Architektura komputerów

Wykład 13

Jan Kazimirski

KOMPUTERY RÓWNOLEGŁE

Klasyfikacja systemów komputerowych

- **SISD** – Single Instruction, Single Data stream
- **SIMD** – Single Instruction, Multiple Data stream
- **MISD** – Multiple Instruction, Single Data stream
- **MIMD** – Multiple Instruction, Multiple Data stream

SISD

- Architektura złożona z jednego procesora
- Procesor przetwarza jeden strumień danych
- Dane przechowywane w pojedynczej pamięci
- Przykład – klasyczny komputer

SIMD

- Jedna instrukcja maszynowa na wielu elementach liczących.
- Każdy element liczący ma przypisany własny strumień danych.
- Przykłady
  - procesory wektorowe – stosowane w superkomputerach
  - rozszerzenia wektorowe (MMX,SSE)
  - akceleratory graficzne, procesory DSP

MISD

- Pojedynczy strumień danych
- Strumień danych przekazywany do elementów wykonawczych
- Elementy wykonawcze wykonują osobne sekwencje instrukcji na strumieniu danych
- Architektura rzadko spotykana
  - Czasami stosowana w krytycznych systemach (redundancja obliczeń i weryfikacja wyników)

MIMD

- Zbiór elementów przetwarzających.
- Każdy element przetwarzający realizuje swoją sekwencję instrukcji.
- Każdy element przetwarzający pracuje na osobnym strumieniu danych.
- Przykłady – większość współczesnych architektur.

MIMD - realizacja

- Systemy z pamięcią dzieloną
  - Systemy SMP (symetryczna wieloprocesorowość)
  - Architektura NUMA (Non Uniform Memory Access)
- Systemy rozproszone
  - Klastry
  - Chmury obliczeniowe
  - Środowiska gridowe

Systemy SMP

- Dwa lub więcej CPU o takiej samej funkcjonalności.
- CPU mają dostęp do wspólnej pamięci i przestrzeni I/O.
- Komunikacja między CPU odbywa się poprzez pamięć.
- Dostęp do danych nie zależy od CPU i położenia w pamięci.

Systemy SMP

- Zalety:
  - Szybka komunikacja poprzez pamięć dzieloną
  - Możliwość rozkładania obciążenia w dowolny sposób (procesory mogą się zastępować)
  - Odporność na awarie (uszkodzenie CPU)
  - Skalowalność – dodawanie kolejnych CPU zwiększa wydajność, dostosowanie wydajności do potrzeb.

Realizacja SMP

- Wspólna szyna.
  - Procesory, pamięć i moduły I/O podłączone do wspólnej szyny.
- Pamięć wieloportowa.
  - Niezależny dostęp każdego procesora i moduły I/O do pamięci.
- Centralna jednostka sterująca.
  - Dodatkowy układ sterujący komunikacją między procesorami, modułami I/O i pamięcią.

SMP – Wspólna szyna

- Wszystkie układy (procesory, moduły I/O, pamięć) podłączone do jednej szyny.
- Zalety
  - Prostota
  - Architektura podobna do architektury z jednym CPU
  - Skalowalność poprzez dołączanie nowych CPU
  - Odporność na awarie.

SMP – Wspólna szyna c.d.

- Wady:
  - Wydajność ograniczona przepustowością i sposobem zarządzania wspólną szyną.
  - Częste dostępy do pamięci obniżają wydajność – konieczność stosowania pamięci podręcznych.
  - Pamięci podręczne wprowadzają kolejny poważny problem – spójność danych w pamięciach podręcznych procesorów.

SMP – Pamięć wieloportowa

- Specjalnie zaprojektowana pamięć pozwala na bezpośrednie podłączenie wielu układów (CPU i moduły I/O).
- Zalety:
  - Duża wydajność.
- Wady:
  - Złożoność pamięci (dodatkowa logika).
  - Problemy z zachowaniem spójności pamięci podręcznej.

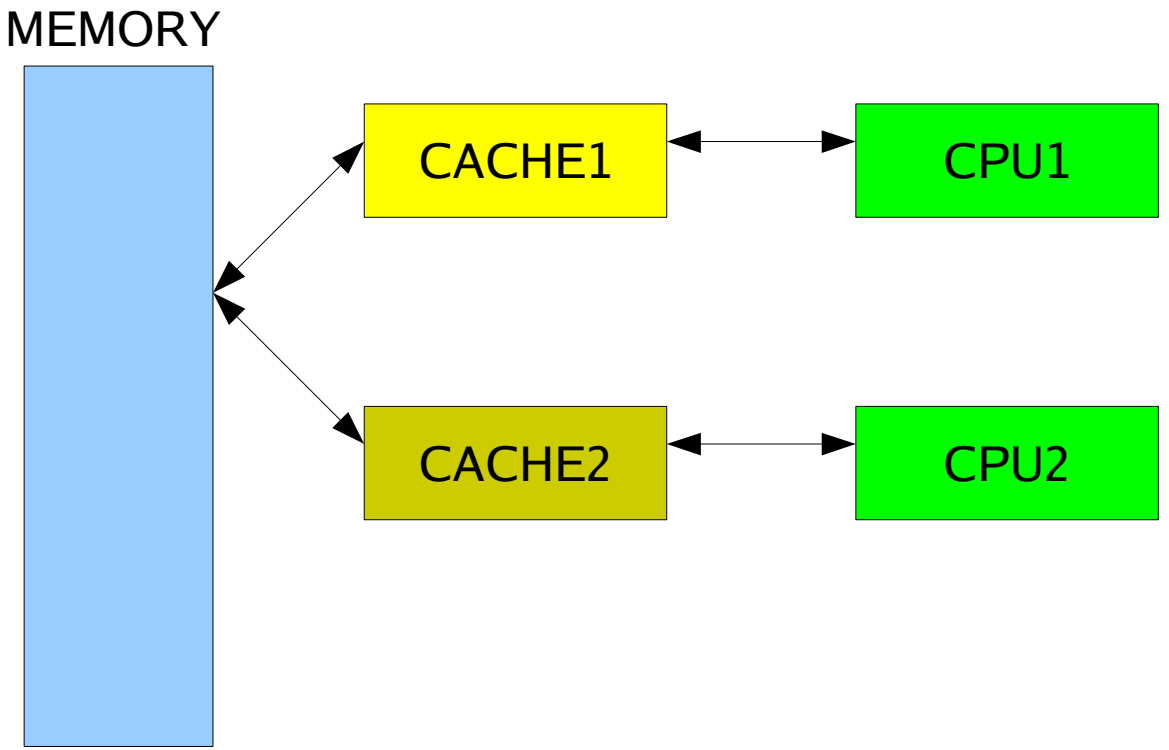
SMP – Centralna jednostka sterująca

- Specjalny kontroler steruje komunikacją między pamięcią a procesorami i modułami I/O
- Wady:
  - Złożona logika kontrolera
  - Kontroler stanowi „wąskie gardło” systemu
- Rozwiązanie było stosowane w komputerach klasy mainframe. Obecnie rzadko spotykane.

Spójność pamięci podręcznej

- Częste odwołania do pamięci głównej zmniejszają wydajność.
- Efekt ten jest zwielokrotniony w systemach SMP.
- W architekturze SMP krytycznego znaczenia nabiera szybka pamięć podręczna lokalna dla danego CPU
- **Problem: W lokalnych pamięciach podręcznych różnych procesorów mogą być kopie tych samych danych**

Spójność pamięci podręcznej c.d.



Procesory CPU1 i CPU2 przechowują w swoich pamięciach podręcznych kopie tych samych danych.

Modyfikacje dokonywane przez jeden z procesorów (na kopii w swojej pamięci podręcznej) nie są widziane przez drugi procesor.

Rozwiązania programowe

- Uniknięcie problemu spójności pamięci cache należy do systemu operacyjnego i kompilatora
  - np. wymuszanie natychmiastowej synchronizacji wszystkich danych współdzielonych z pamięcią główną.
- Rozwiązania programowe są zwykle dosyć konserwatywne i obniżają wydajność pamięci podręcznej.

Rozwiązania sprzętowe

- Sprzętowe mechanizmy zachowania spójności pamięci podręcznej
- Zalety:
  - Problemy rozwiązywane w momencie powstania
  - Efektywne wykorzystanie pamięci podręcznej
  - Rozwiązanie niewidoczne dla programu
- Wada: dodatkowe mechanizmy sprzętowe

Rozwiązania sprzętowe c.d.

- Strategie utrzymania spójności pamięci podręcznej (cache coherence protocols)
  - Wykorzystanie centralnego katalogu bloków pamięci w pamięci podręcznej
  - „Podsłuchiwanie” (snooping) wszystkich transferów na magistrali

Katalog bloków pamięci

- Informacje o blokach w pamięciach podręcznych trzymane są w centralnym katalogu.
- Centralny katalog przechowywany jest w pamięci głównej.
- Dodatkowy kontroler weryfikuje żądania przesłań między pamięcią główną i pamięciami podręcznymi oraz utrzymuje spójny stan pamięci.

„Podsłuchiwanie” magistrali

- Odpowiedzialność za spójność danych spada na kontroler pamięci podręcznej.
- Kontroler pamięci podręcznej „podsłuchuje” magistralę.
- Stosowane strategie:
  - Write Invalidate
  - Write Update

Write Invalidate

- Problem typu: „wielu czytelników, jeden pisarz”.
- Blok danych może istnieć w wielu kopiach w pamięciach podręcznych.
- Modyfikacja bloku danych powoduje unieważnienie innych kopii danych.
- Przykład realizacji: protokół MESI (Modified, Exclusive, Shared, Invalid)

Write Update

- Problem typu „wielu czytelników i pisarzy”.
- Blok danych może istnieć w wielu kopiach w pamięciach podręcznych.
- Modyfikacje bloku danych w pamięci podręcznej są propagowane do innych pamięci podręcznych.
- Niektóre architektury wykorzystują obie strategie (write invalidate i write update).

Architektura UMA i NUMA

- Architektura UMA – Unified Memory Access.
  - Wszystkie procesory mają dostęp do całej pamięci
  - Czas dostępu do wszystkich regionów pamięci jest taki sam
  - Czas dostępu do pamięci jest taki sam dla wszystkich procesorów
  - „Klasyczna” architektura SMP

Architektura UMA i NUMA c.d.

- Architektura NUMA – Non-Unified Memory Access.
  - Wszystkie procesory mają dostęp do całej pamięci.
  - Czas dostępu do różnych regionów pamięci może być różny.
  - Czas dostępu do danego regionu pamięci będzie różny dla różnych procesorów.

Architektura UMA i NUMA c.d.

- NUMA jest rozwiązaniem pośrednim między klasycznym SMP a klastrem komputerów
  - Tradycyjny system SMP
    - limit na liczbę CPU z powodu magistrali.
    - dostęp do wspólnej pamięci.
  - Klaster komputerów
    - rozproszona pamięć, konieczna programowa kontrola spójności danych
    - wygodna skalowalność

Architektura UMA i NUMA c.d.

- NUMA
  - Znaczne lepsza skalowalność (typowy system SMP – do 64 CPU, SGI Origin NUMA – do 1024 CPU.
  - Zachowanie mechanizmu pamięci współdzielonej (łatwa komunikacja) i sprzętowej kontroli spójności danych (CC-NUMA – Cache coherent NUMA)

Systemy rozproszone

- Klaster komputerów
- Chmura obliczeniowa
- Środowisko Grid

Klaster komputerowy

- Grupa osobnych komputerów połączonych siecią
- Logicznie stanowią pojedynczy zasób obliczeniowy.
- Każdy komputer to **węzeł**, węzły wymieniają **komunikaty**.
- Model programowy zakłada, że struktura klastra nie jest widoczna dla użytkownika.

Klaster komputerowy c.d.

- Zalety stosowania klastrów komputerowych
  - Wydajność dopasowana do potrzeb (dopasowanie liczby węzłów).
  - Bardzo elastyczna skalowalność (dodawanie nowych węzłów).
  - Odporność na awarie (inne węzły przejmują obliczenia).
  - Znakomity stosunek ceny do wydajności.

Klaster komputerowy c.d.

- Wady klastrów
  - Trudniejsze w zarządzaniu
  - Zajmują więcej miejsca
  - Większe zużycie prądu
- Klastry sprawdzają się dobrze w problemach obliczeniowych. W przypadku problemów wymagających transferu danych szybkość sieci może limitować wydajność.

Klaster komputerowy – przykłady realizacji

- Klaster Beowulf
  - zbudowany z ogólnie dostępnych podzespołów
  - poszczególne węzły nie posiadają osobnych klawiatur i monitorów
  - węzły w lokalnej sieci, dostępne poprzez jeden z węzłów (server node)
  - pracuje pod kontrolą systemu Linuxo-podobnego
  - wykorzystuje biblioteki PVM, MPI

Klaster komputerowy – przykłady realizacji c.d.

- Klaster Beowulf
  - zbudowany z ogólnie dostępnych podzespołów
  - poszczególne węzły nie posiadają osobnych klawiatur i monitorów
  - węzły w lokalnej sieci, dostępne poprzez jeden z węzłów (server node)
  - pracuje pod kontrolą systemu Linuxo-podobnego
  - wykorzystuje biblioteki PVM, MPI

Klaster komputerowy – przykłady realizacji c.d.

- Sun Solaris
  - Przetwarzanie zorientowane obiektowo
  - Globalne zarządzanie procesami (z możliwą migracją procesów pomiędzy węzłami)
  - Rozproszony system plików
  - System operacyjny dostępny w ramach projektu OpenSolaris

Chmura obliczeniowa

- Rozproszone środowisko obliczeniowe
- Bazuje na modelu „Utility computing”
  - Dostawca usług udostępnia usługę w postaci rozproszonych zasobów obliczeniowych
  - Klient może skorzystać z usług udostępnionych w chmurze
- Chmura zapewnia elastyczne, skalowalne, odporne na błędy i bezpieczne środowisko do uruchamiania aplikacji.

Grid

- System integrujący dużą liczbę urządzeń znajdujących się w różnych lokalizacjach (często odległych)
- Obejmuje komputery, infrastrukturę sieciową, nośniki danych oraz różnorodne sensory.
- Widziany jako wirtualny superkomputer (przezroczysty dostęp do rozproszonych zasobów).
- Otwarte standardy – łączenie różnorodnych technologii.
- Architektura oparta o usługi.

GRID - przykłady

- European Grid Initiative (EGI) – europejskie środowisko gridowe
- Status na 2010 r.
  - 10000 użytkowników
  - prawie 250 000 procesorów (rdzeni)
  - 40 petabajtów przestrzeni dyskowej
  - 317 węzłów w 52 krajach

GRID – przykłady c.d.

- Polskie środowisko gridowe PL-GRID
- Uczestnicy
  - Cyfronet AGH (Kraków)
  - ICM (Warszawa)
  - Poznańskie Centrum Superkomputerowe
  - Akademickie Centrum Komputerowe, Gdańsk
  - Wrocławskie Centrum Superkomputerowe

PL-Grid c.d.

- Planowane zasoby (koniec 2011)
  - moc obliczeniowa 215 TFLOPS
  - przestrzeń dyskowa 2500 TB
- Obszary zastosowań
  - biologia
  - chemia kwantowa
  - fizyka
  - symulacje numeryczne ...

Podsumowanie

- Komputery równoległe – taksonomia
- Systemy SMP
  - charakterystyka
  - sposoby realizacji
  - systemy SMP i pamięć podręczna
- Architektura NUMA.
- Środowiska rozproszone – klastry, chumury obliczeniowe, środowiska gridowe.