Den vanskelige maskinen – eller hvordan lage parallelle programmer på en moderne maskin

Arne Maus,

Gruppe for Objektorientering, Modellering og Språk (OMS)

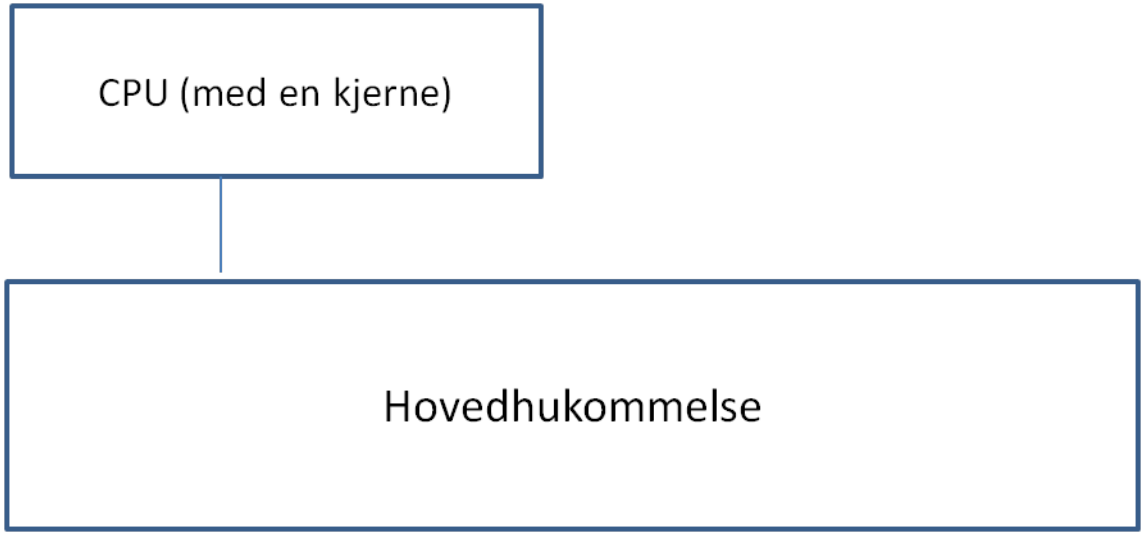
Oversikt – hvorfor og hvordan

- Hvorfor ha parallelle programmer
- Den vanskelige maskinen (DVM)
  - Før 1980
  - Langsom hukommelse ⇨ **cache** 1980
  - Varmegang ⇨ **flerkjerne** 2005
- Hvordan programmere DVM?
  - Sekvensielt
  - Felles data ⇨ **synchonized** method
  - Start, stopp – lese hverandres resultater ⇨ **barrier** - synkronisering.

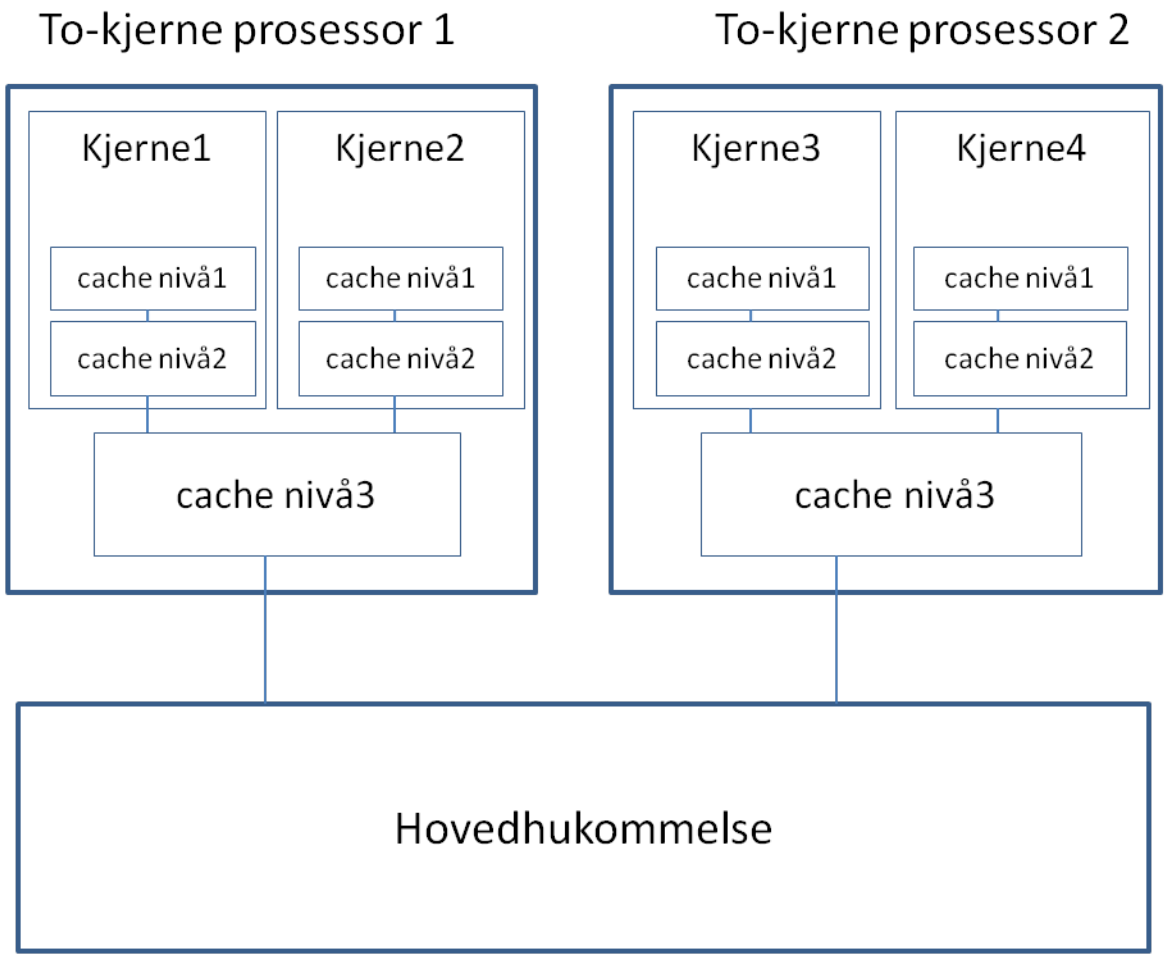
Hvorfor lage et parallell program

1. Noe går for langsomt , egen tråd
   - Tegne på skjerm, søk i database
2. Logikken i problemet blir lettere, en tråd snakker med, og er skrevet for, én bruker
   - Eks: Kundesystem
   - Innlevering av Obliger
3. Få programmet til å gå fortere, med k kjerner
   - Ønsker å gå k ganger fortere

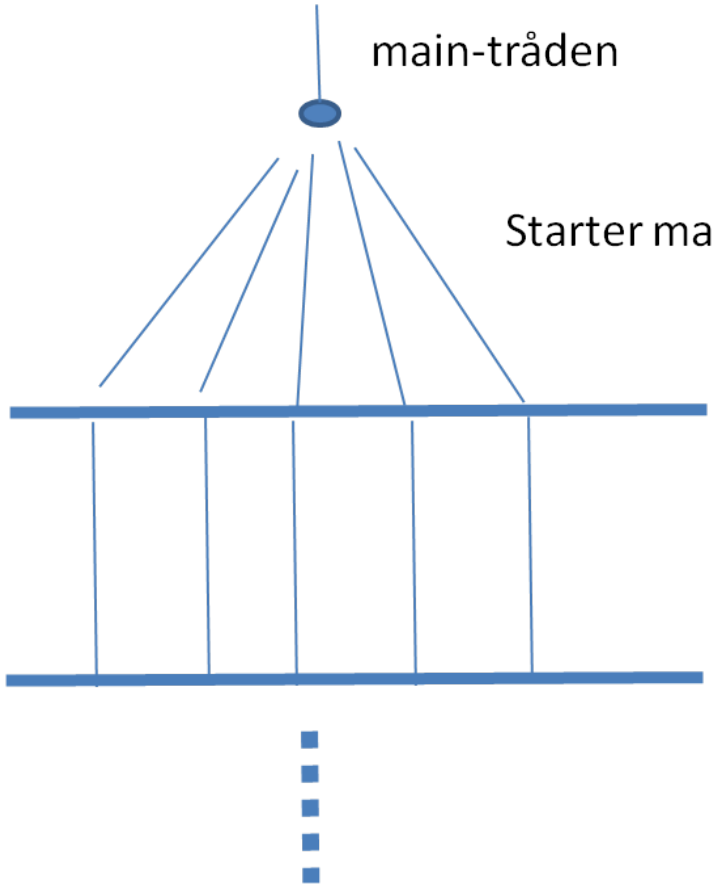
Maskin før 1980:



En 'liten', moderne maskin med 4 kjerner



Starting av tråder og barriere-synkronisering



Barriere, trådene venter på at *alle* trådene ankommer før de kan fortsette

Ny barriere, trådene venter igjen på hverandre før de kan fortsette

Noen regler

- For at det skal være vits å parallellisere et problem, må det ha *mer* enn noen få operasjoner for hvert dataelement. Som en huskeregel kan vi si at hvis den største versjonen av problemet vi greier å kjøre sekvensielt ikke tar mer enn 1 sekund, er det ingen vits i å parallellisere det.
- Start med en godt testet sekvensiell løsning av problemet.
- Se etter om man kan dele ***data*** opp i et antall like store deler, hvor helst hver del kan løses etter den sekvensielle metoden i hver sin tråd – altså i parallell.
- Hvis vi under beregningene må ha felles data, må de alltid beskyttes . All skriving og lesing på disse felles data skjer med synkroniserte metoder. Hvis ***alle*** data alltid er felles er det ingen vits i å parallellisere problemet.
- Når hver tråd har løst sin del, og etter at alle har ventet på *en felles barriere* når de er ferdige, så kan alle trådene etterpå lese resultatene av hverandres beregninger.
- Kanskje er vi nå enten ferdig, eller resultatene fra første beregninger kan kanskje igjen deles opp i flere tråder med en ny barriere., osv.
- Tenk spesielt på hvordan main-tråden skal vente og slippe løs når alle trådene er ferdige og har løst problemet. Det kan godt være en barriere som venter på antall tråder +1, som er main-tråden, som main legger seg og venter på når alle trådene er startet.

Oppsummering