Advanced Topics

Chapter 18

Pipelining

Topics

- Introduction
- Concept of Pipelining
- Pipeline Characteristics
- Software Pipelining
- Software Pipeline Performance And Overhead
- Hardware Pipeline
- Instruction and Data Pipeline
- Hardware Pipeline and Performance Improvement

Topics

- When Pipelining Can Be Used ?
- Pipeline vs Parallel
- Pipeline Architectures
- Pipeline Setup, Stall, and Flush Times
- Superpipeline Architecture

Introduction

- Recall
  - Earlier parts: on processors, memory, I/O as basic functional units
  - Last chapter: parallelism to increase performance
  - Now: pipelining to increase performance

- Chapter covers
  - Motivation
  - Ways of using pipelining
  - How pipelining increases performance?

Concept of Pipelining

- An architecture in which digital information flows through a series of processing components
- Pipeling is a concept used in several situations
- Software pipelining exists

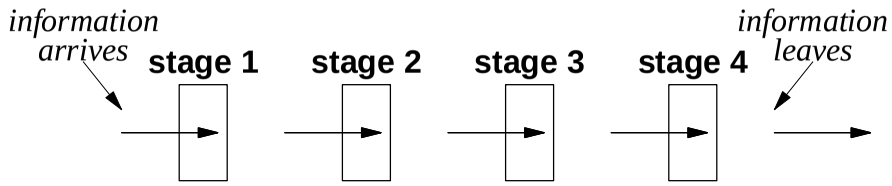


Illustration of the pipeline concept. The example has four stages, and information flows through each stage.

Pipeline Characteristics

- Hardware or software implementation
  - pipelining can be used with hardware or software
  - e.g. of software pipeline: Unix pipe command
- Large or small scale
  - stations can be simple or powerful
  - stages can be few or many

Pipeline Characteristics

- Synchronous or asynchronous communication
  - synchronous is like an assembly, all stations simultaneously move data forward
  - asynchronous allows forwarding at any time; downside stalling.
- Buffered or unbuffered pipelines
  - buffer between stages, helpful with asynchronous pipeline

Pipeline Characteristics

- Finite chunks or continuous bit streams
  - finite chunks e.g. Packets
  - automatic data feed or manual data feed
  - e.g. separate mechanism to move information
- Serial or parallel path
  - between stages data can be moved in single serial path or simultaneously in parallel
- Homogenous or heterogeneous stages
  - heterogeneous: Use hardware suitable for each stage

Software Pipelining

- Allows large complex task to be broken to smaller pieces
- Use Unix pipe "|" command to feed output of one command as input to another command
- Software pipelining does not need underlying hardware pipeline
- Does software pipeline perform better or worse than a single program?
  - depends
  - software pipeline is used to reduce complexity but in certain cases it can improve performance

Hardware Pipelining

- Can reduce complexity, break complex task to smaller manageable pieces
- Maybe able to reuse pieces other designs
- Hardware pipelining offers higher performance

Instruction and Data Pipeline

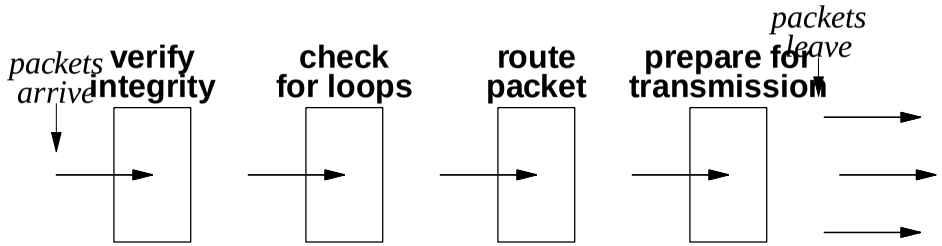
- Instruction pipeline
  - machine instructions are executed using a pipeline,
  - each stage is responsible for a part of the overall fetch-decode-execute cycle most modern processors have instruction
  - pipeline

- Data pipeline
  - data is passed in pipeline, stage to stage.

Hardware Pipelining and Performance Improvement

A data pipeline passes through a series of stages that each examine or modify the data. If it uses the same speed processors as a non-pipeline architecture, a data pipeline will not improve the overall time needed to process a given data item

Even if a data pipeline uses the same speed processors as a non-pipeline architecture, a data pipeline has higher overall throughput (i.e. number of data items processed per second

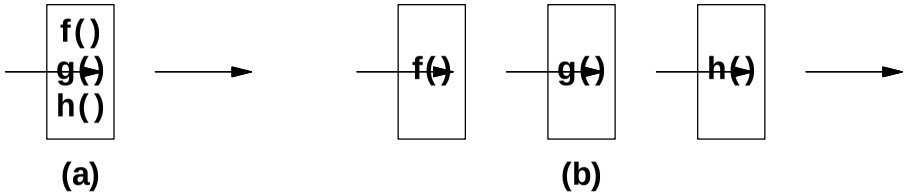


A pipeline used in place of a single processor in an Internet router.

When Pipelining Can Be Used ?

- It must be possible to partition processing into independent stages
- Overhead to move data between stages must be insignificant.
- Processing performed at each stage must take approximately the same
  time as the processing performed at other stages
- Throughput of pipeline is limited by the slowest stage

Pipeline vs Parallel



(a) Processing on a conventional processor, and (b) equivalent processing in a data pipeline. The functions performed in sequence are divided among stages of the pipeline.

Pipeline vs Parallel

- Pipeling divides a series of sequential operations into separate groups that are each handled by a separate stage of the pipeline.
- Pipeline allows each stage to operate in parallel.
- However, unlike a parallel architecture, data has to pass all stages

Pipeline Architectures

- The term pipeline architecture is used when it is the central paradigm around which the system is built.
- Pipelined systems re dedicated to special purpose functions, e.g. network system where high data rates are used.
- General purpose computers restrict pipeline hardware to instruction pipeline in the processor or special purpose pipeline in an I/O device.
  - not many applications can be decomposed into independent operations that can be applied sequentially.

Pipeline Setup, Stall, and Flush Times

- Setup time
  - amount of time to start a pipeline after an idle period
- Flush time
  - amount of time between input being unavailable and the pipeline finishes current processing
- Stalls
  - stage delays because it cannot complete processing

Superpipeline Architecture

- Given pipeline stage is subdivided into multiple partial stages
- Example instruction pipeline has
  - instruction fetch
  - instruction decode
  - operand fetch
  - ALU operation
  - store

Superpipeline Architecture

- The third stage, i.e. operand fetch can be subdivided into
  - decode operand
  - fetch immediate value or register value
  - fetch value from memory
  - fetch indirect operand values
- Superpipeling increases throughput

Summary

- Pipelining is a fundamental concept that is used with both hardware and software.
- Data pipeline does not decrease overall time to process a single data item, it increases overall throughput (items processed/second)
- The pipeline stage requiring the most time to process an item limits the throughput of the pipeline.