Systemy operacyjne III

WYKŁAD 7

Jan Kazimirski

Komputery równoległe

Wydajność komputerów

- Rozwój technologii wiąże się z ciągłym wzrostem wydajności komputerów
- Pierwsze komputery – 1-100 operacji/sek.
- Najszybszy obecnie superkomputer – **Tianhe-1** (Chiny) - 2,507 PFLOPS (peta = 10^{15})
  - 14336 CPU XEON + 7168 NVIDIA Tesla GPU
  - 262 terabajty pamięci operacyjnej
  - 2 petabajty pamięci dyskowej

Granice wydajności komputerów

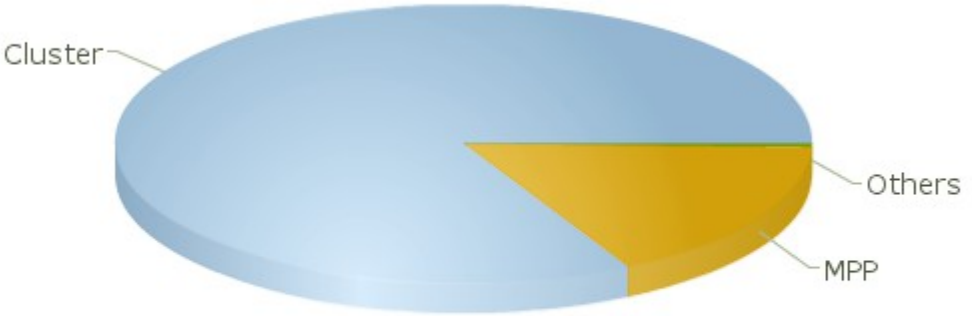
- Szybkość zegara
  - Ograniczenia wynikające z teorii względności
  - Taktowanie 10 Ghz – długość ścieżki < 2cm
  - W przypadku zegara 1 Thz – komputer musi mieć wielkość nie więcej niż 100 mikrometrów.
- Rozpraszanie ciepła
  - Im mniejszy komputer tym trudniej odprowadzić ciepło.

Współczesne superkomputery

- Lista rekordzistów – www.top500.org
- Na stronie znajduje się ranking najszybszych komputerów świata.
- Lista aktualizowana jest 2 razy do roku.
- Ostatnie rekordy:
  - 11/2010 Tianhe-1A – 2566000 GFLOPS
  - 06/2010 Jaguar – 1759000 GFLOPS
  - 06/2009 Roadrunner – 1105000 GFLOPS

Top500

Systemy wieloprocesorowe i rozproszone dominują na liście Top500

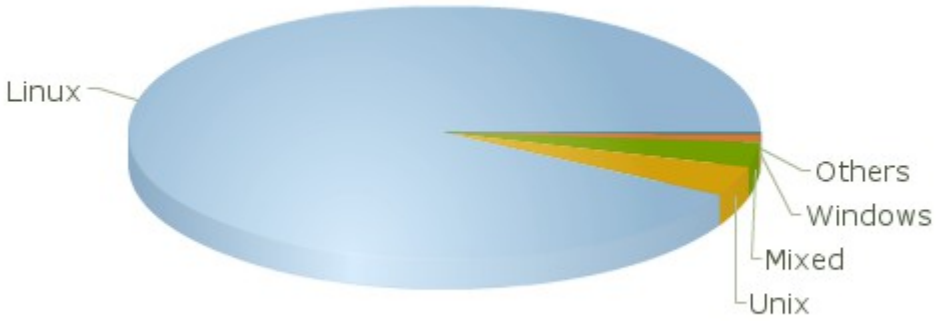


Źródło: www.top500.org

Top500 c.d.

Czy ktoś jeszcze uważa, że nie warto uczyć się Linuksa???

DYGRESJA



Wektoryzacja na poziomie procesora

- Klasyczny komputer – architektura szeregowa
- Wzrost taktowania nie jest wystarczający do efektywnego wzrostu wydajności
- Techniki wektoryzacji na poziomie procesora – potoki, jednostki superskalarne, out-of-order execution

Potoki

- „Linia montażowa” procesora
- Rozkaz podzielony na fazy (pobranie, dekodowanie, pobranie operandów, wykonanie itp.)
- Poszczególne podzespoły jednostki wykonawczej wykonują poszczególne fazy rozkazu.
- Kolejny rozkaz może być pobrany zanim skończy się wykonywanie rozkazów poprzednich
- Pentium4 – potoki 20 fazowe.

Superskalarność

- Zwielokrotnienie liczby jednostek wykonawczych.
- Przykład: PowerPC 970
  - 4 jednostki ALU
  - 2 jednostki FPU
  - 2 jednostki SIMD
- Procesor może wykonywać kilka rozkazów jednocześnie.
- Zwykle łączona jest z potokami.

Inne techniki

- Out-of-order execution – zmiana kolejności wykonywania rozkazów w celu eliminacji zależności i lepszego wykorzystania jednostek wykonawczych
- Wykonywanie spekulatywne – wykonywanie instrukcji za skokiem warunkowym (z wyprzedzeniem).

Taksonomia Flynna

- Klasyfikacja architektur komputerów ze względu na sposób przetwarzania:
  - **SISD** – Single Instruction, Single Data – komputer skalarny
  - **SIMD** – Single Instruction, Multiple Data – komputer wektorowy
  - **MISD** – Multiple Instruction, Single Data
  - **MIMD** – Multiple Instruction, Multiple Data – system wieloprocesorowy, klaster

Model pamięci

- Pamięć dzielona – pamięć wspólna dla wszystkich jednostek wykonawczych
- Pamięć rozproszona – poszczególne jednostki mają osobne obszary pamięci
- Architektury UMA/NUMA
  - UMA – Uniform Memory Access
  - NUMA – Non-Uniform Memory Access

Komputery równoległe

- Procesory wielordzeniowe
- Symetryczna multiprocesorowość (SMP)
- System wieloprocesorowy z rozproszoną pamięcią
- Klastry komputerowe
- MPP – masowa równoległość
- Środowiska gridowe

Oprogramowanie

- Rozproszone systemy operacyjne
  - Ukrywanie szczegółów architektury przed użytkownikiem końcowym
- Aplikacje równoległe
  - **Jawne** – pełna kontrola programisty.
  - **Częściowo jawne** – pod kontrolą programisty i kompilatora
  - **Niejawne** – pod całkowitą kontrolą kompilatora

Model MPI – Message Passing Interface

- Architektura z pamięcią rozproszoną – klaster komputerów.
- Program – zbiór niezależnych procesów
- Komunikacja między procesami za pomocą wymiany komunikatów
- Model MPI określa wygląd API – biblioteki do obsługi przesyłania komunikatów

Elementy MPI

- Funkcje służące do rozpoczęcia, obsługi i zakończenia komunikacji.
- Funkcje przesyłające komunikaty między parami procesów
- Funkcje przesyłające komunikaty między grupami procesów
- Funkcje do tworzenia specyficznych typów danych

MPI – przykład 1/2

```
#include<iostream>
#include<stdio.h>
#include<mpi.h>

using namespace std;

int main(int argc,char* argv[]) {

  int np,id;
  int slave;
  MPI_Status stat;

  MPI_Init(&argc,&argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &id);
  MPI_Comm_size(MPI_COMM_WORLD, &np);
```

MPI – przykład 2/2

```
if(id==0) {
        // master
        for(int i=1;i<np;i++) {
            MPI_Recv(&slave,1,MPI_INT,MPI_ANY_SOURCE ,
                    1,MPI_COMM_WORLD,&stat);
            cout << slave << '\t'
                << stat.MPI_SOURCE << '\t'
                << stat.MPI_TAG << endl;
        };
} else {
        // slave
        MPI_Send(&id,1,MPI_INT,0,1,MPI_COMM_WORLD);
};

MPI_Finalize();
return 0;
};
```

Beowulf

- Klaster komputerowy dający dużą wydajność niewielkim kosztem
- Budowany ze standardowych komputerów PC połączonych siecią Ethernet
- Zazwyczaj pod kontrolą systemu Linuks
- Wykorzystuje biblioteki MPI lub PVM

Mosix

- Oparty na jądrze Linuksa
- Realizuje model SSI (Single System Image) użytkownik widzi klaster jako jedną maszynę
- Zawiera mechanizmy równoważenia obciążenia – potrafi przenosić zadania pomiędzy węzłami (migracja zadań)

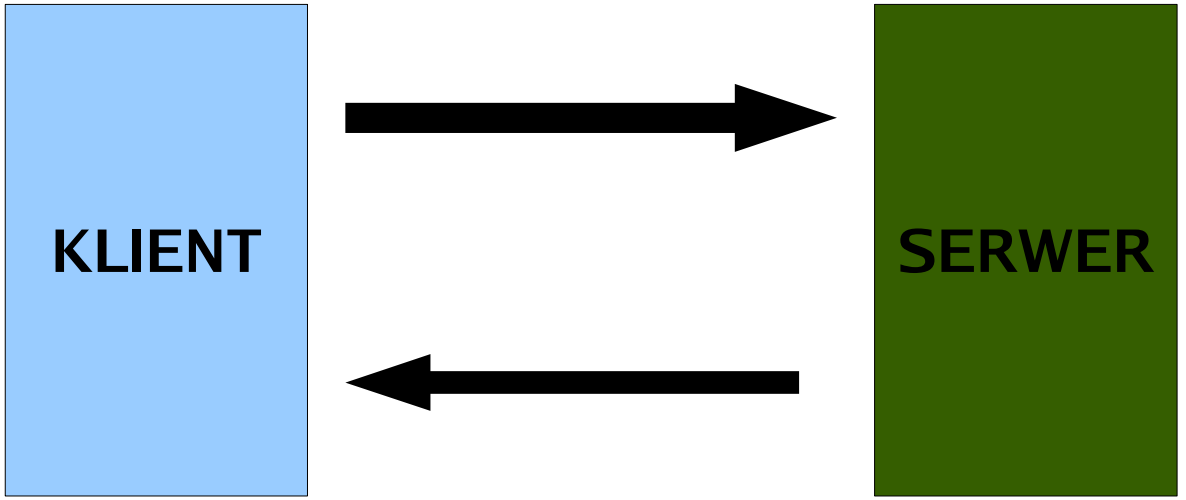
Rozproszony system plików

- Umożliwia połączenie przestrzeni dyskowych wielu komputerów (klaster)
- Dane są dostępne dla dowolnego węzła w klastrze w sposób przezroczysty
- Przykład: system plików **Lustre** (stosowany m.in. w klastrach Tianhe-1A i Jaguar

Technologia klient/serwer

- Klient – komputer osobisty lub stacja robocza. Środowisko graficzne, wygodne w użyciu.
- Serwer – wydajny system z zestawem usług dla klientów. Usługi mogą być współdzielone – np. serwer bazodanowy obsługujący pracowników firmy

Architektura 2-warstwowa



Klasy aplikacji klient serwer

- Przetwarzanie na głównym komputerze
- Przetwarzanie po stronie serwera
- Przetwarzanie po stronie klienta
- Przetwarzanie zespołowe

Przetwarzanie na głównym komputerze

- Całość lub większa część przetwarzania odbywa się na głównym komputerze
- Klient jest tylko terminalem
- Przykład – praca zdalna na komputerze mainframe.

Przetwarzanie po stronie serwera

- Serwer odpowiada za przetwarzanie danych
- Klient odpowiada za prezentację uzyskanych danych.
- Obciążenie przypada w większości na serwer. Klient w zasadzie udostępnia tylko interfejs graficzny
- Architektura tzw. „*cienkiego klienta*”

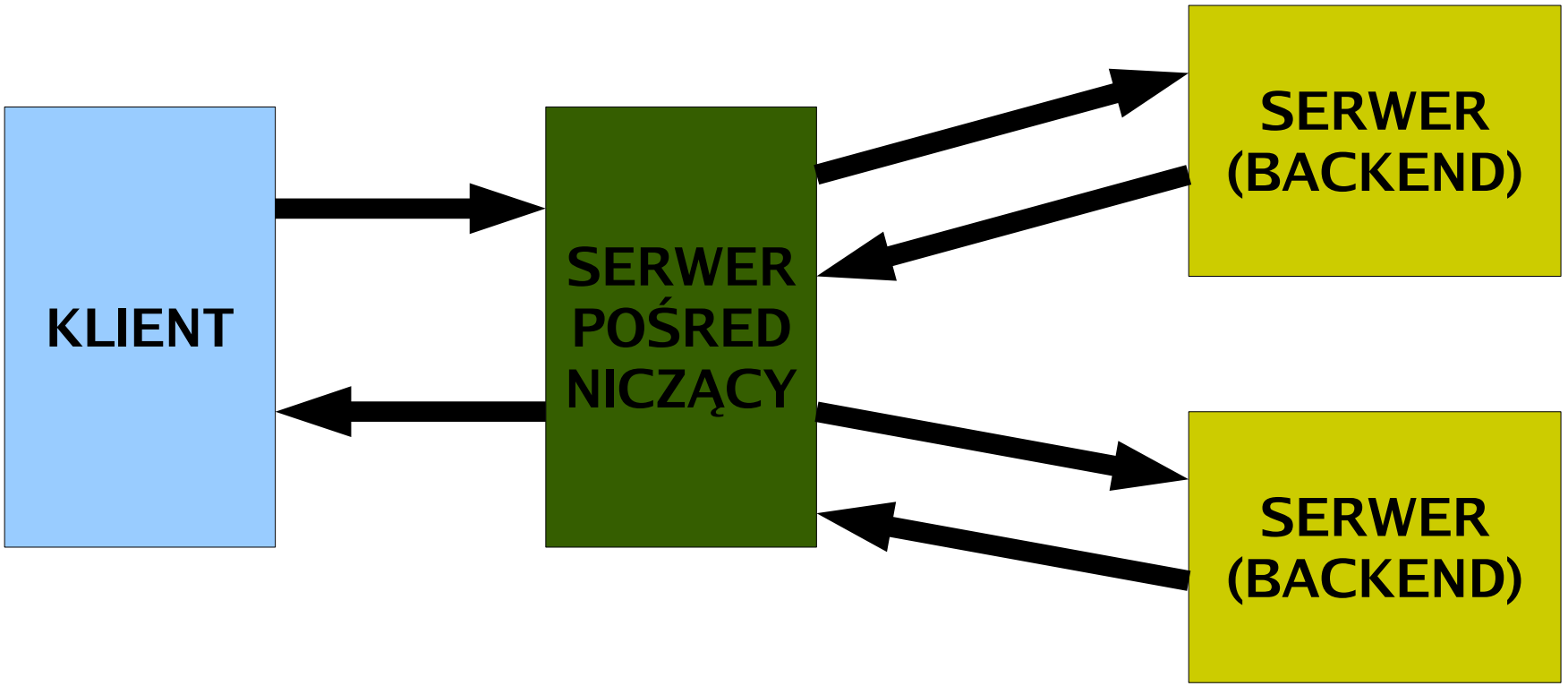
Przetwarzanie po stronie klienta

- Przetwarzanie danych po stronie klienta
- Serwer zawiera tylko podstawowe procedury obsługi i weryfikacji danych
- Obciążenie przypada w większości na klienta
- Architektura tzw. „*grubego klienta*”

Przetwarzanie zespołowe

- Przetwarzanie danych rozkładane jest pomiędzy serwer i klienta w celu optymalizacji obciążenia
- Rozwiązanie najtrudniejsze do realizacji ale zapewniające największą elastyczność i wydajność.

Architektura 3-warstwowa



Zdalne wywoływanie procedur (RPC)

- Komunikacja programów na różnych komputerach za pomocą wywoływania procedur w sposób zdalny
- Prosta składnia.
- Przejrzystość. Wywołanie niewiele różni się od wywołania lokalnego
- Przejrzyste, przenośne interfejsy.

Powiązanie klienta i serwera

- Powiązanie nietrwałe – połączenie zestawiane tylko na czas wykonania procedury.
- Powiązanie trwałe – zestawiane raz i podtrzymywane. Może być wykorzystane ponownie do wywoływania tej samej lub innych zdalnych procedur.

Synchroniczne i asynchroniczne RPC

- Synchroniczne RPC – po wywołaniu procedury klient czeka na rezultat.
- Asynchroniczne RPC – nie blokuje klienta. Klient i serwer mogą niezależnie realizować zadania.

Mechanizmy zorientowane obiektowo

- Obiekty na zdalnych komputerach komunikują się poprzez wymianę komunikatów.
- Klient wysyła żądanie do obiektu brokera (katalog dostępnych usług)
- Broker wywołuje odpowiedni obiekt i przekazuje mu dane.
- Odpowiedź zdalnego obiektu zwracana jest klientowi.

Mechanizmy zorientowane obiektowo c.d.

- **COM** – Component Object Model (Microsoft)
- **CORBA** – Common Object Request Broker Architecture (IBM, Apple, Sun)
- **DCOP** – Desktop Communication Protocol (stosowany w KDE)
- **SOAP** – Simple Object Access Protocol (standard W3C – korzysta z XML i zwykle protokołu HTTP).

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.

SETI@home

- Jeden z pierwszych „nieoficjalnych” gridów
- Uruchomiony w 1999 roku w celu poszukiwania śladów obcych cywilizacji w kosmosie
- Ogólnie dostępny program klienta (Windows, Linux, Mac OS).
- Porcje danych wysyłane z serwera i przetwarzane przez klienta (screensaver)

SETI@home c.d.

- Liczba uczestników projektu (w całym okresie trwania) to ponad 5 mln.
- Projekt jest posiadaczem rekordu Guinnessa za największe obliczenia w historii.
- W 2009 roku moc obliczeniowa projektu wynosiła ponad 769 TFLOPS (RoadRunner – 1105 TFLOPS)

World Community Grid

- Oryginalny projekt SETI@home obejmował tylko jedno zagadnienie
- WCG jest publicznym środowiskiem gridowym służącym do różnorodnych obliczeń.
- Statystyki:
  - ponad 500 tys. użytkowników
  - ponad 1.5 mln komputerów
  - całkowity czas pracy – ponad 400 tys. lat

World Community Grid c.d.

- Aktywne projekty:
  - Computing for Clean Water
  - The Clean Energy Project
  - Help Cure Muscular Dystrophy
  - Help Fight Childhood Cancer
  - Help Conquer Cancer
  - Human Proteome Folding
  - FightAIDS@Home

EGI

- 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

PL-Grid

- Polskie środowisko gridowe
- 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 ...