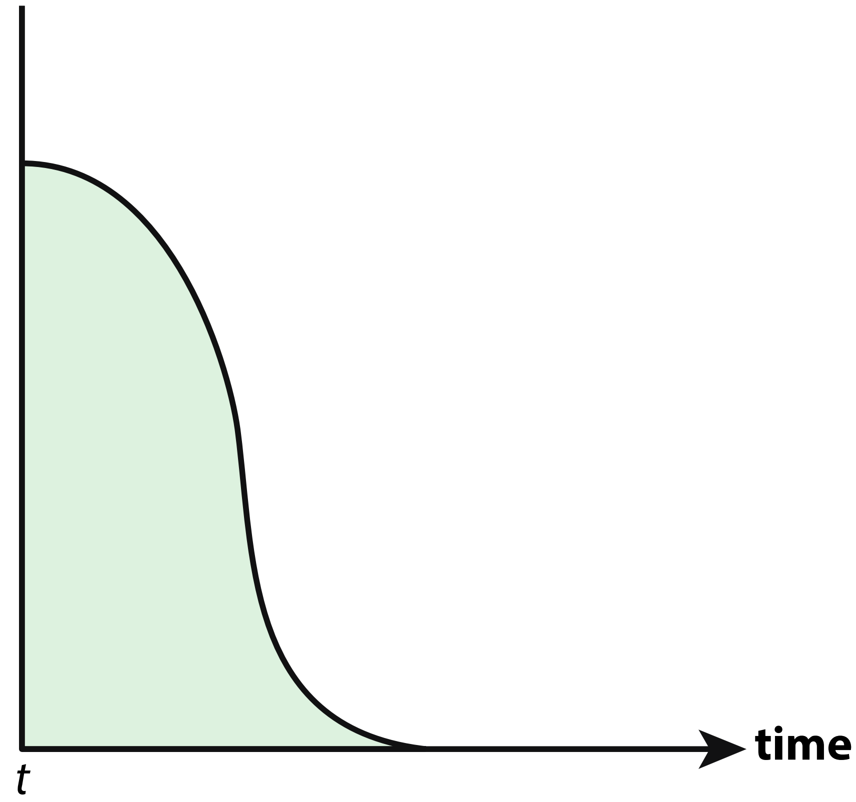


Figure 4.2 Idealized Temporal Locality Behavior: Probability Distribution for Time of Next Memory Access to Memory Unit Accessed at Time *t*

Figure 4.3

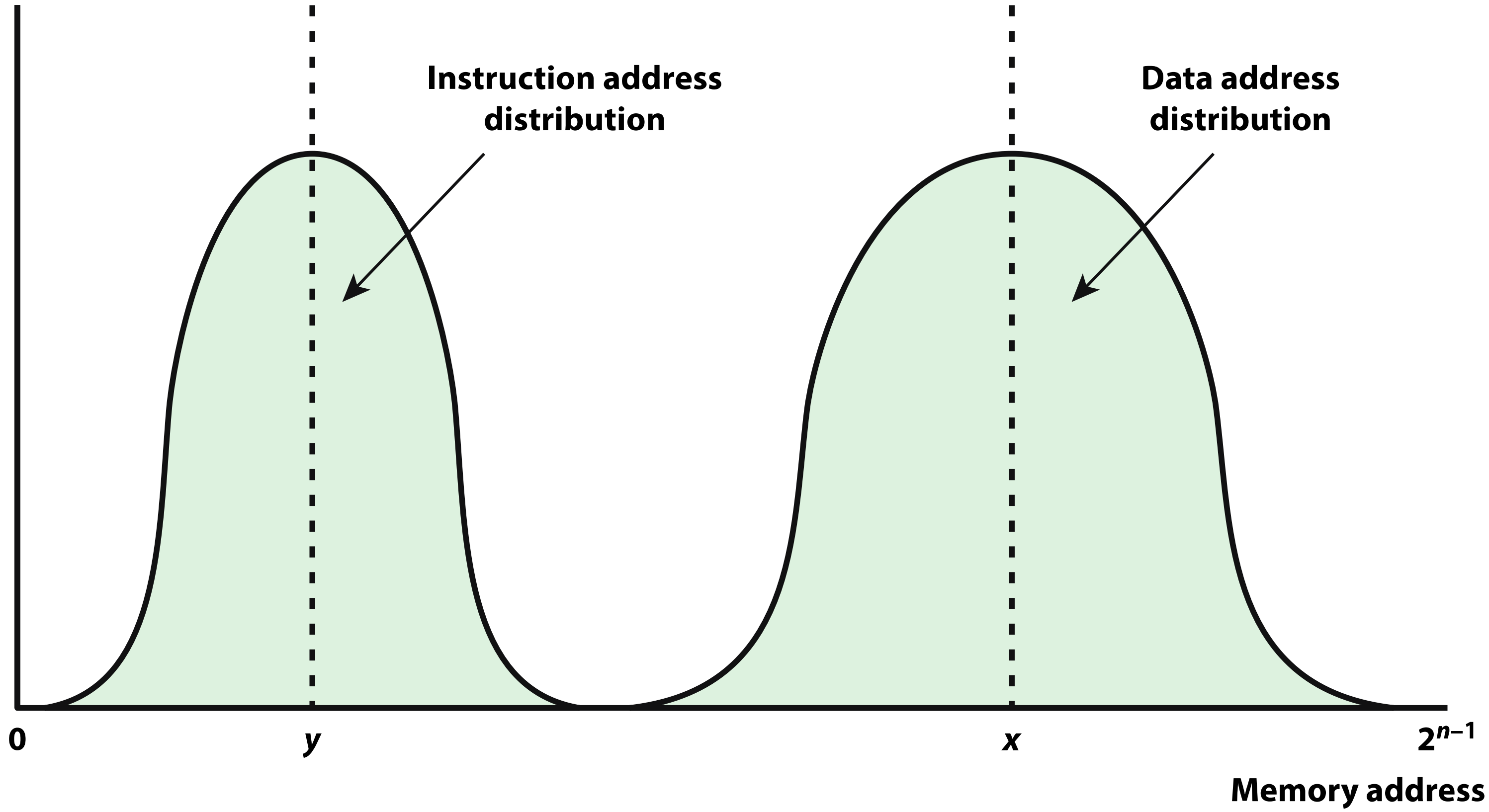


Figure 4.3 Idealized Spatial Locality Behavior: Probability Distribution for Next Memory Access (most recent data memory access at location *x*; most recent instruction fetch at location *y*)

Figure 4.4

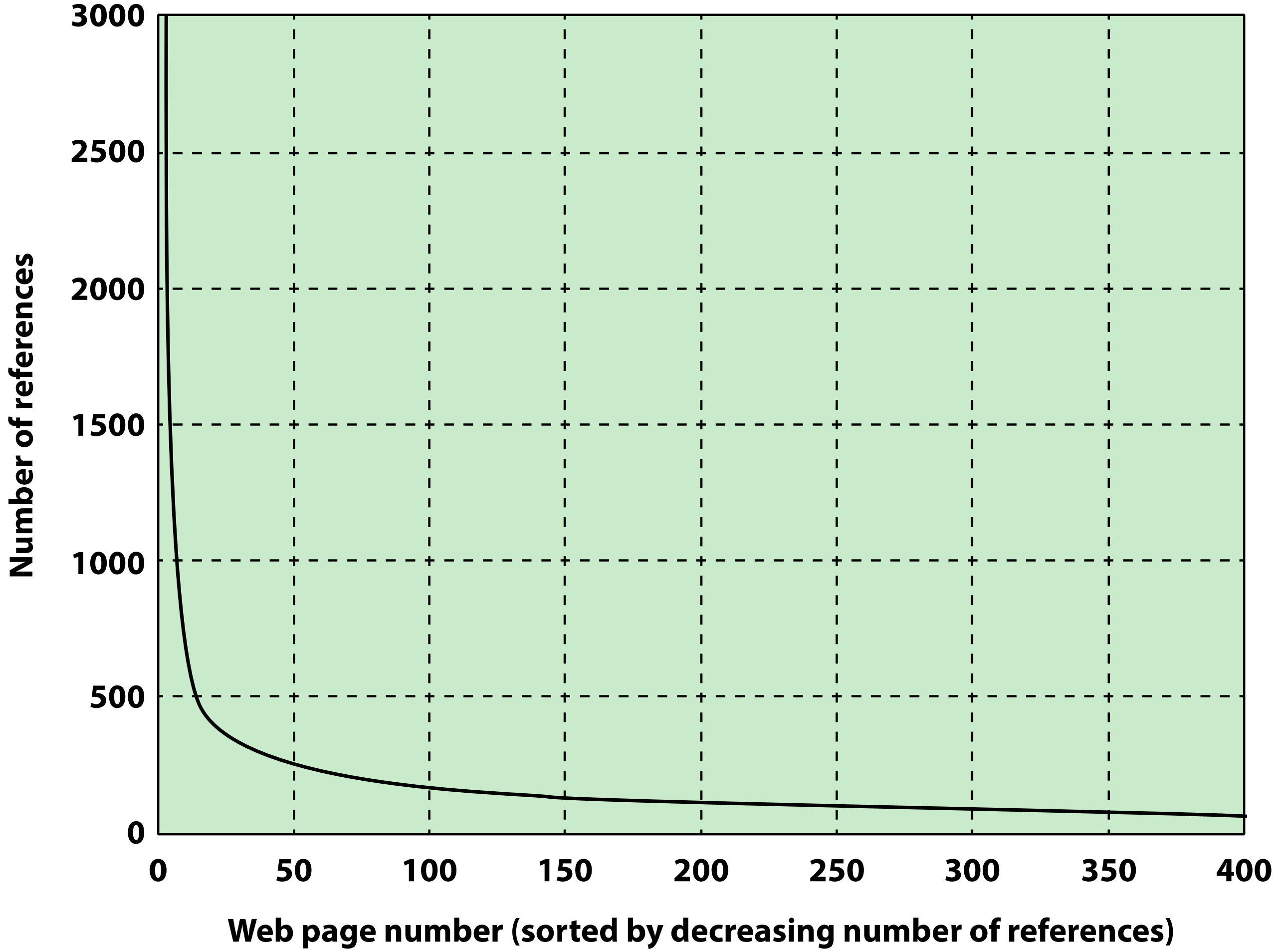


Figure 4.4 Data Locality of Reference for Web-Based Document Access Application

Figure 4.5

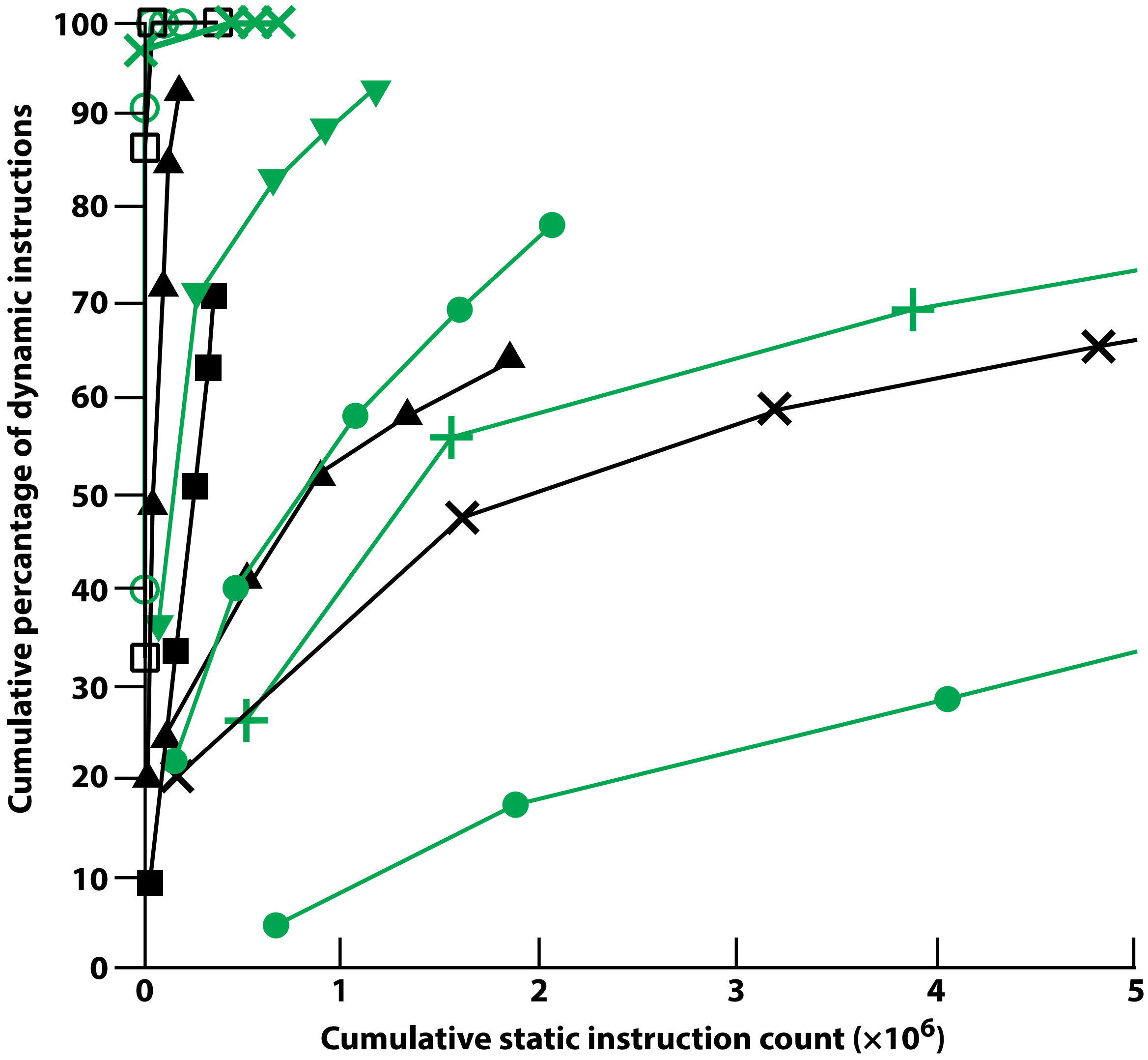


Figure 4.5 Instruction Locality Based on Code Reuse in Eleven Benchmark Programs in SPEC CPU2006

Table 4.1
Key Characteristics of Computer Memory Systems

Location Internal (e.g., processor registers, cache, main memory) External (e.g., optical disks, magnetic disks, tapes) **Capacity** Number of words Number of bytes **Unit of Transfer** Word Block **Access Method** Sequential Direct Random Associative	**Performance** Access time Cycle time Transfer rate **Physical Type** Semiconductor Magnetic Optical Magneto-optical **Physical Characteristics** Volatile/nonvolatile Erasable/nonerasable **Organization** Memory modules

Characteristics of Memory Systems

- Location
  - Refers to whether memory is internal and external to the computer
  - Internal memory is often equated with main memory
  - Processor requires its own local memory, in the form of registers
  - Cache is another form of internal memory
  - External memory consists of peripheral storage devices that are accessible to the processor via I/O controllers
- Capacity
  - Memory is typically expressed in terms of bytes
- Unit of transfer
  - For internal memory the unit of transfer is equal to the number of electrical lines into and out of the memory module

Method of Accessing Units of Data

Sequential access

- Memory is organized into units of data called records
- Access must be made in a specific linear sequence
- Access time is variable

Direct access

- Involves a shared read-write mechanism
- Individual blocks or records have a unique address based on physical location
- Access time is variable

Random access

- Each addressable location in memory has a unique, physically wired-in addressing mechanism
- The time to access a given location is independent of the sequence of prior accesses and is constant
- Any location can be selected at random and directly addressed and accessed
- Main memory and some cache systems are random access

Associative

- A word is retrieved based on a portion of its contents rather than its address
- Each location has its own addressing mechanism and retrieval time is constant independent of location or prior access patterns
- Cache memories may employ associative access

Pearson

Capacity and Performance:

The two most important characteristics of memory

Three performance parameters are used:

Access time (latency)

- For random-access memory it is the time it takes to perform a read or write operation
- For non-random-access memory it is the time it takes to position the read-write mechanism at the desired location

Memory cycle time

- Access time plus any additional time required before second access can commence
- Additional time may be required for transients to die out on signal lines or to regenerate data if they are read destructively
- Concerned with the system bus, not the processor

Transfer rate

- The rate at which data can be transferred into or out of a memory unit
- For random-access memory it is equal to 1/(cycle time)

Memory

- The most common forms are:
  - Semiconductor memory
  - Magnetic surface memory
  - Optical
  - Magneto-optical
- Several physical characteristics of data storage are important:
  - Volatile memory
    - Information decays naturally or is lost when electrical power is switched off
  - Nonvolatile memory
    - Once recorded, information remains without deterioration until deliberately changed
    - No electrical power is needed to retain information
  - Magnetic-surface memories
    - Are nonvolatile
  - Semiconductor memory
    - May be either volatile or nonvolatile
  - Nonerasable memory
    - Cannot be altered, except by destroying the storage unit
    - Semiconductor memory of this type is known as read-only memory (ROM)
- For random-access memory the organization is a key design issue
  - Organization refers to the physical arrangement of bits to form words

Memory Hierarchy

- Design constraints on a computer’s memory can be summed up by three questions:
  - How much, how fast, how expensive
- There is a trade-off among capacity, access time, and cost
  - Faster access time, greater cost per bit
  - Greater capacity, smaller cost per bit
  - Greater capacity, slower access time

Figure 4.6

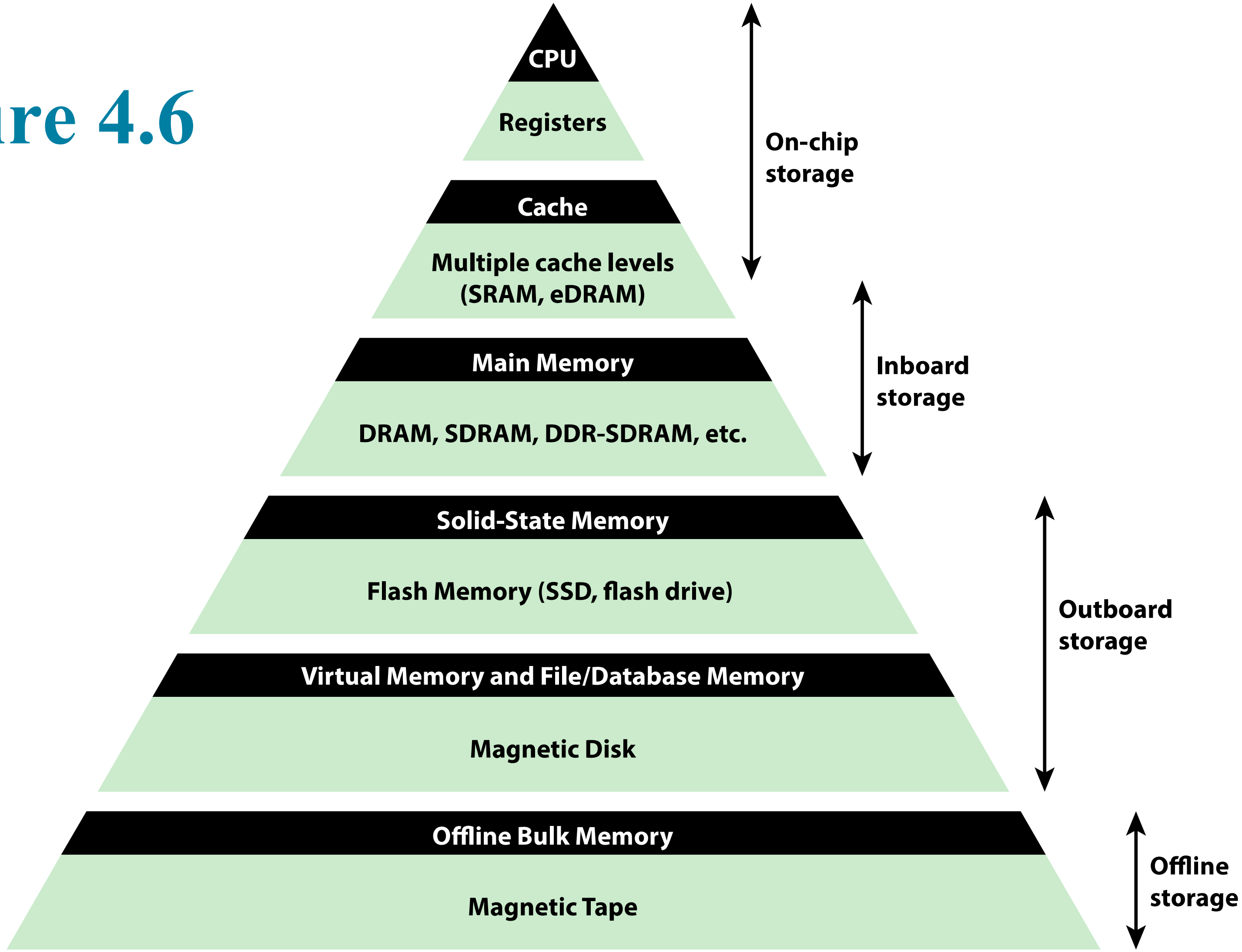


Figure 4.6 The Memory Hierarchy

Figure 4.7

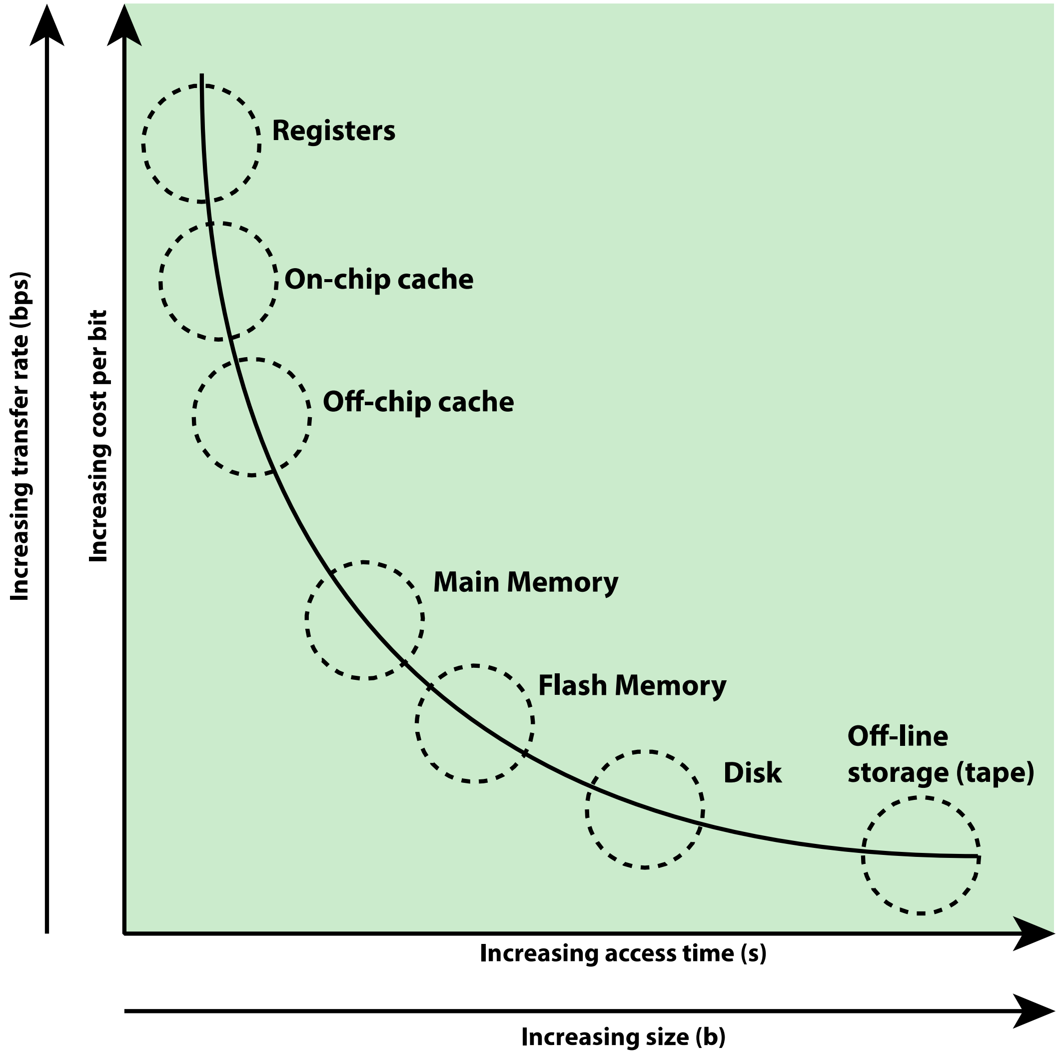


Figure 4.7 Relative Cost, Size, and Speed Characteristics Across the Memory Hierarchy

Figure 4.8

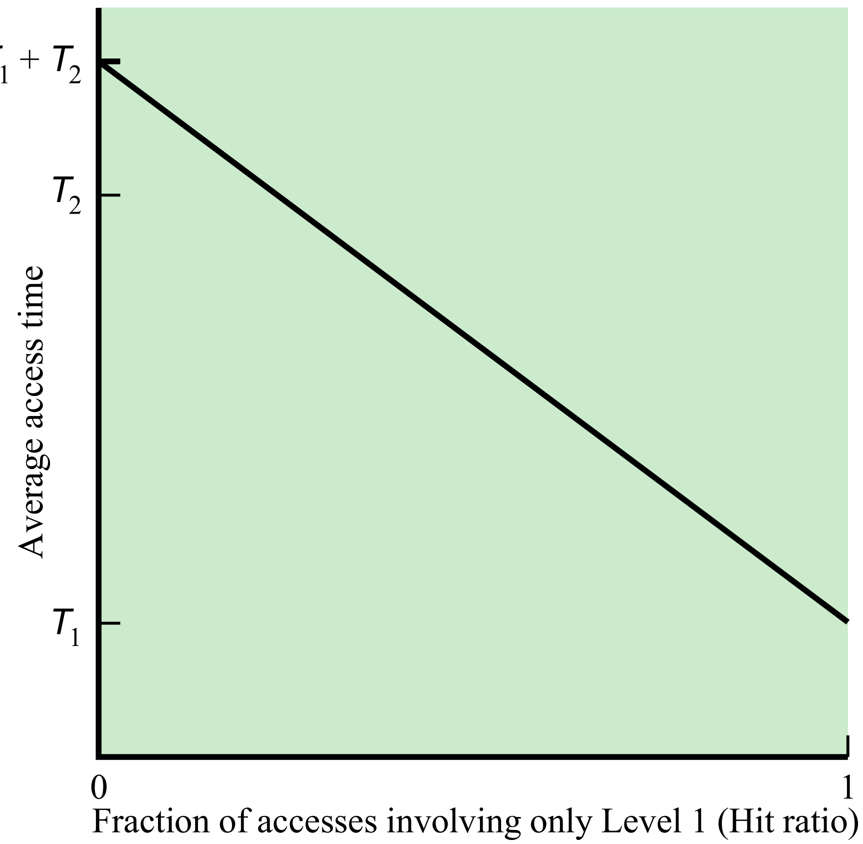


Figure 4.8 Performance of a Simple Two-Level Memory

Figure 4.9

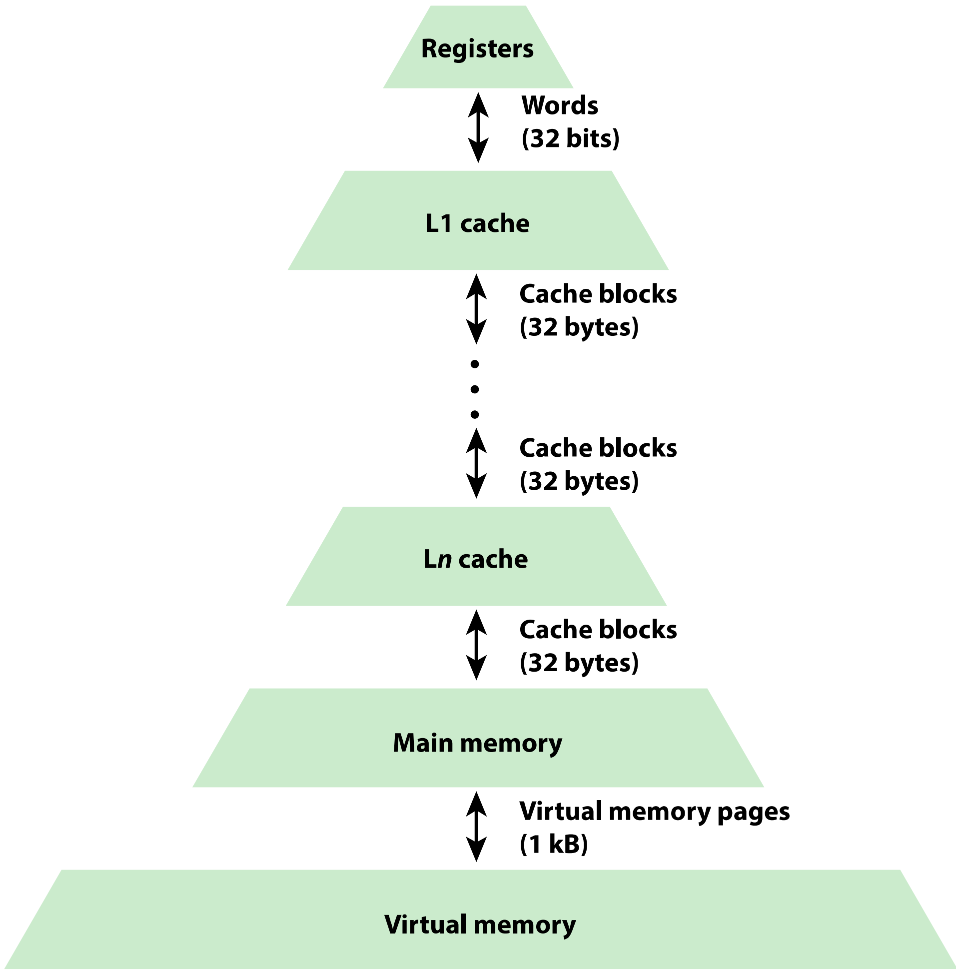


Figure 4.9 Exploiting Locality in the Memory Hierarchy (with typical transfer size)

Table 4.2
Characteristics of Memory Devices in a Memory Architecture

Memory level	Typical technology	Unit of transfer with next larger level (typical size)	Managed by
Registers	CMOS	Word (32 bits)	Compiler
Cache	Static RAM (SRAM); Embedded dynamic RAM (eDRAM)	Cache block (32 bytes)	Processor hardware
Main memory	DRAM	Virtual memory page (1 kB)	Operating system (OS)
Secondary memory	Magnetic disk	Disk sector (512 bytes)	OS/user
Offline bulk memory	Magnetic tape		OS/User

Table 4.2 Characteristics of Memory Devices in a Memory Architecture

Memory

- The use of three levels exploits the fact that semiconductor memory comes in a variety of types which differ in speed and cost
- Data are stored more permanently on external mass storage devices
- External, nonvolatile memory is also referred to as **secondary** memory or **auxiliary** memory

Figure 4.10

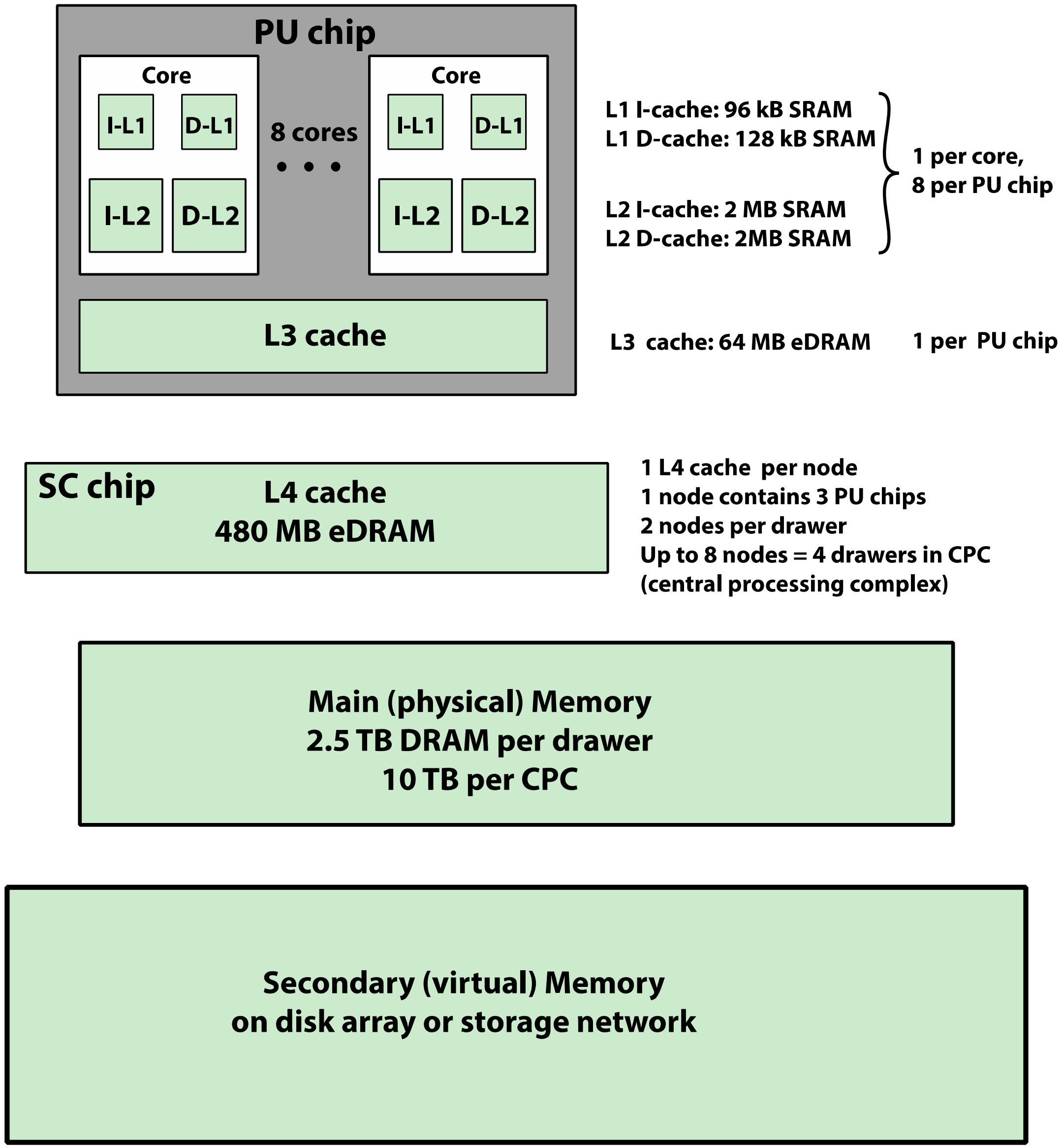


Figure 4.10 IBM z13 Memory Hierarchy

Design Principles for a Memory Hierarchy

Locality

The principle that makes effective use of a memory hierarchy possible

Inclusion

This principle dictates that all information items are originally stored in level M_n where n is the level most remote from the processor

Coherence

Copies of the same data unit in adjacent memory levels must be consistent

If a word is modified in the cache, copies of that word must be updated immediately or eventually at all higher levels

Two-Level Memory Access

- A cache acts as a buffer between main memory and processor, creating a two-level internal memory
- Exploits locality to provide improved performance over a comparable one-level memory
- The main memory cache mechanism is part of the computer architecture, implemented in hardware and typically invisible to the operating system
- Two other instances of a two-level memory approach that also exploit locality and that are, at least partially, implemented in the operating system are virtual memory and the disk cache

Operation of Two-Level Memory

- The locality property can be exploited in the formation of a two-level memory
- The upper-level memory (M1) is smaller, faster, and more expensive (per bit) than the lower-level memory (M2)
- M1 is used as temporary store for part of the contents of the larger M2
- When a memory reference is made, an attempt is made to access the item in M1
  - If this succeeds, then a quick access is made
  - If not, then a block of memory locations is copied from M2 to M1 and the access then takes place via M1
- Because of locality, once a block is brought into M1, there should be a number of accesses to locations in that block, resulting in fast overall service

Figure 4.11

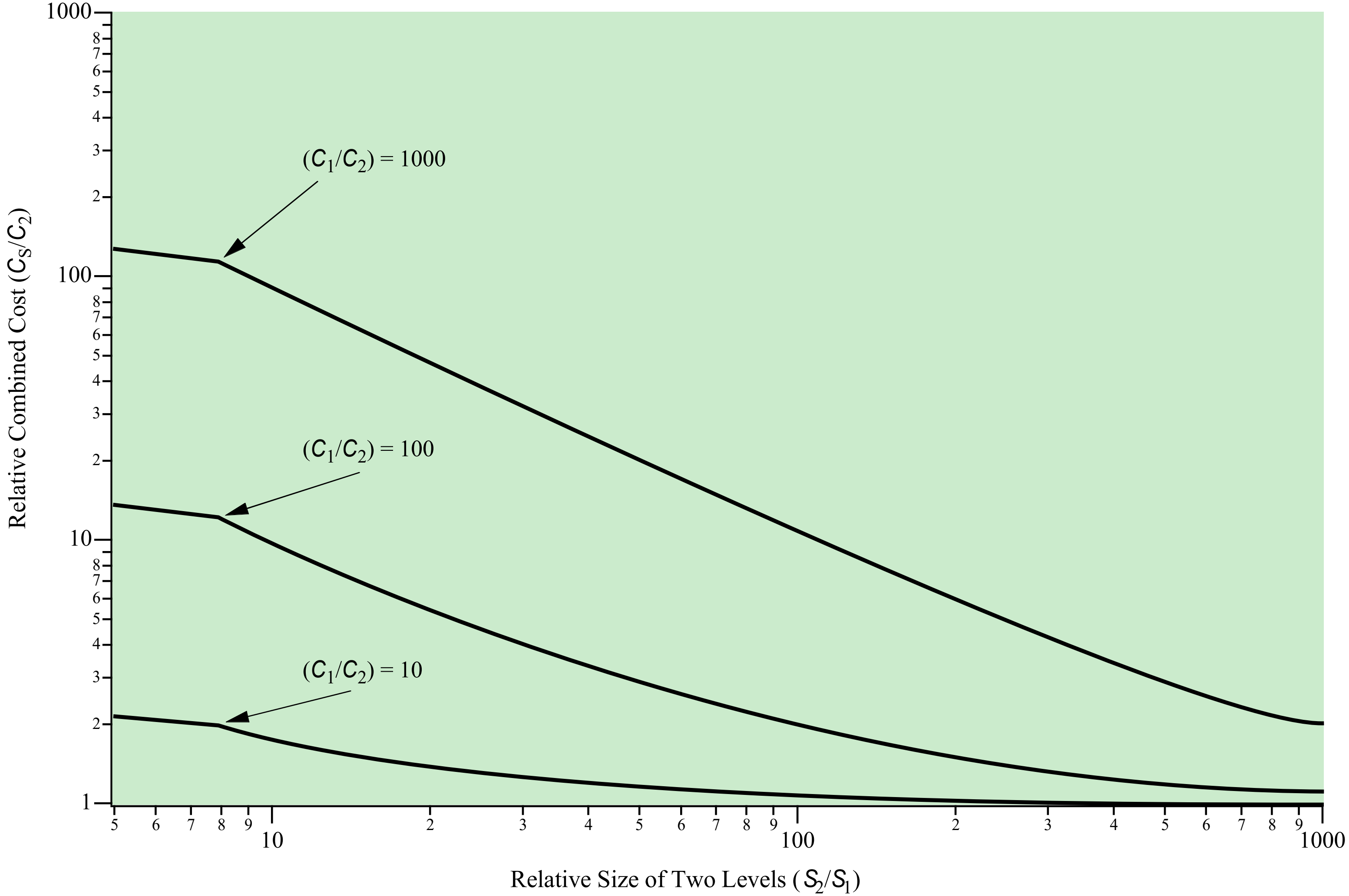


Figure 4.11 Relationship of Average Memory Cost to Relative Memory Size for a Two-Level Memory

Figure 4.12

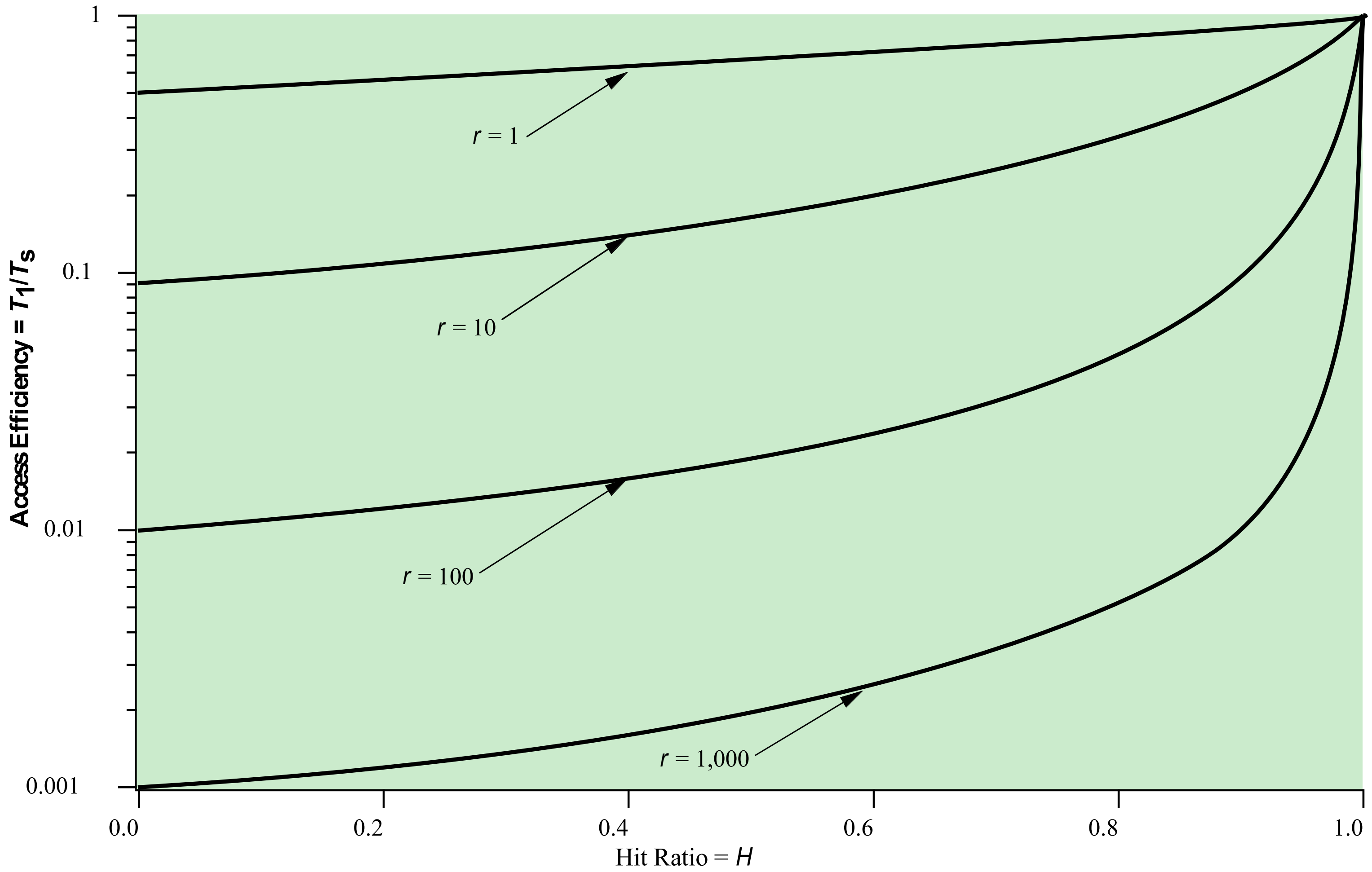


Figure 4.12 Access Efficiency as a Function of Hit Ratio ($r = T_2/T_1$)

Figure 4.13

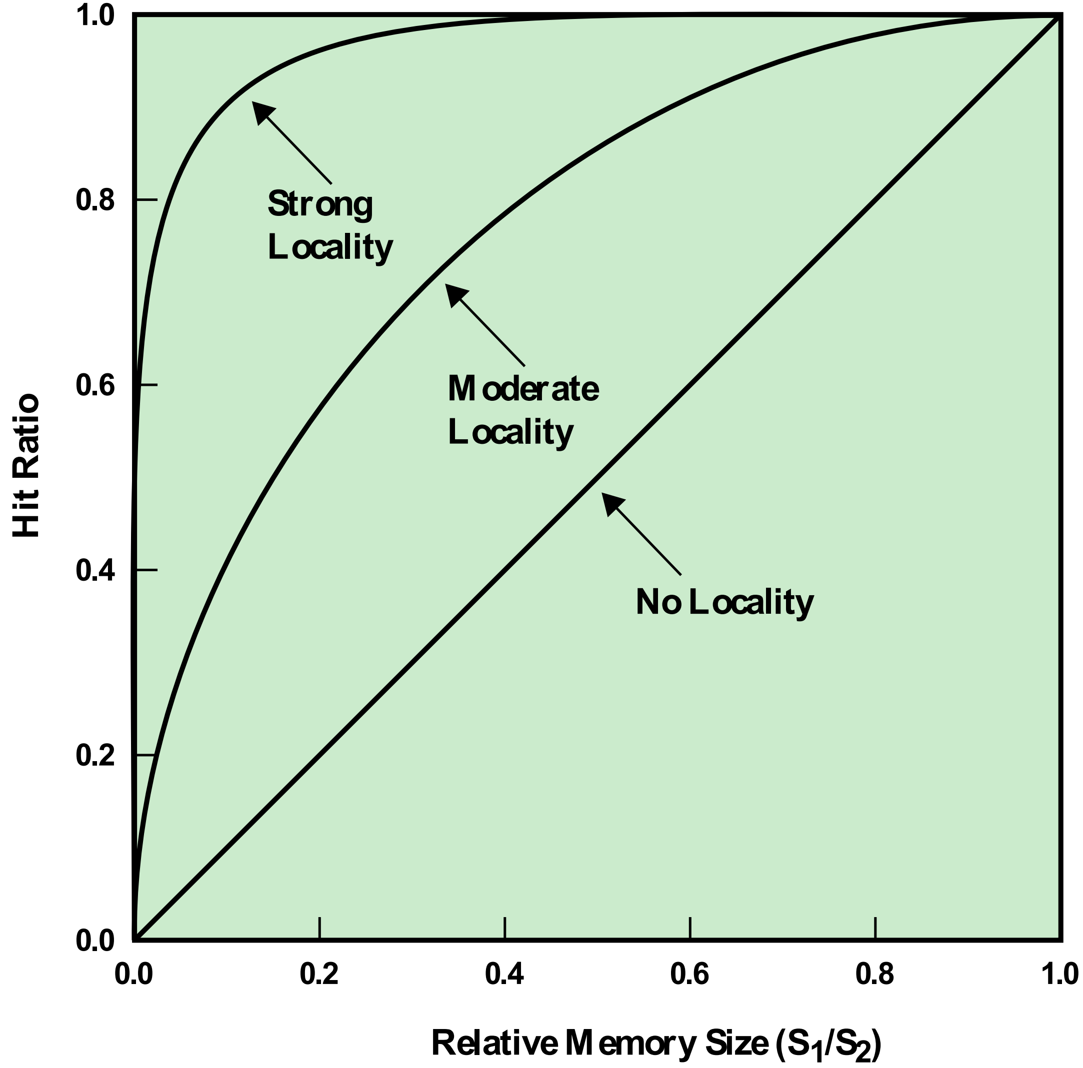


Figure 4.13 Hit Ratio as a Function of Relative Memory Size

Pearson

Figure 4.14

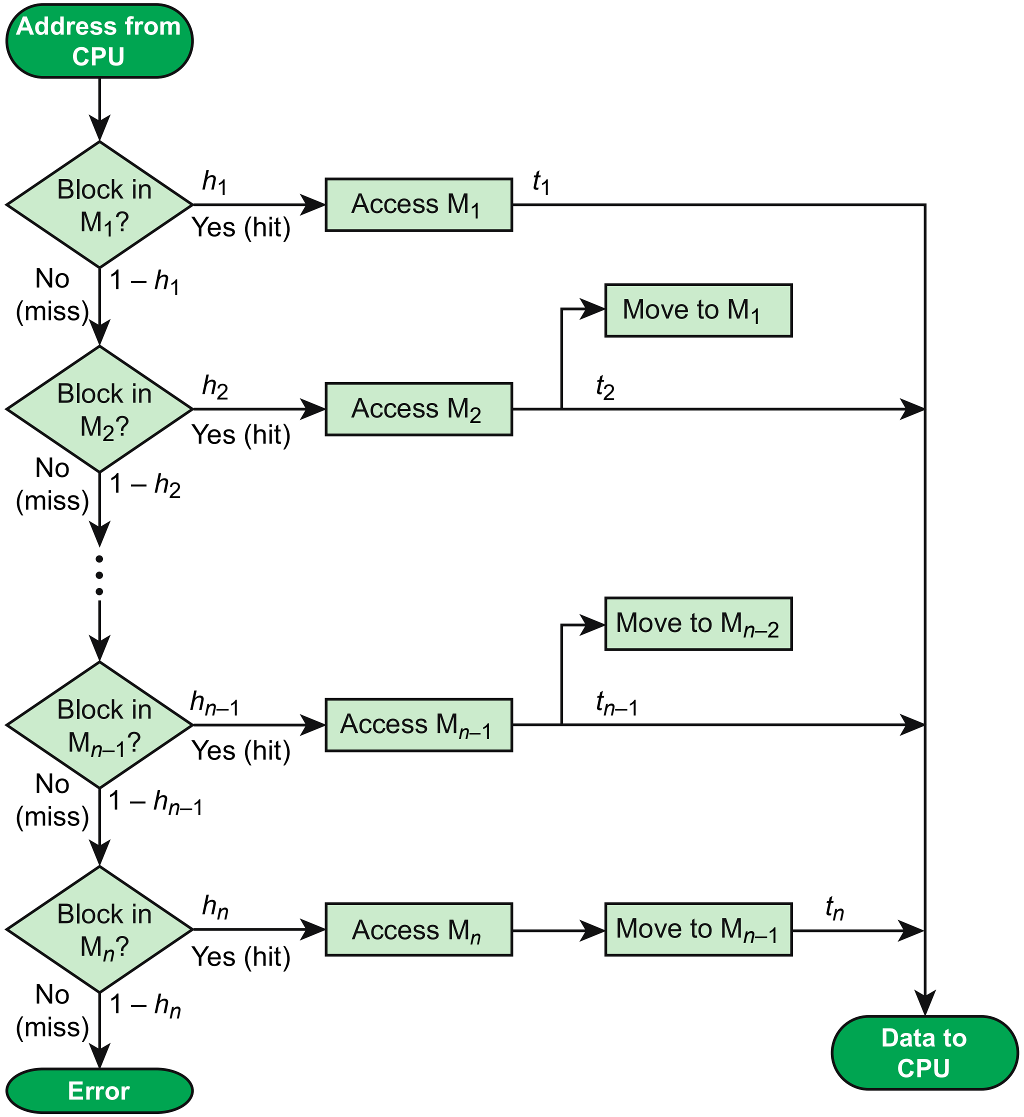


Figure 4.14 Multilevel Memory Access Performance Model

Summary

Chapter 4

- Principle of locality
- Characteristics of memory systems
- Performance modeling of a multilevel memory hierarchy
  - Two-level memory access
  - Multilevel memory access

The Memory Hierarchy: Locality and Performance

- The memory hierarchy
  - Cost and performance characteristics
  - Typical members of the memory hierarchy
  - The IBM z13 memory hierarchy
  - Design principles for a memory hierarchy

Copyright