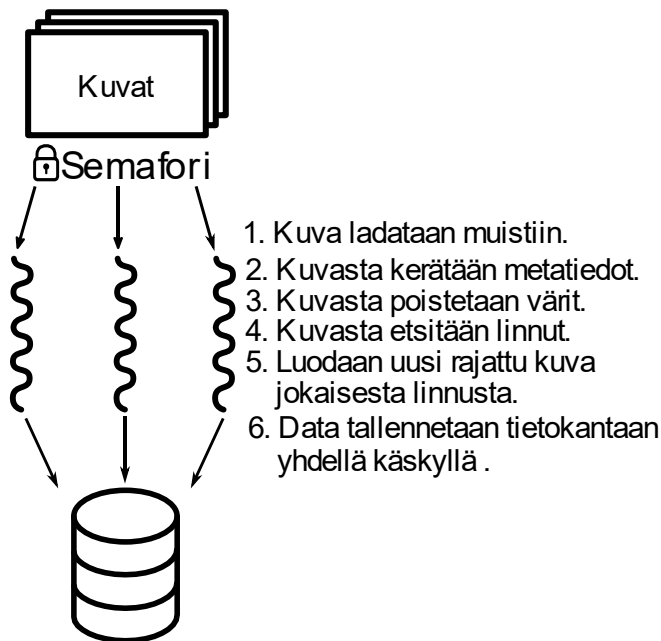


Rinnakkaisuus, työ 1 – Concurrent Design



Kuva 1. Kuva ohjelman toiminnasta. Kun säie on suorittanut kohdat 1 – 6, säie palaa kohtaan 1 aloittaen seuraavan kuvan prosessoinnin.

Kohdassa 1 säie käy lataamassa tietokannasta kuvan käsiteltäväksi ja samalla poistaa valitun kuvan sieltä. Tietokannan semaforilla mahdollistetaan vain yhden säikeen pääsyn tietokantaan kerrallaan. Tietokanta on suojattava semaforilla, jotta ohjelma toimii oikein. Esimerkiksi useampi säie voisi yrittää käsitellä samaa kuvaa tai säie on voinut valita sellaisen kuvan käsiteltäväksi, jonka jokin toinen säie on poistanut tietokannasta. Tämä estetään suojaamalla tietokanta poissulkemisella, esimerkiksi semaforin avulla.

Alkuperäistä hakutietokantaa lukevaa kriittistä aluetta ovat periaatteessa vain juuri kuvien poistamista hoitavat osiot, koska lukuoperaatio itsessään ei muuta tallennettua tietoa. Tässä mielessä ainoastaan tilanne, jossa kaksi eri säiettä yrittää poistaa ja lukea samaa kuvaa yhtä aikaa aiheuttaa ongelmia. Kuitenkin myös saman kuvan prosessointi useampaan kertaan eri säikeissä on täysin turhaa ja tämän takia myös sen tapahtuminen on estettävä.

Kohta 6, eli datan tallentaminen tietokantaan, ei ole kriittistä aluetta, jos ohjelmaa suoritetaan vain yhdellä suorittimella, sillä tallentaminen tapahtuu yhdellä käskyllä, joten ei ole mahdollisuutta, että suorittava säie vaihtuisi kesken tallennusoperaation. Jos ohjelmaa suoritettaisiin useammalla suorittimella, olisi tietokantaan tallentaminen kriittistä aluetta.

Saman kuvan vaiheita 1 - 6 ei ole järkevää jakaa eri säikeille, sillä siitä ei ole mitään hyötyä. Se vain tekisi ohjelmasta monimutkaisemman ja mahdollisesti hitaamman, sillä se lisäisi säikeiden jakamien resurssien määrää.

Eri kuvien rinnakkainen käsittely on puolestaan järkevää. Kun säie käsittelee kuvan alusta loppuun, säikeet pystyvät toimimaan pitkälti toisistaan riippumatta, jolloin niiden ei tarvitse juurikaan odottaa toisiaan tai

23.03.2020

yhteisten resurssien vapautumista. On järkevää jotenkin rinnakkaistaa ohjelma, sillä se vähentää suorittimen joutoaikaa.

Kaiken kaikkiaan yhdeltä säikeeltä menee keskimäärin 0,960 sekuntia suorittaa koko käsittelyprosessi yhdelle kuvalle. Jokaista uutta säiettä kohti kuvan käsittelyn aika kasvaa 500 ms (kuvan lataaminen muistista vie 100+400ms jokaisella ylimääräisellä säikeellä). Tästä seuraa, kuin toisen säikeen nopeus per kuva olisi 1,360 s, kolmannen 1,860 s jne. Laskennallisesti kovin suuren säiemäärän käyttö ei siis ole järkevää, sillä prosessointi aika kasvaa äkkiä jokaista uutta säiettä kohti. Säikeiden määrä tulisikin tasapainottaa siten, että sekä suorittimen joutoaikaa, että ylimääräistä suoritusaikaa kertyisi mahdollisimman vähän.

Paras tapa ohjelmassa käytettävien säikeiden määrän määrittämiseen on ohjelman tehokkuuden testaaminen eri määrillä säikeitä. Optimaaliseen säikeiden määrään vaikuttaa myös ohjelman käsittelemien kuvien määrä.