Kap. 1: Modeller for parallelle beregninger

- **Introduksjon, bakgrunn**
  - parallell-prosessering
  - datamaskinarkitektur ...*helhets*-forståelse
  - nye utfordringer
  - ytelse: mål og faktorer
  - programmeringsomgivelser
- **Multiprosessorer og multimaskiner**
  - delt lager: UMA, NUMA og COMA
  - distribuert lager
- **Multivektor maskiner**
- **SIMD-maskiner**
- **Teoretiske modeller**
  - litt grunnleggende kompleksitetsteori
  - Parallel Random Access Machine: PRAM
  - VLSI modeller
- **Viktige maskiner, "utviklings-linjer"**

Parallell prosessering, bakgrunn

- **Gammel teknikk**
  - parallellitet har alltid vært brukt for å bedre ytelse
- **Oppblomstring på 90-tallet**
  - kommersielle stormaskiner basert på massiv parallelitet (i tillegg til CRAY type maskiner)
  - forbedret teknologi, både komponenter og verktøy
  - "høst-stemning" høsten-94
- **Ulike former, ulike nivåer**
  - generell, varig lærdom
- **Massivt parallell prosessering (MPP)**
  - 1000 prosessorer el. mer
- **Drømmen om TeraFLOPs maskinen**
  10**12 flyt-talls operasjoner pr. sekund
- **Hwang (1993):** "The rapid progress made in hardware technology has significantly increased the economical feasability of building a new generation of computers adopting parallel processing. *However, the major barrier preventing parallel processing from entering the production mainstream is on the software and application side."*

Klassifisering av parallelle datamaskiner

- **Taxonomy: *the science or technique of classification (Webster)***
- **Flynn's taksonomi (skjema) (1966)**
  - en/flere instruksjons/data strømmer
  - den mest brukte
  - generelt ansett som for grov
    - » SISD
    - » SIMD
    - » MISD
    - » MIMD
- **Gordon Bell (1992)**
  - videre klassifisering av MIMD
- **Mange andre finnes**
  - Haendler, Feng, ...
  - Ralph Duncan (1990), høyt nivå, praktisk orientert
  - A. Basu (1987), detaljert, lite kjent
  - ....

Sentrale maskin-karakteristika

- **Topp-hastighet (Eng. peak-performance)**
  - "hastighet som helt sikkert aldri oppnås"
- **Gjennomsnittsverdier alene som ytelses-mål er "uinteressant"**
  - målte verdier må knyttes til et program eller et utvalg av (velkjente) program. "Program-mix" el. "benchmark "
- **Ytelses-indikatorer**
  - klokkerate = (1/syklustid) = ***f***
  - et programs kjøretid kan måles i antall instruksjoner (I_c)
  - kjøretid <> programstørrelse
  - sykler pr. instruksjon (**CPI**), gitt av:
    - » instruksjons-sett
    - » implementasjon (prosessor)
    - » program som kjøres
  - kjøretid (CPU) = (I_c x **CPI**) / ***f***
  - MIPS-rate (se likn. (1.3) side 15)
  - system-ytelse ("gjennomstrømming") (Eng. throughput)
    - » antall programmer utført pr. sekund
    - » normalt lavere enn CPU-ytelse, p.g.a. diverse overhead

Multiprosessorer med delt lager

- **Logisk delt lager**
  - delt lager sett fra programmereren, dvs. ett felles adresserom
  - kan være fysisk delt el. distribuert
- **Tett koplet system**
  - høy grad av ressursdeling og samarbeid
- **Tre hovedtyper, UMA, NUMA, COMA**
  - **UMA** = "uniform memory access"
    - » alle prosessorer har lik aksess-tid til alle ord i lageret
    - » Typisk: prosessorer / PMIN / lagermoduler
    - » Mange kommersielle modeller med lavt antall prosessorer (Bl.a. IBM xx, DEC VAX, Alliant)
  - **NUMA** = "nonuniform memory access"
    - » aksesstid varierer med fysisk plassering av et ord i lageret
    - » Typisk: prosessorer med lokale lager
    - » Mange forskningsprototyper på 80-tallet (bl.a. "HC-16"), noen kommersielle
  - **COMA** = "cache-only memory architecture"
    - » variant av NUMA der alle data er lagret i hurtigbuffere

COMA = "cache-only memory architecture"

- **Variant av NUMA der alle data er lagret i hurtigbuffere**
  - "ikke noe lagerhierarki ved hver prosessor "(?)
    - » men kan ha fler-nivå's hurtigbuffer
  - data vil flytte seg dit de blir brukt!
  - distribuert hurtigbuffer-katalog (fig 1.8)
  - CC-COMA
    - » variant der alle kopier av en lager-blokk er garantert å være konsistente
- **Forskningsprototyper**
  - KSR1 (Kendall Square Research), kap. 9
  - Data Diffusion Machine (DDM), laget ved SICS (Swedish Institute of Computer Science)

Mer om multiprosessorer

- **Hybrider og varianter**
  - CC-NUMA, distribuert fysisk lager og hurtigbuffer-kataloger som benyttes for å opprettholde hurtig-buffer-koherens (Eng. cache-coherence)
    - » DASH (Stanford), kap. 9
    - » Alewife (MIT)
- **Symmetrisk vs. asymmetrisk MP**
  - gitt av fordeling av IU-enheter og OS-ansvar på prosessorene
- **Eksempler, se tabell 1.3**
  - høy-ytelses stormaskin (Eng. mainframe), mini-superdatamaskin, nesten-superdatamaskin☺
- **Typisk**
  - generelle systemer
  - mange brukere
  - vekt på (rimelig lett) programmering
- **Skillet mellom multiprosessorer** *(flere prosessorer m/delt lager/adresse-rom)* **og multimaskiner** *(flere [autonome] maskiner som samarbeider)* **er i ferd med å viskes ut**
  - trend mot et felles (delt) virtuelt adresse-rom realisert vha. fysisk distribuert lager

Multimaskiner med distribuert lager

- **Fysisk (og logisk) distribuert lager**
  - Programmereren ser at lageret er distribuert
  - Kommunikasjon mellom prosessorer må skje ved meldingsutveksling
  - Mer skalerbare, men mindre programmerbare
- **Noder (prosessorer)**
  - prosessor med lokalt lager
  - mer el. mindre selvstendig (autonom) datamaskin
  - ofte i dag med egen dedikert "rutings-prosessor"
- **Løst koplet system**
  - kommunikasjon mellom prosessorer tar vesentlig lenger tid enn internt innen en node
  - ønsker å minimalisere nødvendig kommunikasjon
- **Tradisjonelt; NORMA-maskiner**
  - NORMA = No-remote-memory access
    - » lokalt node-lager kan bare aksesseres av node-prosessor
  - denne begrensningen blir nå mindre aktuell (skjult) gjennom implementasjon av distribuert delt lager
- **Heterogen prosessering**
  - ulike noder
  - standard-protokoller for meldingutveksling
  - se f.eks. IEEE Computer Juni 1993

(Multi)vektor-maskiner

- **Mange navn:**
  - Vektor-maskin, vektor-prosessor, trinnvis prosessering, trinn-prosessor, segment-prosessor, samlebåndsmaskin, "pipeline"-maskin, CRAY-type maskin, tradisjonell superdatamaskin
- **Særtrekk**
  - lange, spesialiserte, meget hurtige "pipeline" for å utføre "tyngre" (gjerne aritmetiske) operasjoner
  - kjeding av operasjoner for å øke samlebåndslengde
  - vektoriserende kompilatorer og FORTRAN
- **Register-til-register arkitektur**
- **Lager-til-lager arkitektur**
- **CRAY 1, 2, X-MP, Y-MP,...**
- **Japanske "CRAY-etterlikninger"**
  - Fujitsu, Hitachi, NEC, ...

SIMD (superdata)-maskiner

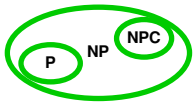
- **Typisk arkitektur, se fig. 1.12**
- **Særtrekk**
  - en kontroll-enhet el. "sequencer", (CU)
    - » kontroll, skalare operasjoner
    - » effektiv kringkasting
    - » evt. aksess til program-lager
  - 64 ... 64*k* prosessorer el. beregningselementer (PE)
    - » instruksjoner for (data)-parallelle beregninger
    - » aritmetiske, logiske
    - » SIMD krever maskering, på/av-skruing
    - » kommunikasjon mellom prosessorer (IPC), rutingsfunksjoner
- **"Kombinerbare" en-bits prosessorer**
- **Eksempler**
  - Illiac IV, MPP, MasPar MP-1 (UiB)
- **Spesielle anvendelser**
- **Mister markedsandeler, men...**
  - MasPar hevder at deres ASIMD (Autonomous SIMD) maskiner er bedre enn MIMD

Teoretiske modeller

- **Abstrakte maskiner**
  - enkle, matematiske, generelle
  - gjør det lettere å uttrykke beregninger
  - fjerner seg fra (maskin)-detaljer og "virkeligheten"
  - oppnår økt portabilitet og varighet av resultater
- **Den ideelle modell**
  - En enkel, lett-brukt, varig og dominerende modell som er "representativ" for den ytelse som kan oppnås i praksis på eksisterende maskiner.
    - » RAM-modellen for en-prosessor maskiner
    - » ????? for parallelle beregninger
- **Fundamental kompleksitetsteori**
  - problem-størrelse og kompleksitetsfunksjoner
  - tidskompleksitet
  - plasskompleksitet
  - deterministiske og ikke-determinstiske algoritmer
  - polynomisk og eksponensiell kompleksitet
  - NP-komplette problemer
  - polylogaritmisk tid og Nick's klasse (NC)
  - P-komplette problemer (iboende serielle)
  - mange, mange andre kompleksitetsklasser

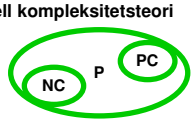
Viktige kompleksitetsklasser

Tradisjonell (seriell) kompleksitetsteori



NP = klassen av problemer som kan løses ikke-deterministisk i polynomisk tid (bare deterministisk i eksponensiell tid)
P = klassen av problemer som kan løses deterministisk i polynomisk tid
NPC = "de vanskeligste problemene i NP m.h.p. løsning i polynomisk tid"

Parallell kompleksitetsteori



P = som over
NC = Nick's Class, problemer som kan løses i poly-logaritmisk tid ("ultraraskt") vha. et polynomisk ("realistisk") # prosessorer
PC = "de vanskeligste problemene i P m.h.p. "ultrarask" løsning med "realistisk" # prosessorer

PRAM-modellen

- **Fortune & Wyllie 1978**
  - dominerende modell for forskning og beskrivelse av parallelle algoritmer
  - generelt kritisert som for urealistisk
- **Arkitektur (fig 1.14)**
  - (Uendelig mange) synkrone prosessorer med hver sin program-teller (dvs. synkron MIMD)
  - Et (uendelig) stort globalt lager (UMA)
  - (små lokale lager)
- **Varianter;**
  - eksklusiv/samtidig lesing/skriving gir 4 varianter
  - EREW PRAM
    - » den svakeste
  - CREW PRAM
    - » den mest brukte (den sterkeste veldefinerte)
  - ERCW PRAM
  - CRCW PRAM
    - » den sterkeste, finnes i en rekke undervarianter
    - » Common
    - » Arbitrary
    - » Minimum
    - » Sum
    - » Priority
    - » Garbage