

SSD-tietoiset hakemistorakenteet

Sisällys

- SSD-muistit
- R-puu
- LCR-puu

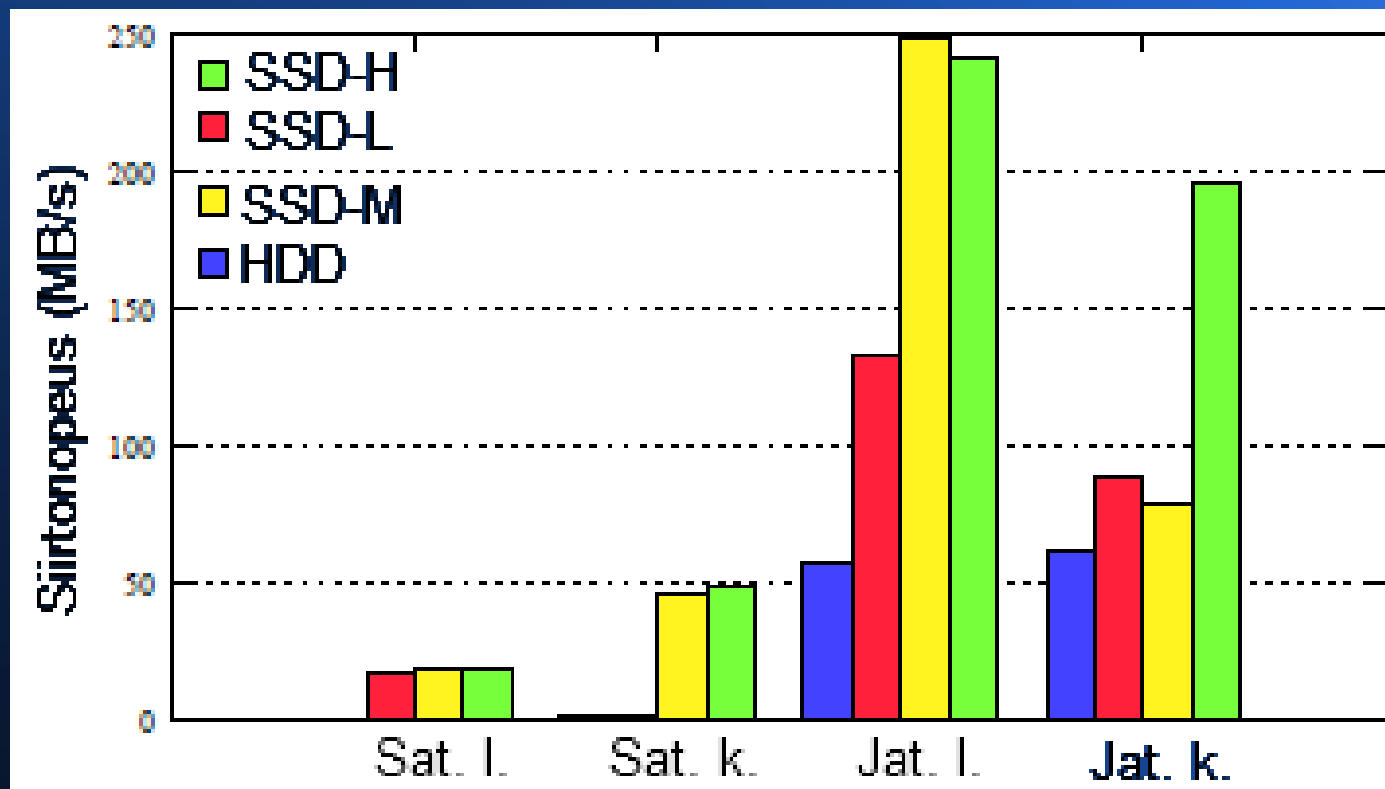
SSD-muistit

- Solid State Drive
- Yleensä flash-pohjaisia, myös muunlainen toteutus mahdollinen
- Koot kasvaneet paljon, riittäviä jo suurillekin datamäärille

Ominaisuudet

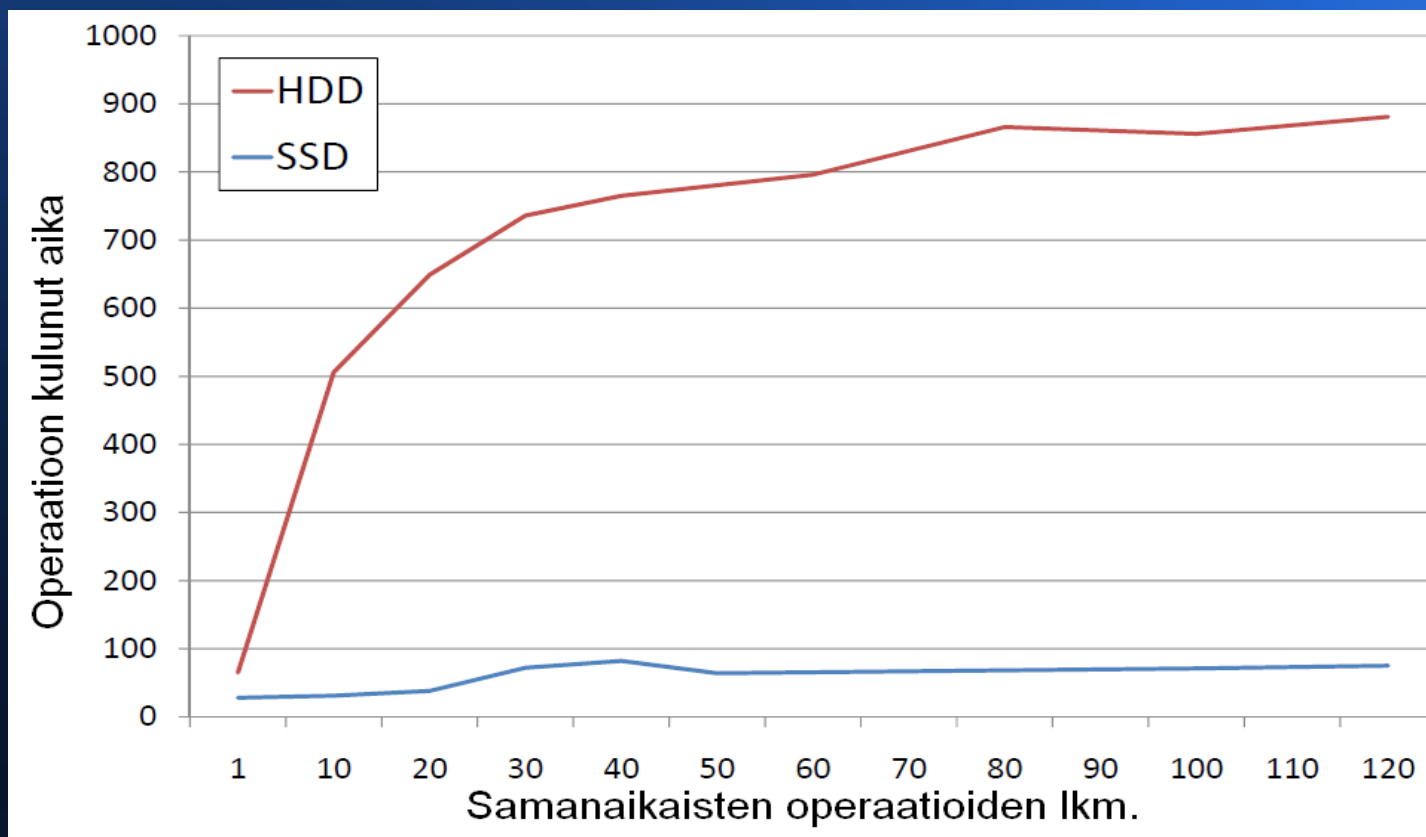
- Levymuisteja nopeampia
- Lukeminen paljon kirjoittamista nopeampaa
 - Erityisesti jatkuvilla muistialueilla
 - Muisti täytyy tyhjentää ennen kirjoittamista

Kuvakin sen kertoo



Oletus

- Pienemmät vasteajat → nopeampi hakemisto?



R-puu

- Hakemistorakenne B-puun tapaan
- Suunniteltu moniulotteisille arvoille
 - Esim. Koordinaatit
- Perustuu arvojen jakamiseen alueisiin, jotka vastaavat puun solmuja
- Yhden arvon lisäys voi aiheuttaa useita muutoksia

R-puu ja SSD

- Toimiva yhdistelmä, muttei paras mahdollinen
- R-puuhun tehtävät satunnaiset muutokset hitaita suorittaa
 - Ei ongelma levymuisteilla, SSD:llä korostuu
- Voidaan muokata paremmaksi: LCR-puu

LCR-puu

- Ei oma rakenteensa, vain lisäys R-puuhun
 - Toimii eri R-puu varianttien kanssa
- Ottaa huomioon SSD-muistien ominaisuuksia ja pyrkii hyödyntämään niitä
 - "SSD-tietoinen"
 - Kantava idea: "Muuta kirjoitukset luvuiksi"

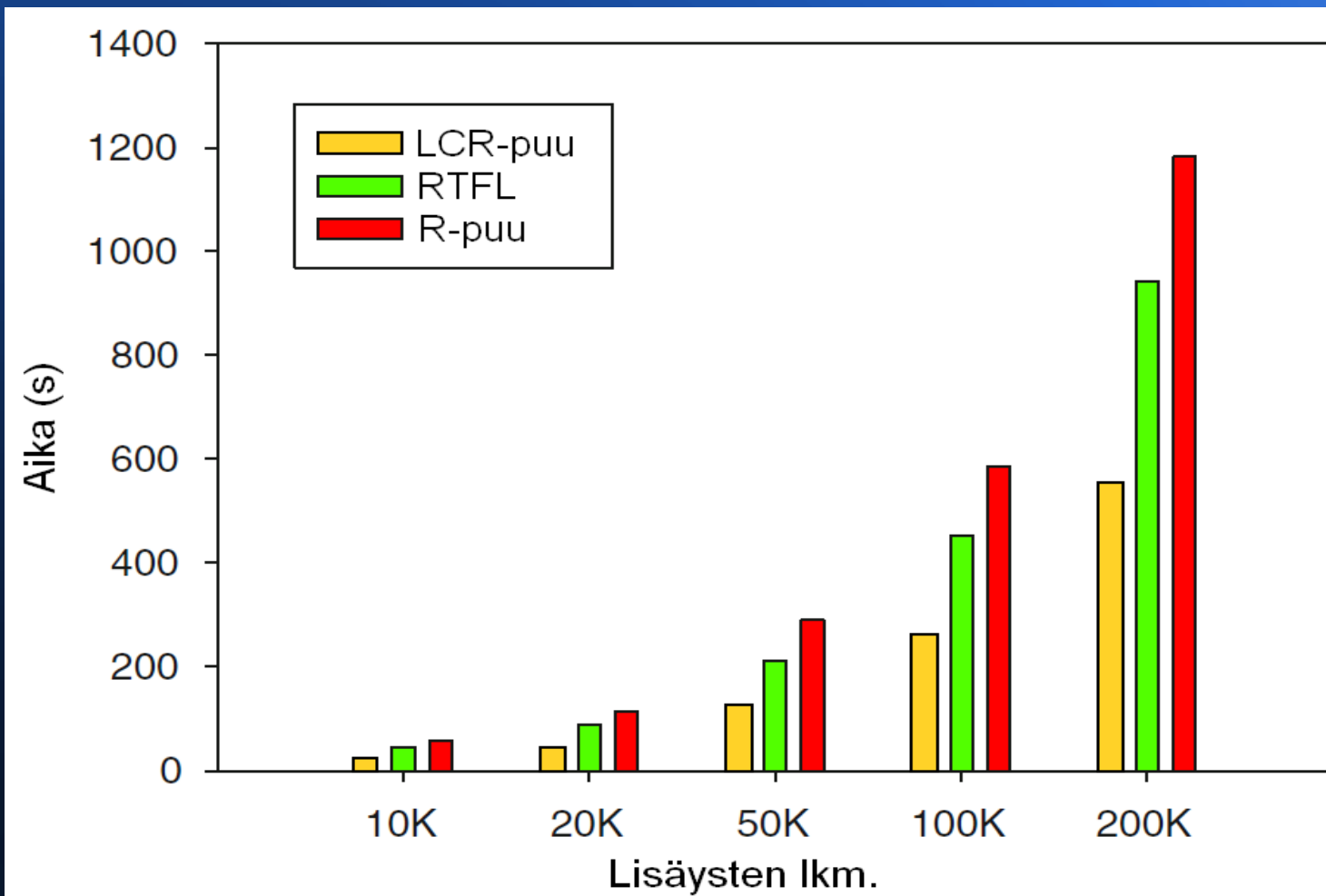
Siis mitä?

- Kokoa pienet muutokset suuremmiksi kokonaisuuksiksi lokin avulla
- Tarkastelemalla lokia ja R-puuta voidaan selvittää solmun tila
- Lokin kasvaessa tarpeeksi suureksi tallenna muutokset

Lokin muodostus

- Kokoa samaa solmua koskevat muutokset peräkkäin
- Varmista yhden solmun lokin tallentuminen yhdelle sivulle
- Tarvittaessa siirrä edellisiä solmun kirjauksia

Lisäysnopeus



Muutosnopeus

