

## Arvutusjõudlus ja ribalaius

Peale salvestusruumi on süsteemides veel muid ressursse, neist kaks olulisemat on (arvutus)jõudlus ja ribalaius. Arvutusjõudlus on neist kahest veidi selgem ja kindlapiirilisem ressurss, ribalaius natukene raskemini hallatav. Samas on nad omavahel tihedalt seotud: suurest jõudlusest pole kasu, kui pole piisavalt ribalaiust tulemuste edasitoimetamiseks, samuti jääb suure ribalaiusega suhtluskanal kasutamata, kui pole piisavalt arvutusjõudlust selle täitmiseks ja haldamiseks.

### Arvutusjõudlus

Arvutusjõudlus peegeldab süsteemi andmetöölemisvõimekust. Põhiline arvutusjõudluse määraja on kasutatav protsessor. Jõudlus sõltub nii protsessori arhitektuurist kui ka taksagedusest. Ressursina erineb arvutusjõudlus salvestusruumist ennekõike selle poolest, et tegu pole salvestatava ressursiga. Sarnaselt salvestusruumiga on arvutusjõudlus lõplik.

Arvutusjõudlust kui ressursi ei ole võimalik (näiteks hiljem kasutamise eesmärgil) salvestada – protsessor on võimeline mingi hulga taktide jooksul tegema mingi hulga operatsioone. Kui me neid operatsioone hetkel ei teosta, siis läheb see hulk ressursi lihtsalt kaotsi. Ehk siis- protsessor toodab kindla aja jooksul kindla hulga lisaväärtust, ja kui sellele ei leidu kohe samal hetkel kasutajaid, siis see lisaväärtus läheb raisku.

Arvutusjõudlus on lõplik – protsessor teeb mingi taktide arvu kohta mingi hulga operatsioone ja ei ole võimeline seda piiri ületama. Mingi osa protsessorite kohta kehtib ka see, et nad ei ole võimelised tegema mingi taktide arvu kohta vähem operatsioone.

### Ribalaius

Ribalaius peegeldab mingi andmeedastuskanali võimet kindlas ajaühikus andmeid edastada. Mida suurema hulga andmeid suudab kanal kindla aja jooksul edastada, seda suurem on ribalaius. Andmeedastuskanali võimekuse määrab tegelikkuses kanalis korraga edastava paketi maht (paralleelsus, laius) ja edastussagedus, aga samuti ka suhtlusreeglistik ja kontrollinfo osa andmevoos. Suhtluskanali võib jaotada vahenditeks, mis teevad suhtluse füüsiliselt võimalikuks – elektrit juhtivad kanalid vms – ja vahenditeks, mis korraldavad ja mõtestavad selle suhtluse – protokollid, standardid.

Andmeedastuskanaleid saab tinglikult jagada kaheks: jagatud kanalid (nt. siinid) ja sisemised (sihtotstarbelised) kanalid.

Jagatud kanaleid iseloomustab see, et tegemist on mitmes mõttes standardiseeritud (elektriliselt, füüsiliselt, protokolliliselt) suhtluskanaliga, mille kaudu suhtlevad erinevad süsteemi komponendid. Komponendid ei pruugi olla ühe tootja omad, samuti ei pruugi nad olla ühte tüüpi.

Jagatud kanalid on näiteks IDE kanalid ja SCSI siin, Ethernet võrk, Laiendussiinid (PCI, ISA).

Sisemised kanalid ühendavad reeglina vähem punkte, on vähem standardiseeritud ja kasutavad lithsaid, tootjaspetsiifilisi protokolle. Samas on sisemised kanalid tihtipeale kiiremad ja kallimad realiseerida.

Sisemine kanal on näiteks protsessori esimese taseme vahemälu (*level 1 cache*) ja protsessori vaheline suhtluskanal.

## **Tarbijad**

### **Arvutusjõudluse tarbijad**

Arvutusjõudlust tarbib kogu programmikood, mis süsteemis jookseb. Seda koodi saab jaotada näiteks operatsioonisüsteemi koodiks ja rakendusprogrammideks.

Rakendusprogrammid (sealhulgas süsteemi teenused) on osa süsteemist, mis puutub otseselt kokku kasutajatega ja sõltub nende soovidest. Ka arvutusjõudlust tarbib rakendusosa tavaliselt vastavalt sellele, kui intensiivselt teda kasutatakse (muidugi on oma osa ka tarkvara kvaliteedil). Sellest tulenevalt on rakendusprogrammide vajadus arvutusjõudluse järgi hüppeline - pigem vajatakse suurt jõudlust mõneks hetkeks.

Operatsioonisüsteemi all mõistame eelkõige operatsioonisüsteemi tuuma, aga ka tuuma mooduleid (seadmedraiverid jms) ja operatsioonisüsteemi utiliite. Suurema osa ajast kasutaja otse operatsioonisüsteemi vahendeid ei kasuta, kasutamine toimub rakendustarkvara vahendusel. Operatsioonisüsteem vajab pigem väiksemat ja keskmist osa kogujõudlusest, kuid seda pidevalt.

Rakendustarkvara ja operatsioonisüsteemi kood mõjutavad vastastikku üksteise jõudlustarvet ja -saadavust - rakendusprogrammide lisandumisel ja vajaduste kasvamisel peab operatsioonisüsteem rohkem tööd tegema. Operatsioonisüsteemi suure koormuse korral läheb palju jõudlust "korralduslike" küsimuste peale ja rakendusprogrammidele jääb vähem arvutusaega.

Samuti tarbib (otse) mõnel määral arvutusjõudlust süsteemis olev riistvara. Riistvaradraiverite poolt tarbitud arvutusjõudlus liigitub pigem operatsioonisüsteemi poolt tarbitud koormuse alla.

### **Ribalaiuse tarbijad**

Kuna ribalaiuse mõiste on arvutusjõudluse mõistest üldisem, mitmekesisem, siis on võimalike tarbijate hulk ka kirjum.

Kuigi ka ribalaiuse tarbimise taga seisab tegelikult mingisugune programmikood, siis siin on otseseks tarbijaks tihemini siiski riistvara (ja tema sisene programmikood) ise.

### **Probleemid**

Lõplike ja jagatavate ressurssidega seotud probleemid jaotuvad põhiliselt kaheks: ressursi on ebapiisavalt või ressursi kasutatakse ebapraktiliselt, ebaühtlaselt.

Arvutusjõudluse puhul võime vaadelda veel ebapiisavuse kahte eri poolust: rakendusprogrammide tekitatud ressursipuudus/ülekoormus ja operatsioonisüsteemi tekitatud ülekoormus.

Ebaühtlus väljendub selles, et suure osa ajast toodetakse arvutusressursi asjatult, samas kui mingi osa ajast on nõudlus suurem, kui toota jõutakse. See on paratamatu igas süsteemis, aga negatiivse efekti vähendamiseks annab siiski ühte-teist teha.

Ressursipuudust võib lahendada mitmel moel:

- koormuse vähendamine
  - sh operatsioonisüsteemi tekitatud koormuse vähendamine
  - sh rakendusprogrammide tekitatud koormuse vähendamine

- sh rakenduste kõrvalejätmine, peatamine...
- arvutusjõudluse suurendamine
  - CPU uuendamine
  - CPU lisamine (SMP) – millal on SMP kasulik?

Ribalaiuse põhiprobleemiks on ressursi ebapiisavus, läbilaskevõime lõplikkus, aga sarnaselt arvutusjõudlusega kohtame ka siin ebapraktilist kasutamist.

Kui tegemist on jagatud suhtluskanaliga, siis võib juhtuda, et üks seade kasutab ebaproportsionaalses mahus ribalaiust, millega häirib teiste, vähenõudlikemate seadmete tööd.

Samuti võib probleemiks tulla see, et tänu suhtluskanali lõplikule läbilaskevõimele ei suudeta teatud piirist alates enam seadmeid teenindada.

Jagatud suhtluskanali näide: võrk, PCI siin ja video

Ebapraktilise kasutamise näide: IDE siin, CD ja HDD samal kanalil

Ribalaiuse probleemidele pakutavad lahendused võib jaotada kolmeks:

- koormuse jagamine
  - paralleelsete kanalite kasutuselevõtt
  - alternatiivsete kanalite kasutuselevõtt
- koormuse vähendamine
  - ebavajalike tegevuste kõrvalejätmine
- mahtude suurendamine
  - lisaseadmed
  - vanade seadmete asendamine

## Üldine jõudlus, „pudelikaelad“

Jõudlust tuleb vaadata ka üldisemast vaatepunktist: millise jõudluse annavad erinevat tüüpi komponendid (ressursid) koostöös? Lõppkasutajale on tähtis just see pool, seega peab süsteemihaldus ka selle ala katma.

Üldised jõudlusprobleemid võib jaotada kahte ossa: mingi alamkomponendi realiseerimisest tekkinud probleemid ja komponentide omavahelisest läbisaamisest tekkivad probleemid (disainiprobleemid).

Jõudluse juures kasutatakse „pudelikaela“ mõistet: see on koht süsteemis, mille jõudlus on võrreldes muu süsteemi jõudlusega väiksem ja seega see koht/detail määrab süsteemi kogujõudluse.

Toodud definitsiooni kohaselt eksisteerivad pudelikaelad kõigis süsteemides – alati leidub vähima jõudlusega komponent. Tähelepanu tuleb pudelikaelale pöörata siis, kui tema jõudlus ei ole muu süsteemiga proportsioonis, vaid oluliselt väiksem.

Pudelikaelu leitakse alamsüsteemide kaupa testides, näiteks regressiivmeetodil.

Mõnikord võib pudelikaelaks muutuda muidu korraliku jõudlusega detail, mis hetkelise tippkoormuse tõttu muudab oma töö-taktikat.

Pudelikaela likvideerimine (tegelikult pudelikael ei kao, vaid liigub komponendile, mille jõudlus on pärast esialgse pudelikaela likvideerimist väiksem) sõltub väga pudelikaela iseloomust: kas tegu on realisatsiooni- või disainiveaga, kõne alla tulevad:

- detaili või tarkvara vahetamine
- realisatsiooni parandamine
- voo ümbersuunamine
- voo reguleerimine

Pudelikaela nihutamisel varitseb meid oht, et vabanenud andmevoog tekitab uue, kriitilisema pudelikaela. Siin on tihti abiks voo reguleerimine. Kui süsteemis on mitu samaviisi väikese jõudlusega komponenti, siis puudub mõte neid ükshaaval vahetada – pudelikael jääb ikkagi samasse „piirkonda“.