

Seminaari keskusmuistitietokannoista

SanssouciDB ja SAP HANA

Juho Tahvanainen

Helsinki, 8.3.2012

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

| | | | |
|--|-------------------------------|---|--|
| Tiedekunta – Fakultet – Faculty | | Laitos – Institution – Department | |
| Matemaattis-luonnontieteellinen tiedekunta | | Tietojenkäsittelytieteen laitos | |
| Tekijä – Författare – Author | | | |
| Juho Tahvanainen | | | |
| Työn nimi – Arbetets titel – Title | | | |
| SanssouciDB ja SAP HANA | | | |
| Oppiaine – Läroämne – Subject | | | |
| Tietojenkäsittelytiede | | | |
| Työn laji – Arbetets art – Level | Aika – Datum – Month and year | Sivumäärä – Sidoantal – Number of pages | |
| Seminaariraportti | 8.3.2012 | 12 sivua | |
| Tiivistelmä – Referat – Abstract | | | |
| <p>Seminaarityö käsittelee liiketoimintasovelluksissa toimivien tietokantojen haasteita, sekä esittelee niiden ratkaisuna keskusmuistissa toimivan tietokannanhallintajärjestelmän uusine teknologioineen. Järjestelmistä esitellään kokeellinen SanssouciDB, sekä tuotantokäyttöön kehitetty SAP HANA.</p> <p>ACM Computing Classification System (CCS): H.2.4 [Database management], Systems</p> | | | |
| Avainsanat – Nyckelord – Keywords | | | |
| SanssouciDB, SAP HANA, keskusmuistitietokannat | | | |
| Säilytyspaikka – Förvaringställe – Where deposited | | | |
| Muita tietoja – Övriga uppgifter – Additional information | | | |

Sisältö

| | |
|---|----|
| 1 Johdanto | 4 |
| 2 Tietokannat suurissa liiketoimintasovelluksissa | 5 |
| 2.1 OLAP- ja OLTP-järjestelmien ongelmat | 5 |
| 2.2 Sarakepohjaiset tietokannat | 6 |
| 2.3 Keskusmuistitietokannat | 7 |
| 3 SanssouciDB | 7 |
| 3.1 Esittely..... | 7 |
| 3.2 Tekniset yksityiskohdat | 8 |
| 3.2.1 Rinnakkaisuus | 8 |
| 3.2.2 Tiedon pakkaaminen | 9 |
| 3.2.3 Tiedon pysyvä tallentaminen | 9 |
| 4 Yhteenveto | 11 |
| Lähteet..... | 12 |

1 Johdanto

Tiedolla on suuri, jatkuvasti kasvava merkitys nykyaikaiselle liiketoiminnalle. Yritykset pyrkivät tekemään päätöksensä perustuen lukuisten erilaisten sovellusten tarjoamaan tietoon. Liiketoimintasovellusten kehittyessä niiden tuottaman tiedon määrä kasvaa kiihtyvällä vauhdilla [Pla11]. Yritys saattaa haluta suorittaa valtaviin tietomassoihin perustuvan markkinatilanneanalyysin, tai järjestelmän säännöllisin väliajoin suoritettaviin toimintoihin saattaa kuulua tietointensiivistä toimintaa, kun tarvesuunnittelua. Näin raskaat toimenpiteet vaativat tietokannalta paljon. Ensinnäkin, tietokannan tulee pystyä säilyttämään järjestelmän tuottama ja tarvitseva valtava tietomassa. Järjestelmien tiedontallennuskapasiteetin ollessa jo kehittynyt pisteeseen, että magneettista tallennusmediaa voidaan lisätä käytännössä rajattomasti, voidaan tämän kysymyksen sanoa olevan ratkaistu. Toinen nykyaikaiselta järjestelmältä haluttu vaatimus on tiedon nopea käsittely. Tämä vaatimus on huomattavasti hankalampi toteuttaa kuin pelkkä tallennuskapasiteetin kasvattaminen [Gra06].

On kuitenkin huojentavaa todeta, että magneettisen tallennusmedian ohella myös keskusmuistin hinta on laskenut tallennuskapasiteetin samalla kasvaessa [Pla09]. Tietonsa keskusmuistiin levyn sijaan tallentavia, lukemiseen sekä kirjoittamiseen soveltuvia tietokantajärjestelmiä on ehdotettu ratkaisuksi tähän ongelmaan [Pla11][Kea11]. Keskusmuistissa sijaitsevan tiedon lukeminen on tuhansia kertoja levyllä sijaitsevan tiedon lukemista nopeampaa. Edes välimuistin käyttö ei tuo levyllä sijaitsevan tiedon luku-/kirjoitusnopeutta lähellekään keskusmuistia. Toinen (myös keskusmuistitietokannoissa käytetty) nopeutuskeino on tietokannan muuttaminen riviperustaisesta sarakeperustaiseen [Sto05]. Liiketoimintasovellusten on usein havaittu hakevan tietoa sarakkeen perusteella rivin sijaan, joten on perusteltua tallentaa tieto sarake kerrallaan rivin sijasta [Pla11].

Hasso Plattnerin tutkimusryhmän Potsdamin yliopistossa kehittämä kokeellinen SanssouciDB on esimerkki keskusmuistissa toimivasta sarakepohjaisesta tietokannanhallintajärjestelmästä [Pla11]. SanssouciDB toimii hajautetusti usealla palvelimella muodostaen useista keskusmuisteista yhden yhtenäisen tallennusalueen. SanssouciDB:n suunnittelussa on myös pyritty täyttämään liiketoimintasovelluksilta vaadittuja ominaisuuksia, joita tarvitaan sekä analytiikassa että operatiivisessa toiminnassa.

SAP AG:n keskusmuistitietokantaratkaisu SAP HANA muistuttaa teknisiltä ratkaisuiltaan SanssouciDB:tä hyvin läheisesti [Fär11]. Aloittaen analytiikkasovelluksia, on SAP:lla lopullinen

pyrkimys levittää HANA:n tarjoama keskusmuistitietokantaratkaisu kaikkialle tarjoamiinsa liiketoimintasovelluksiin..

2 Tietokannat suurissa liiketoimintasovelluksissa

Tietoa käsittelevät liiketoimintasovellukset ovat perinteisesti säilyttäneet tietoa kahdessa eri paikassa sen käyttötarkoituksen mukaan: kirjoitettavaa sekä luettavaa transaktiotietoa säilytetään perinteisessä tietokannanhallintajärjestelmässä (OLTP), ja vain luettavaa, analytiikkaan käytettävää tietoa säilytetään tietovarastossa (OLAP) [Jac09]. Näiden rakenteita ja toimintaa on pyritty optimoimaan tehtäviinsä sopiviksi. Esimerkiksi SAP-järjestelmissä tämä jako menee niin, että toimitusketjun optimointiin ja logistiikan ohjaamiseen käytettävä SAP SCM-järjestelmä lukee ja kirjoittaa omaan tietokantaansa [Sap12a], ja raportointiin sekä analytiikkaan käytettävä SAP BI lukee omaansa [Sap12b]. Tämä dualistinen arkkitehtuuri tuo mukanaan suorituskyvyllisiä hyötyjä, kun transaktiotietoa sisältävää tietokantaa ei kuormiteta analytiikalla. Erillisten tietokantojen rakenteet voidaan myös suunnitella optimaalisiksi omiin tehtäviinsä. Tässä arkkitehtuurissa on myös ongelmia, kuten seuraava kappale kertoo.

2.1 OLAP- ja OLTP-järjestelmien ongelmat

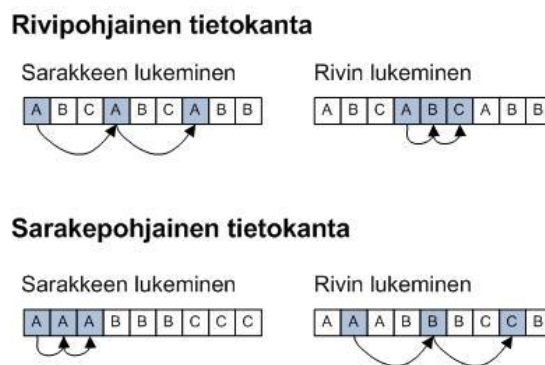
Niin sanottu ”big data”, eli nykyaikaisten tietointensiivisten liiketoimintasovellusten tuottaman tietomäärän räjähdysmäinen kasvu on luonut tarpeen tehostaa tiedonhallintaa [Jac09]. Edellisessä kappaleessa mainittu jako OLAP- ja OLTP-järjestelmiin on yksi ratkaisusta. Sillä on kuitenkin mainitun tiedon kahdentamisen lisäksi myös muitakin ongelmia. Yksi näistä on päivittäiseen toimintaan tarvittavan tiedon määrä. Tietovarasto ei tarjoa helpotusta tilanteisiin, joissa muokattavan tiedon määrä kasvaa suureksi [Pla11]. Esimerkiksi logistiikan toiminnanohjausjärjestelmän tuotekohtainen ohjaustaulu saattaa hyvinkin sisältää kymmeniä miljoonia rivejä jatkuvasti muokattavaa tietoa. Tällaisen jatkuvasti muuttuvan tietomassan samanaikainen lukeminen ja kirjoittaminen on haasteellista.

Tietovarastoille on myös luonteenomaista niistä haluttujen raporttien ennakkomäärittely. Tällä nopeutetaan usein käytettyjen raporttien tuottamista, mutta kokonaan uusien raporttien laatiminen vaatii uutta kehitystyötä, eikä se onnistu nopeasti. Muuttuvissa markkinatilanteissa toimiminen saattaa vaatia reaaliaikaiseen tietoon perustuvia aiemmin määrittelemättömiä raportteja lyhyellä varoitusajalla [Pla11].

Tietokantoja voidaan luonnollisesti optimoida myös perinteisillä menetelmillä, kuten hakemistoilla ja säännöllisellä arkistoinnilla, mutta ohjelmistojen kasvavien liiketoimintavaatimusten vuoksi nämä menetelmät eivät kuitenkaan ole riittäviä. Tietokannanhallintajärjestelmän muuttaminen riviperusteisesta sarakeperusteiseen, sekä tietokannan siirtäminen magneettilevyiltä keskusmuistiin ovat mullistavia menetelmiä, joilla voidaan saada huomattavia suorituskykyetuja [Pla09].

2.2 Sarakepohjaiset tietokannat

Perinteiset tietokannanhallintajärjestelmät tallentavat tietokannan sisällön levyille rivi kerrallaan [Sto05]. Tällainen järjestelmä sopii erityisen hyvin yksittäisen rivien lisäämiseen tietokantaan, kuten myös tilanteisiin joissa tietoa haetaan tietokannasta rivi kerrallaan. Käytännön kokemukset ovat kuitenkin osoittaneet, että tietokannat sisältävät usein suuren määrän sarakkeita, joista vain pientä osaa tarvitaan. Esimerkiksi SAP ERP-järjestelmän verkostopistekohtaista tuotetietoa sisältävä MARC-taulu sisältää kymmeniä sarakkeita, joista esimerkiksi tarvesuunnittelua tehtäessä luetaan vain muutamaa. Tiedon tallentaminen levyille sarake kerrallaan nopeuttaa tämänkaltaisten suurten, rajattua sarakemäärä käyttävien toimenpiteiden suorittamista. Kuva 1 esittää tiedon lukujärjestystä rivi- ja sarakepohjaisissa tietokannoissa.



Kuva 1: Tiedon lukujärjestys rivi- ja sarakepohjaisissa tietokannoissa.

Sarakkeina rivien sijaan tallennettavaa tietoa on myös tehokkaampaa pakata pienempään tilaan, mikä osoittautuu erittäin hyödylliseksi keskusmuistitietokantojen kohdalla. OLAP-mallin tietovarastot ovat jo kauan käyttäneet sarakeperusteista tietokantaratkaisua [Pla09]. Uusiin, liiketoimintakäytössä tehtyihin yleisimpiin tietokantakyselyihin perustuvat tutkimukset ovat osoittaneet, että oikein optimoituina myös OLTP-tyypin operatiiviset tietokannat voivat hyötyä sarakepohjaisuudesta [Pla09].

2.3 Keskusmuistitietokannat

Keskusmuistitietokannalla tarkoitetaan tietokantajärjestelmää, joka käyttää tietonsa ensisijaisena tallennusmedianana keskusmuistia. Siirtämällä tietokanta magneettiselta tallennusmedialta keskusmuistiin saavutetaan merkittävää suorituskykyhyötyä, kun tiedonsiirtoon kiintolevyltä keskusmuistiin ei kulu aikaa. Magneetti- tai SSD-levy ottaa varmuuskopiointiin käytettävän tallennusmedian roolin keskusmuistin toimiessa tiedon operatiivisena säilytyspaikkana. Kasvanut suorituskyky mahdollistaa kokonaan uusien liiketoimintasovellusten kehittämisen. Jopa työpöytäsovellukset kuten taulukkolaskenta hyötyvät tiedon siirtämisestä keskusmuistiin [Pla11]. Muutos ei tule tapahtumaan kivuttomasti, sillä myös vanhat sovellukset on toteutettava uudestaan, mikäli ne pyritään siirtämään käyttämään keskusmuistipohjaista tietokantaa [Loo11]. Ensimmäiset keskusmuistia ensisijaisena tallennusmedianana käyttävät tietokannanhallintajärjestelmät ovat tulleet markkinoille parin viime vuoden aikana [Loo11].

3 SanssouciDB

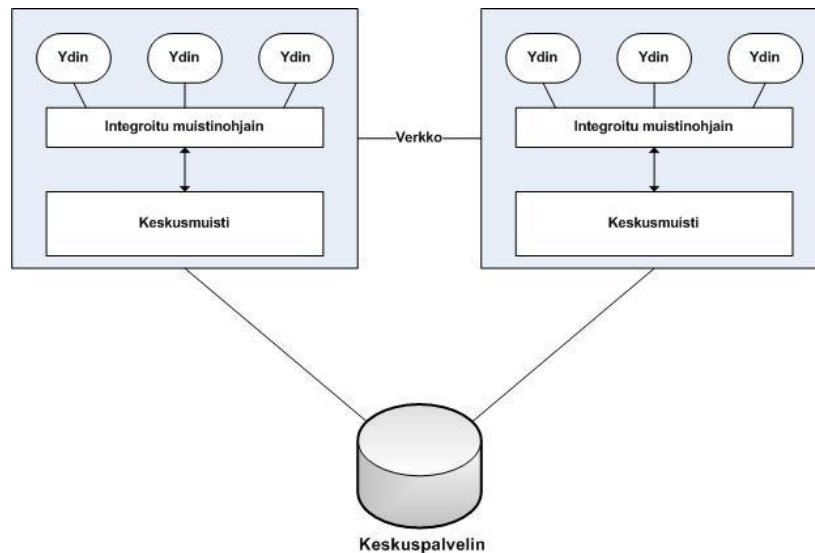
Potsdamin yliopiston Hasso Plattner-instituutissa kehitetty SanssouciDB on kokeellinen esimerkki keskusmuistitietokannoista. SanssouciDB:n tavoitteena on toimia keskusmuistitietokannoissa käytettyjen teknologioiden esiselvityksenä, ja tarjota teoreettista sekä kokeiltua pohjatietoa todellisille liiketoiminnassa käytettäville tietokannanhallintajärjestelmille [Pla11]. SAP:n vuonna 2010 maailmanlaajuisesti julkaisema SAP HANA toteuttaa SanssouciDB:ssä kuvattuja teknologiaratkaisuja käytännön tasolla [Fär11].

3.1 Esittely

SanssouciDB on hajautettu, keskusmuistia ensisijaisena tallennuspaikkanaan käyttävä tietokannanhallintajärjestelmä. SanssouciDB:ssä relaatiomuotoisena tallennettua tietoa hajautetaan usean keskenään identtisen koneen keskusmuistiin niin, että jokainen koneista vastaa yksin omasta osastaan kokonaistietomäärästä, eli kyseessä on ns. shared nothing-arkkitehtuuri. Yhden palvelimen sisäinen toiminta tapahtuu shared-memory -arkkitehtuurin mukaisesti suuren määrän prosessoriytimiä lukiessa ja kirjoittaessa jaettuun, yhteiseen muistiin. SSD-levyä tallennusmedianaan käyttävä keskuspalvelin vastaa järjestelmän toiminnan koordinoinnista sekä tiedon varmuuskopiointista [Pla11]. Kukin yksittäisistä palvelimista sisältää useita prosessoriytimiä, sekä suhteellisen suuren määrän keskusmuistia. Kuva 2 esittää järjestelmän arkkitehtuuria. Hajautettu arkkitehtuuri mahdollistaa järjestelmän skaalautumisen erilaisiin- ja kokoiisiin tarkoituksiin. Esimerkiksi suuri, 25 kappaletta 64-ytimisiä kahden teratavun

keskusmuistilla varustettuja palvelimia kykenee sisältämään maailman suurimpien yritysten toiminnassaan käyttämän tiedon [Pla11].

SanssouciDB:ssä hyödynnetään kappaleessa 2.2 esiteltyä tiedon sarakepohjaista tallentamista. Koska järjestelmän halutaan soveltuvan hyvin myös tiedon rivipohjaiseen käsittelyyn, on siinä käytetty rivi- ja sarakepohjaista hybridikäytäntöä; tieto tallennetaan sarakepohjaisesti niin, että usein käytetyt arvoyhdistelmät tallennetaan peräkkäin [Pla11].



Kuva 2: SanssouciDB:n arkkitehtuuri.

3.2 Tekniset yksityiskohdat

SanssouciDB sisältää rivi/sarake –hybridirakenteen lisäksi muitakin merkittäviä teknisiä ratkaisuja. Tässä alakappaleessa tutkitaan niistä tarkemmin tiedon rinnakkaista käsittelyä ja pakkaamista. Myös tiedon pysyvään säilyttämiseen SSD-levyllä luodaan katsaus.

3.2.1 Rinnakkaisuus

Suorittimien kehitys on viimevuosina siirtynyt kellotaajuuden nostosta rinnakkaisten ydinten määrän kasvattamiseen. SanssouciDB:ssä rinnakkaisuutta hyödynnetään sekä ydinten että palvelimien välillä [Pla11]. Tämä tarkoittaa käytännössä tehtävien pilkkomista mahdollisimman hyvin yksittäisille koneille suoritettaviksi, sekä yhden palvelimen sisäisen toiminnan optimointia usealle ytimelle sopivaksi jaetun muistin sekä prosessien kontrolloidun määrän avulla.

Yhden SanssouciDB-palvelimen sisältäessä suuren määrän ytimiä, voidaan prosessien määrä yhtä ydintä kohti rajoittaa yhteen. Tällä ratkaisulla vältetään prosessinhallintasta koitua ylimääräinen kuorma, kun ydin ei joudu vuorottelemaan usean prosessin välillä. SanssouciDB:ssä käytetty

rivi/sarake –hybridirakenne hyötyy vielä erikseen ytimen SSE-käskyistä, joilla voidaan mahduttaa usean operaation tulos yhteen ytimen rekisteriin. Näin usea arvo voidaan hakea muistista käsiteltäväksi samalla kertaa, joka hyödyttää rivi/sarake-hybridirakennetta [Pla11].

Jotta järjestelmä kykenisi hyödyntämään rinnakkaisuutta täysimittaisesti, on myös sen ohjelmakoodin oltava sopivaa siihen. SanssouciDB:n tietoa käsittelevät algoritmit on suunniteltu sen mukaan, toimivatko ne palvelinten vai ydinten välillä. Palvelinten käsitellessä yksin omistamaansa tietoa, ja ydinten viitatessa jaettuun tietoon, on nämä erot huomioitava algoritmien ja ohjelmakoodin toteutuksessa [Pla11].

3.2.2 Tiedon pakkaaminen

Suurimmat tietokannat saattavat sisältää jopa useita petatavuja tietoa. Vaikka keskusmuistin määrä kasvaa, ja hinta halpenee samalla jatkuvasti, on suurien tietokantojen mahduttaminen edes hajautettuun keskusmuistiin haastavaa. SanssouciDB:ssä ratkaisuna käytetään tiedon pakkaamista. Käytetty rivi/sarake –hybridirakenne sopii erityisen hyvin pakattavaksi, sillä siinä samankaltainen tieto (yhden sarakkeen arvot) sijaitsee lähellä toisiaan [Pla11].

SanssouciDB:ssä käytetään lähtökohtaisesti kevyitä pakkaustekniikoita. Raskaammat, enemmän suoritinaikaa vievät pakkaustekniikat saattavat enemmissä määrin käytettyinä hidastaa suuren tietomassan käsittelyä [Pla11]. Kevyitä pakkaustekniikoita ovat esimerkiksi sanastojen käyttö usein toistuville merkkijonoille, peräkkäisten merkkijonojen ja niiden yhdistelmien merkitseminen yhteisellä tunnuksella ja sen kertoimella, tarpeettomien nollien tiputtaminen tietotyypeistä, ja niin edelleen. Tehokkain pakkaustekniikka on sidoksissa vahvasti siihen, minkälaisessa muodossa tieto on tallentunut kantaan [Pla11].

3.2.3 Tiedon pysyvä tallentaminen

Ensimmäisiä keskusmuistitietokannoista mieleen tulevia haasteita on tiedon jatkuvuus; keskusmuistiin tallennettu tieto katoaa virran katketessa. On täten selvää, että tietoa on säilytettävä keskusmuistin ohella myös pysyvämmässä tallennusmediassa. SanssouciDB:ssä SSD-levyjä käyttävä keskuspalvelin vastaa tiedon pysyväästä säilytyksestä ja lokituksesta [Pla11]. Haasteena SanssouciDB:ssä muiden keskusmuistitietokantojen tavoin on tämän säännöllisen toimenpiteen ajoitus: liian usein tehty kopiointi hidastaa järjestää, mutta toisaalta liian harvoin tehty kopiointi vaikeuttaa virhetilanteista toipumista. SanssouciDB ratkaisee tämän priorisoimalla usein käytetyt osat tietokantaa useammin kopioitaviksi, ja harvemmin käytetyt osat harvemmin kopioitaviksi [Pla11].

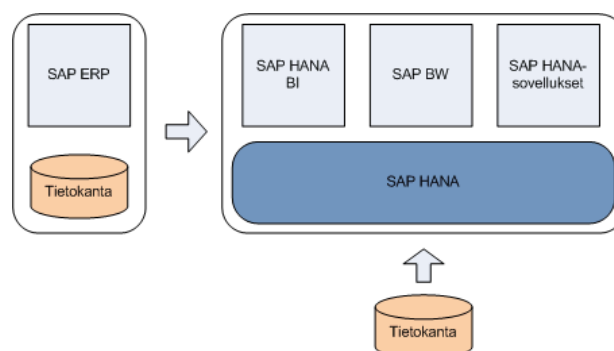
Rivi/sarake-pohjaista tietoa sisältävä tietokanta, kirjoitusoptimoitua tietoa sisältävä välimuisti, sekä tehdyt muutokset sisältävä loki ovat ne osat, jotka pitää palauttaa virhetilanteen sattuessa. Itse tietokanta tallennetaan välimuistin kanssa yhteenkoottuna snapshot-tyylillä SSD-levylle määrätyn väliajoin. Myös loki tallennetaan SSD-levylle. Lokitusta helpottaa insert only –periaate: kaikki tehdyt muutokset luovat uuden rivin tietokantaan, ja vanhat rivit jätetään aikaleiman avulla huomioimatta. Tämä yksinkertaistaa lokin kirjoittamista ja lukemista [Pla11].

3.3 Jatkumo – SAP HANA

SanssouciDB:n laatinut tutkimusryhmä on tiukasti sidoksissa SAP AG:hen, onhan sen päähahmo Hasso Plattner myös yksi SAP:n perustajista. Täten ei ole yllättävää, että SanssouciDB:ssä esiteltyjä ominaisuuksia esiintyy SAP:n vuonna 2010 julkaisemassa avauksessaan keskusmuistitietokantojen markkinoille, SAP HANA:ssa [Fär11]. SAP on panostanut viime vuosina HANA:an huomattavan paljon, ja ensimmäiset HANA-natiivit sovellukset ovat ilmestyneet markkinoille vuosien 2011 ja 2012 aikana [Sap12c].

Tällä hetkellä HANA toimii erillisenä korkean suorituskyvyn analytiikkatietokantana. Tietosisältö replikoidaan yrityksen päätietokannasta HANA:an, jossa siitä voidaan muodostaa ad hoc-raportteja. Maaliskuussa 2012 HANA:a hyödyntävien sovellusten määrä on vielä varsin rajallinen [Sap12c]. Järjestelmän lopullinen tavoite on korvata operatiivisten SAP-järjestelmien tietokannat kokonaan, jolloin kaikki järjestelmissä käsiteltävä tieto sijaitsee yhdessä nopeassa tietokannassa [Fär11]. Tämä mahdollistaa reaaliaikaiseen tietoon perustuvan analytiikan, sekä tavanomaisten liiketoimintasovellusten suorituskyvyn merkittävän parannuksen [Pla11]. Tavoitteen saavuttaminen vie kuitenkin aikaa, ja lähitulevaisuudessa todennäköisempi konfiguraatio on analytiikan korvaaminen HANA:lla operatiivisen puolen jatkaessa itsenäisesti entisellään (kuva 3) [Fär11].

Vaikka HANA kykenee toimimaan myös muiden kuin SAP:n toimittamien järjestelmien kanssa, suurin hyöty siitä saadaan SAP-sovellusten kanssa, koska HANA sisältää useita SAP-spesifisiä käskyjä syvälle tietokantaan integroituina [Fär11].



Kuva 3: Esimerkkikonfiguraatio: SAP HANA-pohjainen analytiikka, erillinen SAP ERP-järjestelmä sekä ulkoinen tietokanta.

SanssouciDB:ssä esitellyistä teknisistä ratkaisuista HANA:ssa toteutetaan muun muassa aiemmin mainitut shared-nothing tiedon hajauttaminen usealle moniytimiselle palvelimelle, tietokannan rivi/sarake –hybridirakenne, relaatioiden sisällön pakkaaminen, sekä keskitetty SSD-pohjainen varmuuskopio/ohjauspalvelin. Mainittava on myös HANA:n ”beynd SQL”-ominaisuudet: mahdollisimman suuri osa tiedonkäsittelystä pyritään siirtämään lähelle tietoa, eli sinne missä sen toiminta on tehokkainta. Sumeat SQL-hakulausekkeet ja kyselyihin upotetut osaset ohjelmakoodia ovat esimerkkejä tästä [Fär11].

4 Yhteenveto

Modernien liiketoimintasovellusten laajeneva toimintaympäristö ja kasvavat suorituskykyvaatimukset ovat luoneet tarpeen kehittää vaihtoehtoja perinteiselle OLTP/OLAP-arkkitehtuurille. Käsiteltävän tiedon määrän kasvu, sekä halu pyrkiä kohti reaaliaikaista analytiikkaa ja tehokasta operatiivista toimintaa sanelevat tulevaisuuden kehityspolut. Keskusmuistiin tietonsa tallentavat tietokannanhallintajärjestelmät ovat tällä hetkellä vahvin ehdokas korvaamaan muut tietokantaratkaisut.

Kokeellinen keskusmuistitietokanta SanssouciDB esittelee useita teknisiä ratkaisuja, joita voidaan hyödyntää varsinaisissa tuotantokäyttöön tarkoitetuissa järjestelmissä. Verrattaessa perinteisiin tietokantoihin, merkittävinä eroina tiedon keskusmuistiin siirtämisen lisäksi voidaan mainita siirtyminen riveistä rivi/sarakepohjaisuuteen, relaatiotiedon pakkaaminen sekä toimintalogiikan suorittaminen lähellä tietoa.

SAP:n vuonna 2010 julkaisema SAP HANA on yhtiön lyhyen ja pitkän tähtäimen suunnitelma keskusmuistitietokantojen käyttöönottoa varten. HANA toteuttaa useita SanssouciDB:ssä esitetyjä ominaisuuksia keskittyen aluksi puhtaaseen analytiikkaan. SanssouciDB:ssäkin mainittu analytiikan ja operatiivisen tiedonkäsittelyn yhdistäminen kuuluu sekin HANA:n tulevaisuuteen.

Siirtymä nykyjärjestelmistä puhtaasti keskusmuistissa toimiviin, OLAP & OLTP:n yhdistäviin tietokantoihin tulee olemaan pitkä. Vaikka niillä voidaan saavuttaa kiistattomia etuja, ja SAP HANA:n kaltaisia tuotteita on ilmestynyt markkinoille, on muutos erityisesti operatiivisella puolella hankala nykyisten sovellusten uudelleentoteuttamistarpeen vuoksi. Analytiikka onkin se alue, jolla keskusmuistitietokantojen voidaan nähdä yleistyvän suhteellisen nopeasti. Tulevaisuuden voimme

kuitenkin nähdä kuuluvan analytiikan ja operatiiviset tarpeet toteuttaville keskusmuistitietokannoille.

Lähteet

- Fär11 Färber, F., SAP HANA Database - Data Management for Modern Business Applications. *SIGMOD Record*, 40, 12 (2011), 45-51.
- Gra06 Gray, J., Tape is Dead Disk is Tape Flash is Disk RAM Locality is King, http://research.microsoft.com/~gray/talks/Flash_is_Good.ppt [8.3.2011]
- Jac09 Jacobs, A., The Pathologies of Big Data. *Communications of the ACM*, 50, 8 (2009), 36-44.
- Kea11 Keall, C., In-memory databases - the next big thing? , <http://www.nbr.co.nz/article/memory-databases-next-big-thing-ck-96642> [8.3.2011]
- Loo11 Loos, H. et al., In-memory databases in business information systems. *Business & Information Systems Engineering*, 6 (2011), 389-395.
- Pla09 Plattner, H., A common database approach for OLTP and OLAP using an in-memory column database. *SIGMOD'09*, Rhode Island, USA, 2009.
- Pla11 Plattner, H., Zeier, A. *In-memory data management*. Heidelberg, Saksa, 2011.
- Sap12a SAP Supply chain management. <http://www.sap.com/solutions/business-suite/scm/index.epx> [8.3.2012]
- Sap12b SAP Netweaver business warehouse. <http://www.sap.com/solutions/sapbusinessobjects/data-warehousing/sapnetweaver-business-warehouse/index.epx> [8.3.2012]
- Sap12c SAP HANA. <http://www.sap.com/hana/apps-by-business/index.epx> [8.3.2012]
- Sto05 Stonebraker, M. et al, C-Store: A Column-oriented DBMS. *Proceedings of the 31st VLDB Conference*, Trondheim, Norja, 2005.