

TIE-02500 rinnakkaisuus harjoitustyö

2.

Henri Laakso (240062), Niko Lappalainen (253002) ja Antti Tolonen (247589)

1 Tehtävänanto

Tehtävänantona on suunnitella rinnakkaisuutta hyödyttävä toteutus webbipalvelulle nimeltä Kissabook. Palveluun voidaan ladata kuvia, joista etsitään kissat erillisiin kuviin ja ne tallennetaan järjestelmään josta käyttäjät voivat käydä kuvia katselemassa.

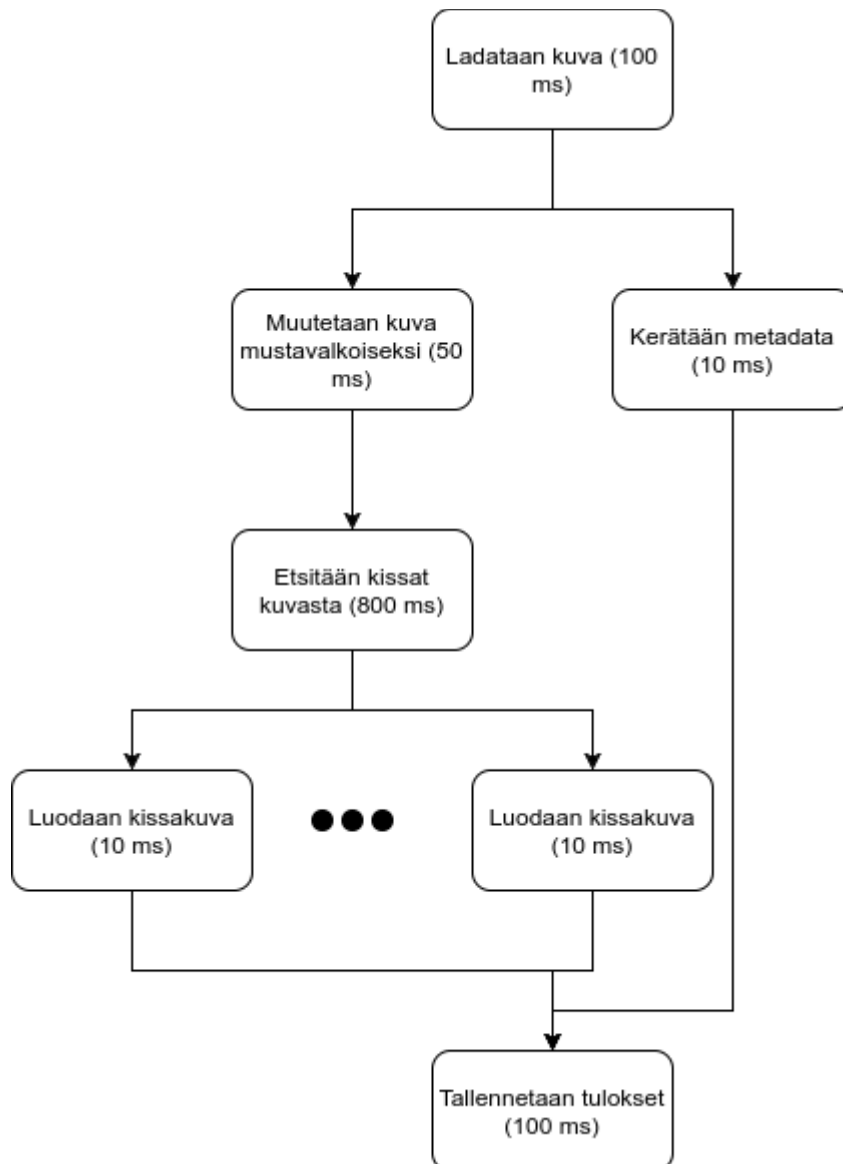
Tehtävänannossa kuvien käsittelyprosessi on kuvattu seuraavasti:

“Ohjelma suorittaa seuraavat operaatiot järjestyksessä kullekin kuvalle:

- 1. Ladataan kuva muistiin käsiteltäväksi. Keskimääräinen latausaika 100ms. Levyjärjestelmän kummallisuudesta johtuen operaatio hidastuu 500ms aina jokaista uutta säiettä kohti (siis yksi säie lataa kaksi kuvaa peräkkäin 100+100ms, mutta kaksi säiettä rinnakkain 100+500ms).*
- 2. Kerätään kuvan metadatasta GPS-koordinaatit ja muu tarvittava informaatio (10ms)*
- 3. Muutetaan kuva mustavalkoiseksi (50ms)*
- 4. Etsitään mustavalkokuvasta kaikki löytyvät kissat (800ms)*
- 5. Luodaan jokaisesta löytyneestä kissasta uusi rajattu kuva (10ms per kissa)*
- 6. Alkuperäinen kuva, metatieto ja uudet rajatut kuvat talletetaan yhdessä tietokantaan (100ms, kaikki talletetaan yhtenä tietokantakutsuna)*

Alkuperäisessä ohjelmassa kaikki em. toimenpiteet suoritetaan loppuun ennen kuin seuraava kuva otetaan käsittelyyn. Tästä johtuen palvelu on melko hidas isoille datamäärille (tuhansia kuvia per sekunti...)”

Sama prosessi voidaan kuvata Kuvassa 1 esitetyllä kaaviolla.



Kuva 1. Kaavio kuvien käsittelyprosessista

Tehtävänannon perusteella oletamme kyseessä olevan jatkuvasti pyörivä palvelu joka prosessoi kuvia mahdollisimman nopeasti sitä mukaa kun käyttäjät niitä lataavat, eikä esimerkiksi kuvia eräajona prosessoiva ohjelma.

2 Rinnakkaisuuden hyödyntäminen

2.1 Rinnakkaistettavat osat kuvien käsittelyssä

Tehtävänannon perusteella kuvien lukeminen hidastuu mikäli useammat säikeet yrittävät samanaikaisesti lukea kuvia levyltä, joten kuvien lukemisessa rinnakkaisuudesta ei voida olettaa saatavan mitään hyötyä.

Kuvien prosessointi sen jälkeen kun ne on luettu muistiin voidaan sen sijaan hyvin pitkälti rinnakkaistaa. Ilmeisemmin, eri kuvien prosessointi voidaan suorittaa rinnakkain eri suorittimilla. Lisäksi kuvankäsittelyprosessin sisällä on osia, joita voidaan suorittaa

yhtäaikaisesti. Kuten Kuvasta 1 nähdään metadatan kerääminen ja eri kissakuvien luonti voidaan suorittaa samanaikaisesti. Sen sijaan kuvan muuntaminen mustavalkoiseksi ja kissojen etsintä on pakko suorittaa peräkkäin eikä niiden suorittamisessa ole mahdollista hyödyntää rinnakkaisuutta.

3 Toteutusvaihtoehdot rinnakkaisuudelle

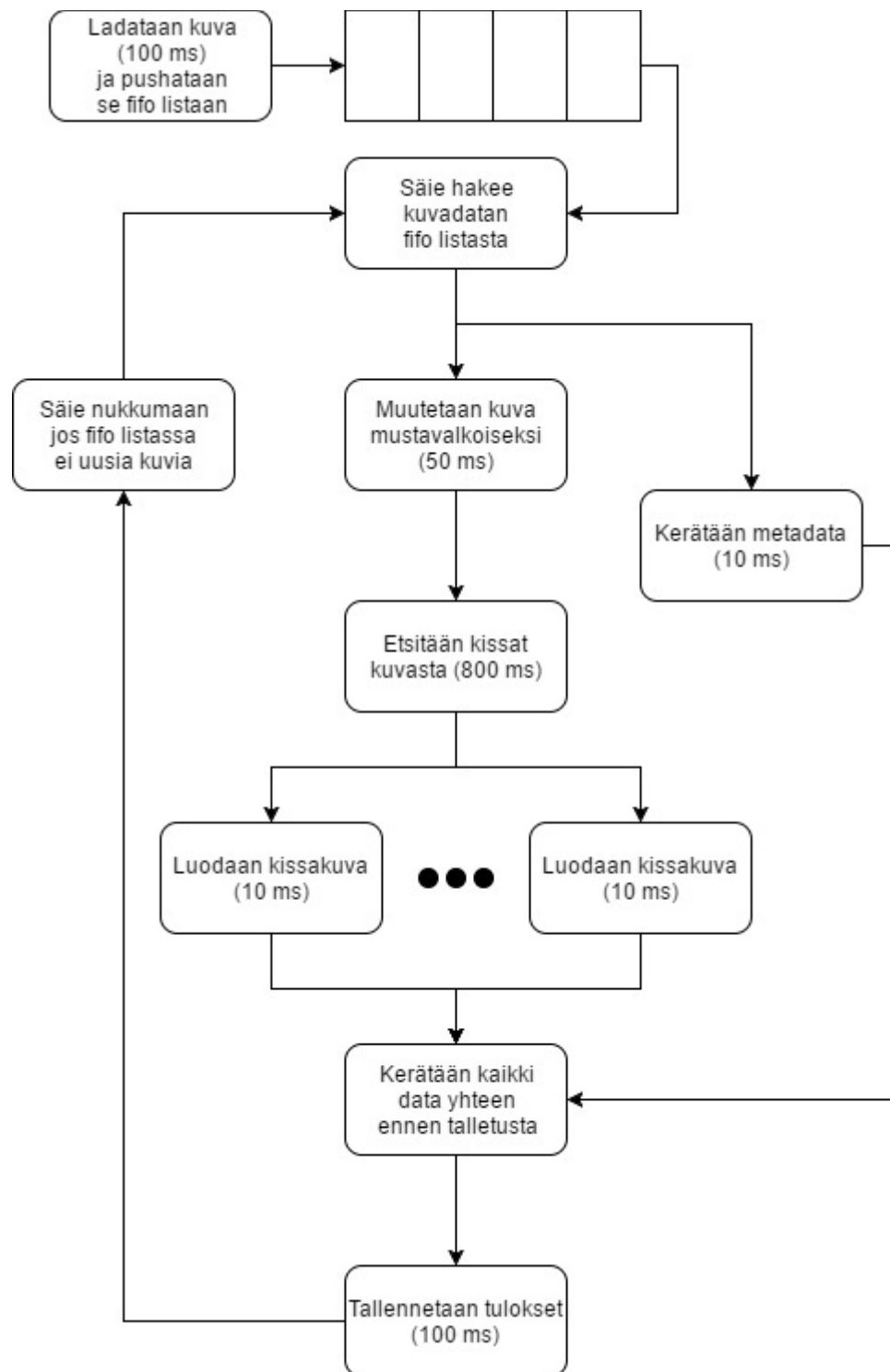
3.1 Ehdotettu toteutus

Mielestämme käytännössä toimivin ratkaisu rinnakkaisuuden toteuttamiseksi olisi niin sanottuun tuottaja-kuluttajamalliin pohjautuva ratkaisu. Tässä tapauksessa esimerkiksi kuvien lukua voisi suorittaa yksi säie, joka lataisi kuvia muistissa olevaan puskuriin.

Kuvien käsittelyn hoitaisivat kuvankäsittelysäikeet joiden tehtävä olisi käydä katsomassa onko puskurissa uusia kuvia käsiteltäväksi ja jos kuvia on saatavilla ne suorittaisivat kuvankäsittelyn ja tallentaisivat kuvat tietokantaan.

Kuvankäsittelysäikeiden järkevää määrää rajoittaa lukusäikeen toiminta. Tässä tapauksessa noin kymmen säiettä pystyy prosessoimaan kuvia samaa tahtia kuin lukusäie saa niitä luettua muistiin, joten lisäsäikeille ei löytyisi enää työtä.

Tuottaja-kuluttajamallin mukaisen toteutuksen yleisrakenne on kuvattu Kuvassa 2.



Kuva 2. Kaavio ehdotetusta ohjelma-arkkitehtuurista

3.1.1 Lukusäikeen toiminta

Lukusäie lukee kuvia levyiltä ja kirjoittaa niitä suojattuun FIFO-jonoon, puskuriin jossa on tilaa yhtä monelle kuvalle kuin on kuvankäsittelysäikeitä. Lukusäie lataa kuvia kovalevyiltä aina kun niitä on siellä saatavilla ja puskuri ei ole täynnä. Se joko pyörii luopissa jossa se haastelee onko levyllä kuvia, tai sitten se on nukkumassa kunnes kuvia levyille kirjoittava

prosessi sen herättää. Jos puskuri täyttyy, säie jää nukkumaan kunnes kuvankäsittelysäie herättää sen hakiessaan kuvan.

3.1.2 Kuvankäsittelysäikeiden toiminta

Kuvankäsittelysäikeiden tehtävä on siis lukea kuva puskurista ja tallentaa prosessoidut tulokset tietokantaan. Tämän tehtävän sisällä on osioita joita voidaan myös rinnakkaistaa (kts. Kuva 1), metadatan kerääminen voidaan tehdä rinnakkain muun kuvankäsittelyn kanssa ja eri kissakuvien luominen voidaan rinnakkaistaa.

Näiden suorittamisessa voidaan myös hyödyntää rinnakkaisuutta niin että kuvankäsittelysäie käynnistää uudet säikeet näiden operaatioiden suorittamiseksi rinnakkain ja synkronoi nämä säikeet ennen tulosten kirjoittamista tietokantaan.

Kuvankäsittelyssä erillisiä kissakuvia käsittelevien säikeiden tarvitsee päästä käsiksi yhtä aikaa käsiteltävään kuvaan, mutta koska niiden tarvitsee ainoastaan lukea kuvaa ei tästä yhtäaikaaisesta käsittelystä aiheudu ongelmia. Kissakuvia käsittelevien säikeiden täytyy myös yhtä aikaa päästä kirjoittamaan tuloksensa jonnekin, tässä samanaikaiseen kirjoittamiseen liittyvä ongelmat voidaan välttää niin että kuvankäsittelyä koordinoiva kuvankäsittelysäie antaa jokaisella kissakuvia käsittelevälle säikeelle erillisen viittauksen listan alkioon, johon kuva tulee tallettaa.

Kuvankäsittelysäie odottaa metadatasäikeen valmistumista, tallentaa metadatan tietokantakutsutietotyyppiin ja jää odottamaan kissakuvasäikeiden valmistumista. Tietokantakutsun lähetettyä säie lähettää herätteen lukusäikeelle.

Mikäli kuvankäsittelysäikeille ei ole kuvapuskurissa käsiteltävää dataa ne menevät nukkumaan. Lukusäie herättää kuvankäsittelysäikeet kun uutta dataa on saapunut käsiteltäväksi.

3.1.3 Tulosten tallennus

Tulosten tallennus tietokantaan tapahtuu tietokannanhallintajärjestelmän avulla. Tietokannanhallintajärjestelmän oletetaan huolehtivan tallennukseen liittyvistä rinnakkaisuusongelmista kuten yhtäaikaaisesta kirjoituksesta useammasta säikeestä.

3.1.4 Ohjelman toiminnan lopettaminen

Ohjelma on käytännössä palvelintyyppinen ohjelma, jonka annetaan ajaa taustalla. Ohjelman suoritus pitää olla kuitenkin lopetettavissa. Ohjelmalla on siis erillinen säie, joka odottaa lopetussignaalia ja asettaa yhteisen atomisen lopetuslipun. Tämän jälkeen säie antaa herätteen nukkuville kuvankäsittelysäikeille, jotka joko jatkavat suoritusta normaalisti jos puskurissa on kuvia tai tarkastavat lopetuslipun ja lopettavat toimintansa. Lukusäie odottaa kaikkien kuvankäsittelysäikeiden loppumista ja lopettaa lopuksi oman toimintansa.

3.2 Vaihtoehdot ehdotetulle toteutukselle

3.2.1 Sarjallisen ohjelma rakenteen monistaminen

Yksinkertaisin rinnakkainen toteutus on ajaa sarjallista toteutusta rinnakkain monessa säikeessä. Tämä kuitenkin hidastaa levyltä lukua huomattavasti joten rinnakkaisuuden hyödyt jäävät hyvin rajallisiksi.

3.2.2 Useampia kuvankäsittelysäikeitä ja isompi kuvapuskuri

Ehdotetussa toteutuksessa kuvankäsittelysäikeiden määrä ja kuvapuskurin koko on mitoitettu olettaen että kuvankäsittelysäikeet pystyvät jatkuvasti prosessoimaan kuvia samaa tahtia. Mikäli koneella on muita prosesseja jotka välillä vievät kuvankäsittelysäikeiden suoritusaikaa, on tehokkaampaa ladata kuvia puskuriiin niin paljon kuin mahdollista ja sitten kun kuvankäsittelysäikeille on tarjolla suoritusaikaa prosessoida kuvia rinnakkain mahdollisimman monella säikeellä puskurin tyhjentämiseksi.