

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikka
Tietoliikennetekniikka

Tutkintotyö

Pasi Huotari

Oracle-pohjainen tietojärjestelmä globaalissa ympäristössä

Työn ohjaaja
Työn teettäjä
Tampere 2008

lehtori Jorma Peltoniemi
Metso Minerals Oy, valvojana kehityspäällikkö Jarkko Kilponen

TAMPEREEN AMMATTIKORKEAKOULU

Tietotekniikka

Tietoliikennetekniikka

Huotari, Pasi

Oracle-pohjainen tietojärjestelmä globaalissa ympäristössä

Tutkintotyö

65 sivua

Työn ohjaaja

Jarkko Kilponen

Työn teettäjä

Metso Minerals Oy

Huhtikuu 2008

Hakusanat

Oracle, Tietokanta, Klusteri, RAC, HP-UX

TIIVISTELMÄ

Tässä työssä tutkittiin mahdollisia ratkaisuja aikaansaada tietojärjestelmän saatavuudelle sekä vasteajalle taso, joka kriittiseltä järjestelmältä vaaditaan nykyaikaisessa globaalissa yritys ympäristössä. Lisäksi työn tavoitteena oli kuvata yleisesti tietojärjestelmän toiminta yrityksen sisäiseen käyttöön. Tutkittavana järjestelmänä oli Aton PDM -järjestelmä, jota käytetään tuotetiedon hallintaan Metso Minerals Oy:ssä. Työssä esitellään mm. palvelinten arkkitehtuuria sekä rajapintoja muihin järjestelmiin. Järjestelmään liittyy olennaisena osana Oracle-tietokanta.

Vasteaikaan vaikuttavana merkittävänä tekijänä tutkittiin Oracle-tietokannanhallintaohjelmiston sisäisiä muistiparametrejä ja kuinka näitä muuttamalla voidaan saada järjestelmää tehokkaammaksi. Myös tietokannan indeksoinnilla ja analysoinnilla todettiin olevan merkitystä vasteaikoihin. Oracle-osuudessa tarkasteltiin myös Oraclen arkkitehtuuria sekä erilaisia tiedon varmistusmenetelmiä.

Korkealla saatavuudella pyritään minimoimaan menetetty aika mahdollisessa järjestelmän virhetilanteissa. Tämän toteuttamiseksi työssä tutkittiin erilaisia klusterijärjestelmiä, joissa useampi palvelin liitetään järjestelmään työskentelemään yhdessä. Klusteroinnilla voidaan saavuttaa järjestelmän korkea saatavuus tasaamalla asiakasohjelmilta tuleva kuorma useille palvelimille. Vaihtoehtoisesti klusteroinnin avulla voidaan toisen palvelimen vikaantuessa ottaa toinen palvelin käyttöön. On myös mahdollisuus suorittaa molempia em. tapoja yhtäaikaaisesti. Tutkittavat klusterijärjestelmät olivat Oracle RAC, Windows Server 2003, HP-UX, joista kaksi viimeisintä ovat samalla käyttöjärjestelmiä. Käsitellyt klusterijärjestelmät eivät ole suoraan yhteensopivia Aton PDM:n kanssa. Klusteroinnin toteuttaminen järkevästi aiheuttaisikin ohjelmistoräätälöintejä Atonin ohjelmistontoimittajalta.

TAMK UNIVERSITY OF APPLIED SCIENCES

Computer science

Telecommunications engineering

Huotari, Pasi Oracle based data system in global environment

Thesis 65 pages

Thesis supervisor Jarkko Kilponen

Commissioning Company Metso Minerals, Inc.

April 2008

Keywords Oracle, Database, Cluster, RAC, HP-UX

ABSTRACT

This thesis studied methods for reaching high availability and good response time in an information system. These are required in a modern global company environment. Furthermore, the objective of the work was to describe the general operation of the information system for internal use of the company. The target of the examination was Aton PDM system, which is used for product data management at Metso Minerals. The thesis also discusses architecture of Aton servers and interfaces to other systems. Oracle database is an essential part of the information system.

In the Oracle section of the study, architecture of Oracle and different backup methods of information were examined. Internal memory parameters of Oracle database management system were considered a significant factor on the response time. The system can be made more efficient by changing these parameters. It was also stated that indexing and analyzing the database had relevance to response times.

With high availability, the aim is to make the offline time of the system as small as possible. To carry this out, different cluster systems in which more servers are connected to one system to work together were studied. With clustering, the payload coming from client programs can be balanced to several servers. Alternatively, clustering can be used to bring another server into use when the primary server is damaged. There is also a possibility to use these two preceding methods simultaneously. The cluster systems reviewed were Oracle RAC, Windows Server 2003 and HP-UX, the last two of which are also operating systems. The cluster systems in question were not directly compatible with Aton PDM system. Consequently, carrying out the clustering reasonably would require software customizations from the Aton software supplier.

ALKUSANAT

Työ sai alkunsa yrityksen tarpeesta hankkia lisää tietoa Oracle-tietokannanhallintajärjestelmän toiminnasta. Tietojärjestelmän päivitysideonin yhteydessä huomattiin uusia mahdollisuuksia myös järjestelmän vikasietoisuuden parantamiseksi. Näiden kahden asian yhdistelmästä syntyi oma tutkintotyöni aihe. Varsinainen kirjoitustyö tehtiin kevään 2008 aikana, mutta luonnollisesti työskenneltyäni jo jonkin aikaa yrityksen *System Development* -tiimissä, olen huomaamattani tehnyt tätä tutkimustyötä jo useamman vuoden ajan.

Haluan kiittää Metso Minerals Oy:tä ja kehityspäällikkö Jarkko Kilposta mielenkiintoisen, haastavan sekä ajankohtaisen työn tarjoamisesta. Kiitos kuuluu myös työtovereilleni asiantuntemuksesta, auttamisesta sekä kannustuksesta työntekoon. Lisäksi haluan kiittää Modultek Oy:tä sekä Quest Software Oy:n tuotepäällikkö Jani Lindforsia kattavan asiantuntemuksen, koulutuksen sekä materiaalin tarjoamisesta, joka toimi oivana pohjana työlleni.

Tampereella 28.4.2008

Pasi Huotari

SISÄLLYSLUETTELO

Tiivistelmä

Abstract

Lyhenteet ja merkit

1	JOHDANTO.....	1
	1.1 Tutkintotyön määrittely	1
	1.2 Työn rakenne	2
	1.3 Metso Minerals Oy	2
2	TIETOKANNAT	4
	2.1 Relaatiomalli	4
	2.2 SQL-kieli	7
3	ATON PDM -JÄRJESTELMÄ	11
	3.1 Rooli yrityksessä	12
	3.2 Arkkitehtuuri.....	14
	3.2.1 Tiedon varastointi	14
	3.2.2 Looginen toiminta.....	16
	3.2.3 Käyttöliittymä	18
	3.3 Rajapinnat	18
4	ORACLE	20
	4.1 Tietokanta	20
	4.1.1 Fyysinen taso	21
	4.1.2 Looginen taso	24
	4.2 Instanssi	24
	4.2.1 Instanssin käynnistyminen.....	25
	4.2.2 Muistirakenne	26
	4.3 Suorituskyky	28
	4.3.1 Muistiparametrit	29
	4.3.2 Taulualueiden uudelleenorganisointi.....	30
	4.3.3 Indeksit ja tilastot.....	32
	4.4 Varmistukset ja palautus	34
	4.4.1 Fyysinen varmistus	34
	4.4.2 Looginen varmistus	35
	4.4.3 Palautus.....	35
5	KLUSTEROINTI	37
	5.1 Oracle RAC.....	40
	5.1.1 Klusteriohjelmisto	41
	5.1.2 Levyjärjestelmä	43
	5.1.3 Klusterinoodit	44
	5.1.4 Kuormantasaus	44
	5.2 Windows Server 2003.....	47
	5.2.1 Palvelinklusteri	47
	5.2.2 Kolmannen osapuolen sovellukset	49
	5.3 HP-UX	50
	5.3.1 Pääkomponentit	50
	5.3.2 Virtuaalisointi ja klusterointi	52
6	TULOKSET JA YHTEENVETO	57
	Lähteet	60

LYHENTEET JA MERKIT

CAD	<i>Computer Aided Design.</i> Tietokone-avusteinen suunnittelu.
CBO	<i>Cost Based Optimizer.</i> SQL-hakujen optimoija Oraclessa.
CORBA	<i>Common Object Request Broker Architecture.</i> Mahdollistaa ohjelmisto-komponenttien yhteenliittymisen riippumatta ohjelmointikielestä ja siitä, millä palvelimella ne sijaitsevat.
CSS	<i>Cascading Style Sheet.</i> Tyylitiedosto HTML-sivuja varten.
DBMS	<i>Database Management System.</i> Tietokannanhallintajärjestelmä.
DLL	<i>Dynamic Link Library.</i> Microsoftin kehittämä konseptista kirjastosta, missä dll-tiedostot koostuvat aliohjelmista itse ohjelmiston kehittämiseksi.
EAI	<i>Enterprise Application Integration.</i> Teknologia, jolla integroidaan yrityksen ohjelmistoja toisiinsa.
eParts	Sähköisten varaosakirjojen luontijärjestelmä.
EDMS	<i>Engineering Data Management System.</i> Suunnittelutiedonhallintajärjestelmä.
ERP	<i>Enterprise Resource Planning.</i> Toiminnanohjausjärjestelmä.
Ethernet	Lähiverkkotekniikka.
FTP	<i>File Transfer Protocol.</i> Tiedonsiirtomenetelmä kahden tietokoneen välillä.
GIM	<i>Global Item Master.</i> Aton-työkalu, jolla pystytään hallitsemaan lokaalit ja globaalit nimikkeistöt ja näiden väliset relaatiot.
HP	<i>Hewlett-Packard.</i> Tietokonevalmistaja.
HTML	<i>Hypertext Markup Language.</i> Kuvauskieli web-sivujen tekemiseen.
IIOP	<i>Internet Inter-ORB Protocol.</i> CORBA-tekniikan tietoliikenneprotokolla.

Java	Sun Microsystemsin kehittämä teknologiaperhe ja ohjelmistoalusta.
Java VM	<i>Java Virtual Machine</i> . Virtuaaliympäristö, jossa Java-ohjelmia suoritetaan.
JDBC	<i>Java Database Connectivity</i> . Rajapinta Java-ohjelmistojen ja erilaisten SQL-tietokantojen välillä.
JFC	<i>Java Foundation Classes</i> . Käytetään graafisten käyttöliittymien ohjelmoinnissa.
JSP	<i>JavaServer Page</i> . Servlet-rajapintaa hyödyntävä tekniikka, jolla voidaan upottaa Java-koodia web-sovelluksiin.
LAN	<i>Local Area Network</i> . Lähiverkko.
MIC	<i>Metso Inventory Control</i> . Varastohallintajärjestelmä.
MTS	<i>Multithreaded shared Servers</i> . Eräs Oraclen tapa hoitaa keskusmuistin allokointia käyttäjäsessioille. Tässä tavassa ei muistia anneta jokaiselle sessiolle erikseen.
PDM	<i>Product Data Management</i> . Tuotetiedon hallinta.
PGA	<i>Program Global Area</i> . Oraclen looginen muistialue tietokantapalvelimen keskusmuistissa. Hoitaa mm. käyttäjäsessioita.
RAID	<i>Redundant Array of Independent Disks</i> . Teknologia, joka käyttää useampaa fyysistä kovalevyä yhtä aikaa.
RMAN	<i>Recovery Manager</i> . Oraclen apuohjelma, jota käytetään varmistusten ottoon sekä tietokannan palautukseen.
SAN	<i>Storage Area Network</i> . Arkkitehtuuri, jolla yhdistetään ulkoisia kovalevyjä tietokoneisiin niin että käyttöjärjestelmä näkee sen paikallisena kovalevynä.
SCSI	<i>Small Computer System Interface</i> . Väylä, jolla välitetään tietoa tietokoneen ja oheislaitteen (esim. kiintolevyn) välillä.
SGA	<i>System Global Area</i> . Oraclen looginen muistialue, joka sijaitsee tietokantapalvelimen keskusmuistissa. Hoitaa mm. puskurointia.
SID	<i>Oracle System ID</i> . Tietokannan nimi.
Sivutus	<i>Paging</i> . Käytetään, kun halutaan allokoida tietokoneen kovalevyltä muistia keskusmuistiksi.

SQL	<i>Structured Query Language</i> . Standardoitu, tietokannoissa käytettävä hakukieli.
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i> . Käytetään Internet-liikennöinnissä.
TeamCenter	Suunnittelutiedonhallintajärjestelmä (EDMS).
UDP	<i>User Datagram Protocol</i> . Siirtokerroksen yhteydetön tietoliikenneprotokolla.
VIP	<i>Virtual Internet Protocol Address</i> . Virtuaalinen IP osoite.
VPN	<i>Virtual Private Network</i> . Mahdollistaa turvallisen liikennöinnin Internetin välityksellä esimerkiksi yrityksen verkkoon.

1 JOHDANTO

Yrityksen kasvaminen aiheuttaa usein tiukentuvia vaatimuksia myös tietojärjestelmien luotettavuudelle ja skaalautuvuudelle. Näin on tapahtunut myös Metso Mineralsilla. Kasvu on lisännyt luonnollisesti tietojärjestelmien käyttäjämääriä eikä järjestelmien käyttöaika rajoitu pelkästään Suomen toimisto aikaan. Jos liiketoiminnan kannalta tärkeä tietojärjestelmä ei ole saatavilla, kustannukset ovat yritykselle suuret.

Yrityksellä on paineita saada tietojärjestelmien teknisestä toiminnasta lisää tietoa. Tällä pyritään ehkäisemään riippuvuutta tietojärjestelmäosaajista, jotka ovat yleensä ns. kolmansia osapuolia. Näin ollen järjestelmien ongelmien korjauksiin saadaan parempi vasteaika, kun yrityksen sisällä työskentelee myös henkilöitä, joilla on syvällistä tietoa järjestelmän toiminnasta.

1.1 Tutkintotyön määrittely

Tämän työn tarkoituksena on tehdä tutkimus Metso Mineralsille Oracle-tietokannanhallintajärjestelmästä sekä sen klusteroinnista. Tutkimus liittyy Aton tuotetiedonhallinta -järjestelmään, joka käyttää tiedon tallentamiseen Oracle-tietokantaa. Tämän dokumentin on myös tarkoitus toimia perehdytysmateriaalina Aton PDM:ään sekä tietokantoihin, minkä vuoksi dokumentti on pyritty näiltä osin kirjoittamaan mahdollisimman selväkielisesti ja yleisellä tasolla.

Työ liittyy osittain projektiin, jossa HP ottaa valvontaansa yrityksen palvelimet. Tämä mahdollistaa paremman palvelinasiantuntemuksen. Tarve on myös Aton PDM:n version päivitykseen, joka mahdollistaa klusteriteknologian käytön tietokannan siirtyessä uuteen Oracle 10g -versioon. Työhön ei kuulu päivityksen aiheuttamien taloudellisten kustannusten arviointi. Työhön ei myöskään kuulu päivityksen testaus eikä implementointi. Klusterointijärjestelmien käsittelystä on jätetty pois virtuaalipalvelimet.

Työn tavoitteena on kertoa ne mahdollisuudet, joilla voidaan saada turhat käyttökatkokset pois sekä parantaa tietokannan suorituskykyä. Lisäksi tavoitteena on antaa selkeä kuvaus koko tietojärjestelmän arkkitehtuurista sekä Aton PDM:n että Oraclen puolelta yrityksen käyttöön.

1.2 Työn rakenne

Tässä työssä käsitellään Aton PDM -järjestelmää, Oracle-tietokannanhallintajärjestelmää ja -tietokantaa sekä klusterointia. Aton PDM -järjestelmän yhteydessä kerrotaan, mikä on järjestelmän rooli yrityksessä sekä miten arkkitehtuuri on teknisesti toteutettu. Oracle-tietokannanhallintajärjestelmän toimintaa tullaan käymään läpi luvussa 4. Tämä luku sisältää kuvauksen Oracle-tietokannasta ja järjestelmän eri elementeistä. Lisäksi selvitetään Oraclen suorituskykyyn vaikuttavia asioita. Luvussa 5 selvitetään järjestelmän klusterointia sekä sovellus- että laitteistotasolla ja käsitellään eri klusterointijärjestelmiä.

1.3 Metso Minerals Oy

Metso Minerals on maailmanlaajuinen johtaja kiven ja mineraalien käsittelyn sekä metallinkierrätyksen aloilla. Metso Mineralsin organisaatio koostuu kolmesta liiketoimintalinjasta.

- Maarakennus (*Construction*)
- Kaivosala (*Mining*)
- Kierrätys (*Recycling*)

Tärkeimpinä tuotteina voidaan mainita murskaimet, seulat, kuljettimet sekä liikkuvat murskaus- ja seulontalaitokset (kuva 1). /5/



Kuva 1 Liikkuva Lokotrack LT106 -murskauslaitos

Samoja tuotteita valmistetaan useassa eri yksikössä, mikä on osasyynä tietojärjestelmien globaaleihin vaatimuksiin. Metso Mineralsilla on tuotantoyksiköitä 36 maassa sekä myynti- ja huoltoyksiköitä 45 maassa. Henkilöstöä on yhteensä n. 10 000. /5/

Yritys kuuluu osana Metso-konserniin, jonka muina merkittävänä liiketoiminta-alueina voidaan mainita automaatiojärjestelmiä valmistava Metso Automation sekä paperikoneita valmistava Metso Paper. Metso työllistää yhteensä 26 800 henkilöä. /5/

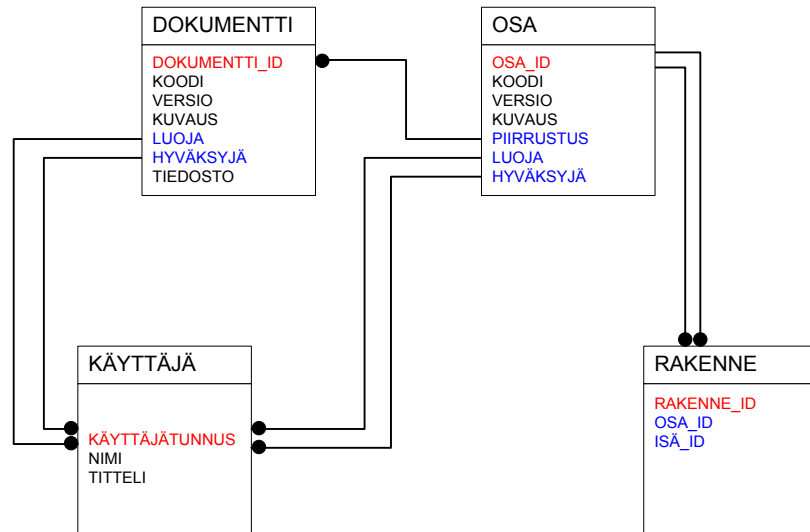
2 TIETOKANNAT

Tässä luvussa esitellään tietokantaa yleisellä tasolla sekä määritellään, mitä tällä termillä tässä materiaalissa tarkoitetaan. Koska dokumentin on tarkoitus myös toimia perehdytysmateriaalina, on lukijan hyvä tietää ensin perusasioita tietokannoista.

Sana tietokanta on hyvin monissa yhteyksissä käytetty termi. Tietokanta voi esimerkiksi koostua kokoelmasta lehtiartikkeleita tai kuvia. Tietokanta on kokoelma järjestettyä informaatiota (dataa), jota voidaan päivittää, lisätä ja tuhota. Tietokantaan liittyy olennaisena osana tietokannanhallintajärjestelmä. Näitä ovat esimerkiksi *Oracle*, *Microsoft SQL Server* sekä vapaaseen lähdekoodin perustuva *MySQL*, jota käytetään lähinnä web-sovelluksissa. Tietokannan ja käyttäjän välillä on usein jokin sovellus, jolloin itse tietokanta jää peruskäyttäjältä näkymättömiin. Tällaisena sovelluksena toimii myös Aton PDM. /1/

2.1 Relaatiomalli

Tässä luvussa käsitellään tietokantaa, joka koostuu tauluista ja niiden välisistä suhteista, eli se on tietokannan relaatiomalli. Muita malleja ovat esimerkiksi hierarkkiset ja oliotietokantamallit. Informaatio tallennetaan tauluihin, jotka koostuvat riveistä ja kentistä. Kentät voidaan myös ajatella sarakkeiksi, mikä helpottaa dataa sisältävän tietokantataulun ymmärtämistä, kun sitä verrataan normaaliin taulukoon. Yhdellä taulun rivillä tulee olla aina yksilöivä tekijä, joka erottaa rivin muista riveistä. Tätä tekijää kutsutaan nimellä pääavain (*Primary Key*). Tämä voi muodostua joko taulun yhdestä kentästä (sarakkeesta) tai useammasta kentästä. Taulujen väliset yhteydet hoidetaan viiteavaimilla (*Foreign Key*). Kuvassa 1 on esitetty esimerkki relaatiotietokannasta, jota yritys voisi käyttää tuotetiedonhallintaan. Esimerkki on kovin karkea, eivätkä tietokannat yleensä ole näin pieniä. /2/



Kuva 2 Esimerkki relaatiotietokannasta /2/

Kuvasta 2 nähdään, että käyttäjätunnus-kenttä on yksilöllinen tunniste Käyttäjätaulussa. Muut tämän taulun kentät ovat vain informatiivisia, mutta esimerkiksi titteli-kenttää voisi käyttää vaikka oikeuksien hallintaan. Tietyllä tittelillä työskentelevällä henkilöllä on järjestelmään tietynlaiset oikeudet.

Osa-taulussa hallitaan valmistettavien osien tiedot. Taulussa on osa_id pääavaimena. Pääavainta merkitään tietokantakuvausissa usein punaisella värillä. Lisäksi osa-taulussa on useita viiteavaimia, joita merkitään tavallisesti sinisellä värillä. Viittaus pääavaimen näkyy kuvassa relaatioviivoina taulujen välillä. Viivan päässä oleva pallo näyttää pääavaimen. Osa-taulussa on viiteavain esimerkiksi dokumentti-taulun dokumentti_id-kenttään. Taulukossa 1 on esitetty, mitä osa-taulu voisi sisältää.

Taulukko 1 Osa-taulun sisältö

OSA_ID	KOODI	VERSIO	KUVAUS	PIIRUSTUS	LUOJA	HYVÄKSYJÄ
0001	34587	0	ALUSLEVY	0001	SISA	PETO
0002	54576	0	ASENNUSLEVY	0002	MIMÄ	PETO
0003	35546	0	MUTTERI	0003	SISA	PETO
0004	21145	0	KOKOONPANO	0004	MIMÄ	PETO
0005	54576	1	ASENNUSLEVY	0005	MIMÄ	

Tässä kohtaa ainakin osa taulukon tiedoista vaikuttaa hyvin sekavilta. Tämä johtuu relaatiotietokannan luonteesta. Toiseen tauluun annetaan pieni viite tiedosta, mutta varsinainen selkokielineen informaatio on tallennettu toiseen tauluun. Esimerkiksi piirustus-kentässä on vain tieto relaatiosta dokumentti-taulussa olevaan dokumenttiin, mutta todellinen tieto löytyykin dokumentti-taulusta. Sama asia käy ilmi myös luoja ja hyväksyjän yhteydessä. Taulukoissa 2 ja 3 on esitetty dokumentti- sekä käyttäjä-taulujen sisällöt, joiden avulla myös osa-taulun tiedot on helppo selvittää.

Taulukko 2 Käyttäjä-taulun sisältö

KAYTTAJATUNNUS	NIMI	TITTELI
PETO	Toivonen, Pertti	Pääsuunnittelija
SISA	Saarela, Sirpa	Suunnittelija
MIMÄ	Mäki, Mikko	Suunnittelija

Taulukko 3 Dokumentti-taulun sisältö

DOKUMENTTI_ID	KOODI	VERSIO	KUVAUS	LUOJA	HYVÄKSYJÄ	TIEDOSTO
0001	34587	0	ALUSLEVY	SISA	PETO	34587_0.PDF
0002	54576	0	ASENNUSLEVY	MIMÄ	PETO	54576_0.PDF
0003	35546	0	MUTTERI	SISA	PETO	35546_0.PDF
0004	21145	0	KOKOONPANO	MIMÄ	PETO	21145_0.PDF
0005	54576	1	ASENNUSLEVY	MIMÄ		54576_1.PDF

Taulukkoja 2 ja 3 tutkiessa käy ehkä paremmin ilmi viiteavaimien merkitys. Jos tarkastelemme osa-taulun sisältöä luoja ja hyväksyjän kohdilta ja vertaamme sitä käyttäjä-tauluun, huomataan, että kaikki hyväksytyt osat on hyväksynyt pääsuunnittelija Toivonen ja että osat on tietokantaan luonut kaksi suunnittelijaa. Dokumentti-taulussa löytyy kaikki piirustukset, joita osa-taulun piirustus-kentässä on esitetty. Dokumentti-taulussa olevaa tiedosto-kenttää voi tietokannan ja käyttäjän

välillä toimiva käyttöliittymä hyödyntää, kun halutaan hakea piirustusta esimerkiksi tiedostopalvelimelta.

2.2 SQL-kieli

SQL (*Structured Query Language*) on suhteellisen yksinkertainen standardoitu tietokonekieli, jota käytetään tietokannanhallinnassa. Tällä kielellä voi esimerkiksi poistaa, lisätä ja muokata tietokannan sisältämiä tietoja.

Tärkeimmät komennot SQL-kielessä ovat SELECT, jolla valitaan tietoja kannasta, INSERT, jolla lisätään tietoja kantaan, DELETE, jolla tuhoetaan tietoja kannasta sekä UPDATE, jolla muokataan olemassa olevia tietoja. Näitä kolmea komentoa käsitellään seuraavaksi esimerkein. Ohessa tulee myös muita SQL:n perusasioita. Esimerkeissä käytetään kuvan 2 mukaista tietokantaa.

Seuraavaksi on listattu, mitä SELECT-lauseeseen tarvitaan.

- SELECT, mitä kenttiä haetaan
- FROM, mistä tauluista haetaan
- WHERE, millä ehdoilla
- ORDER BY, miten lajitellaan

Näistä ainoastaan SELECT ja FROM ovat pakollisia, mutta jos WHERE-osiossa ei anneta ehtoja haun rajaamiseksi, tulee haku palauttaa kaikki rivit taulusta.

Esimerkkinä on SELECT-lause kuvan 2 mukaisesta tietokannasta.

```
SELECT KOODI, VER, KUVAUS  
FROM OSA  
WHERE KOODI = '54576'  
ORDER BY KUVAUS;
```

Yllä oleva haku palauttaisi osa-työkalusta kaikki rivit, joiden koodi-kentän arvona on 54576. Lajittelu suoritetaan kuvaus-kentän mukaan laskevassa aakkosjärjestyksessä. Jos ORDER BY -rivin lopussa oli sana DESC, niin silloin hakutulokset järjestettäisiin nousevassa aakkosjärjestyksessä. Tässä kohtaa lajittelu on hieman turhaa, koska kuvaus eri versioilla on hyvin usein sama. Muutenkin lajittelun voi ennemmin suorittaa taulukkolaskenta-ohjelmassa, esimerkiksi Excelissä. Huomioitavaa on, että SQL-kieli ei ota kantaa siihen, onko lause kirjoitettu isoilla vai pienillä kirjaimilla. On kuitenkin muistettava, että esimerkiksi Oracle-tietokannanhallintajärjestelmä käsittelee pienet ja isot kirjaimet erilaisina. Jos haku on tehty ensin isoilla kirjaimilla kirjoitettuna ja sitten pienillä, niin Oracle ei pysty palauttamaan edellistä hakua välimuistista, koska se ei tunnista hakulauseita samaksi.

Alla on esimerkki SELECT-lauseesta, jossa haettavia tauluja on useampia. Tässä käytetään jälleen kuvan 2 tietokantaa esimerkkinä.

```
SELECT A.KOODI, A.VER, B.TIEDOSTO
FROM OSA A, DOKUMENTTI B
WHERE A.PIIRUSTUS = B.DOKUMENTTI_ID AND
A.KOODI = '54576'
```

Yllä oleva haku listaisi kaikki osan 54576 versiot ja niihin liittyvien dokumenttien piirustustiedostot. Kun haetaan useasta eri taulusta tietoa, täytyy käyttää taulutunnisteita (*alias*). Osa-työkalulle annetaan tunniste A, dokumentti-työkalulle annetaan tunniste B. Tunnisteita tulee käyttää, koska taulut sisältävät samannimisiä kenttiä. Esimerkiksi koodi- ja versio-kentät esiintyvät sekä osa-työkalussa että dokumentti-työkalussa. Tunnisteella voidaan erotella, mistä taulusta on kyse. Tunnisteena WHERE-osiossa voitaisiin käyttää myös taulun nimeä, jolloin WHERE-osioon ei tarvitsisi laittaa tunnistetta, mutta kokemus on osoittanut tämän vievän aikaa. Lisäksi tällainen menettely ei ole edes mahdollista, jos sama taulu esiintyy FROM-osiossa useaan kertaan.

Alla on listattu, mitä INSERT-lauseessa tarvitaan

- INSERT INTO, mihin tauluun lisätään ja mihin kenttiin
- VALUES, mitä arvoja

Jos haluttaisiin lisätä uusi käyttäjä tuotetiedonhallinnan tietokantaan, niin voitaisiin käyttää seuraavaa lausetta:

```
INSERT INTO KAYTTAJA (KAYTTAJATUNNUS, NIMI, TITTELI)
VALUES ('VIVI', 'Virtanen Ville', 'Kehitysinsinööri');
```

Käyttäjätunnukseksi tulisi siis VIVI, Käyttäjän nimeksi Ville Virtanen sekä titteliksi kehitysinsinööri.

Seuraavaksi on listattu, mitä DELETE-lauseeseen tarvitaan.

- DELETE FROM, mistä taulusta tuhotaan
- WHERE, millä ehdoilla rivejä tuhotaan

Kaikki dokumentit, jotka on luonut Sirpa Saarinen, tuhottaisiin seuraavasti:

```
DELETE FROM DOKUMENTTI
WHERE LUOJA = 'SISA'
```

Käyttäjätunnusta voi olla hankalaa muistaa. Näin ollen voidaankin alihakuja hyväksi käyttäen hakea käyttäjätunnus seuraavasti:

```
DELETE FROM DOKUMENTTI
WHERE LUOJA IN
(
    SELECT KÄYTTÄJÄTUNNUS
    FROM KÄYTTÄJÄ
    WHERE NIMI = 'Saarinen, Sirpa'
);
```

Alihaut ovat puhtaita SELECT-lauseita. Nämä pitää vain erottaa suluilta päähausta. Lisäksi pitää käyttää IN-operaattoria ennen alihakua, jos tuloksia odotetaan useam-

pia. Jos alihaun tiedetään palauttavan vain yhden rivin, niin voitaisiin IN-operaattorin tilalla käyttää yhtäsuuruusmerkkiä (=). Siis yllä olevassa lauseessakin voitaisiin IN-sanana sijasta käyttää yhtäsuuruusmerkkiä.

UPDATE-lauseen syntaksi on esitetty seuraavaksi

- UPDATE, mitä taulua päivitetään
- SET, mitä kenttiä päivitetään ja mihin arvoon
- WHERE, millä ehdoilla

Esimerkiksi jos käyttäjän sukunimi vaihtuu, täytyy tietojärjestelmään päivittää uusi nimi. Tämä toteutettaisiin seuraavalla tavalla:

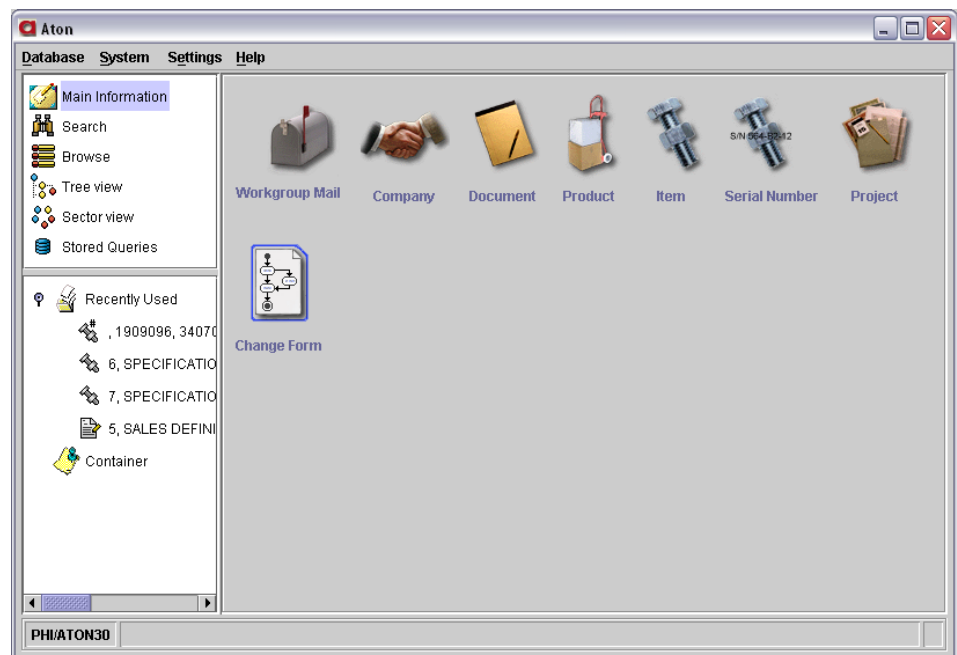
```
UPDATE KAYTTAJA  
SET NIMI = 'Toivonen, Sirpa'  
WHERE NIMI = 'SISA';
```

3 ATON PDM -JÄRJESTELMÄ

Aton PDM -tuotetiedonhallintajärjestelmä on porilaisen ohjelmistoyhtiö Modultek Oy:n kehittämä sovellus, joka mahdollistaa tuotetietojen hallittavuuden. Aton-sovelluksella voidaan hallita seuraavia asioita:

- Nimikkeet
- Dokumentit
- Yksilölliset toimitukset
- Tuotemuutokset
- Projektit
- Tuotteet
- Nimikkeistä koostuvat tuoterakenteet
- Yhtiöt

Sovellus koostuu erilaisista näytöistä, kuten esimerkiksi nimike-, dokumentti- sekä yksilönäytöistä. Saatavissa on Java-pohjainen käyttöliittymä sekä selaimessa toimiva portal-käyttöliittymä. Kuvassa 3 on esitetty Aton PDM:n päänäyttö (Java). /3/



Kuva 3 Atonin päänäyttö

Lisäksi Atonissa on olemassa käyttäjäoikeushallinta sekä rajapintoja muihin järjestelmiin, kuten esimerkiksi CAD- tai ERP-järjestelmiin. /3/

3.1 Rooli yrityksessä

Aton PDM -järjestelmää käytetään Metso Mineralsin Construction-liiketoimintalinjan nimikkeiden, nimikerakenteiden, dokumenttien, sarjanumeroiden, muutospyyntöjen ja -tiedotusten hallintaan. Aton PDM -käyttäjäkuntaan kuuluu mm. suunnittelu-, tuotanto-, jälkimarkkina- ja alihankintayksiköitä. Se on siis globaali järjestelmä, jota käytetään kaikissa maarakennus-liiketoimintalinjan suunnittelu- ja tuotantoyksiköissä. Aton-palvelimia on yhteensä 12:ssa eri tuotanto- ja suunnittelu-yksikössä. Kuvassa 4 nämä yksiköt on merkattu keltaisella.

Järjestelmään sisältyy myös GIM (*Global Item Master*) -työkalu, jolla hallitaan risiiviittaukset globaalien ja lokaalien nimikkeiden (komponenttien) välillä. Kuvassa 4 GIM:iä käyttävät yksiköt on merkattu violetilla värillä. GIM mahdollistaa nimikkeiden harmonisoinnin sillä tavalla, että eri yksiköt voivat edelleen käyttää omia koodiaan myymissään tuotteissa. Nämä koodit liitetään Aton PDM:ssä yhteen globaaliin nimikkeeseen.



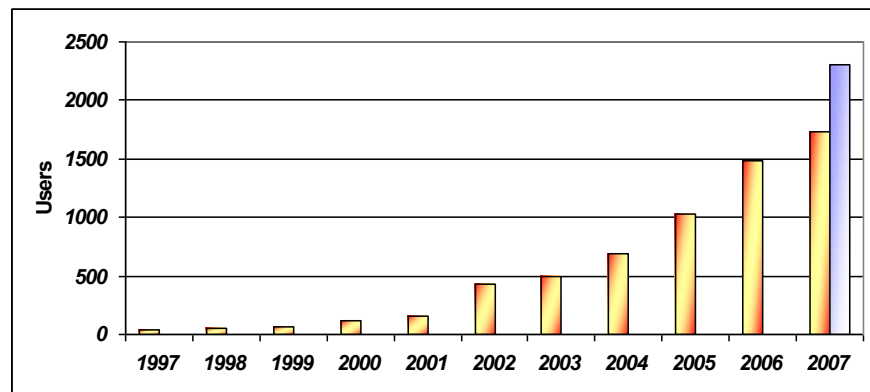
Kuva 4 Aton PDM:ää käyttävät yksiköt /4/

Kuvassa 5 on esitetty Atonin datasisältöä. Kaikki tiedot ovat peräisin helmikuun 2008 lopusta. Kuvasta 5 nähdään, että suurin datamäärä on yksilön rakennetaulus- sa. Myös nimikkeitä kokonaisuudessaan on hyvin paljon, mutta näistä vain osa on käytössä. Tämä johtuu siitä, että yrityksen laajenemisen myötä on järjestelmään tuotu useista eri ERP-järjestelmistä nimiketietoutta, mutta tätä nimikkeistöä ei ole vielä päivitetty nimikkeeltä vaadittavan tiedon laadun mukaiseksi. Myös muutosob- jekteja on muita vähemmän, koska ensimmäinen muutosobjekti otettiin käyttöön vasta vuoden 2006 keväällä Tampereella.

	Nimikkeitä yhteensä Käytössä olevia nimikkeitä Rakennereivejä	1 337 000 294 000 1 924 000
	Kaikki dokumentit Tiedoston sisältävät dokumentit PDF (Piirustus) DXF & DWG JT (Direct Model, 3D View) Others (Word, Excel...)	721 000 511 000 58 % 18 % 13 % 11 %
	Muutosobjekteja yhteensä Muutostiedostuksia Muutospyyntöjä	2 150 1 385 725
	Komponentteja	1 415 000
	Toimitettuja yksilöllisiä koneita Yksilörakennereivejä	17 000 29 772 000

Kuva 5 Aton PDM:n datasisältö Metso Mineralsilla

Kuvassa 6 on esitetty Atonin huima käyttäjämäärän kasvu viimeisen kymmenen vuoden ajalta. Viimeinen palkki sinisellä on käyttäjätunnusten kokonaismäärä jär- jestelmässä. Muut palkit ovat käyttäjämääriä aktiivisten käyttäjien suhteen. Tämä yhdessä datasisällön kanssa tekee järjestelmästä todella kriittisen, jolloin on tarvet- ta turvata järjestelmän saatavuus, toiminnallisuus sekä hyvä suorituskyky isollakin käyttäjämäärällä vuorokauden ympäri.



Kuva 6 Käyttäjämäärän kehitys vuosina 1997 - 2007 /4/

3.2 Arkkitehtuuri

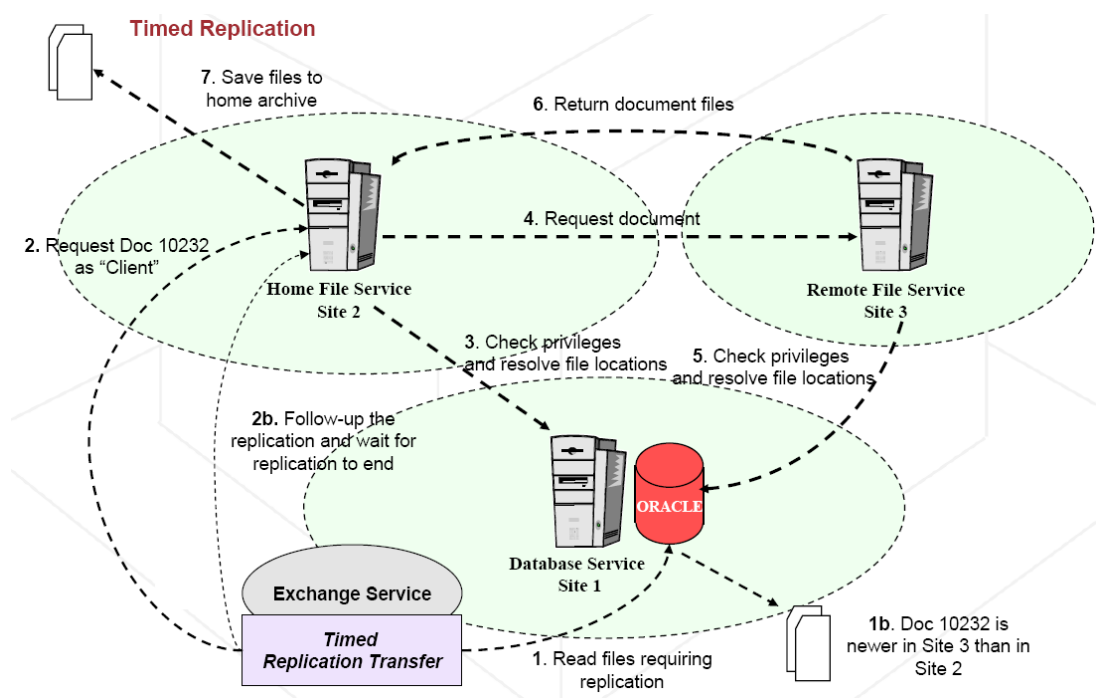
Aton-järjestelmän arkkitehtuuri lähtee globaalista näkökulmasta. Yrityksellä voi olla useita yksiköitä eri paikoissa, ja kuitenkin vaaditaan yhtenäinen paikka kaiken tuotetiedon säilyttämiseksi. Tässä kohtaa järjestelmäarkkitehtuuria esitellään kolmella eri tasolla. Alimpana tasona on tiedon varastointi, joka koostuu tietokannasta sekä tiedostovarastosta dokumenteille. Toisena tasona on sovellustaso, joka koostuu ohjelman loogisesta toiminnallisuudesta. Tähän kuuluvat mm. Java- ja CORBA-teknologiat. Viimeisenä tasona on rajapinta käyttäjään eli käyttöliittymä. /6, s. 1 - 4/

3.2.1 Tiedon varastointi

Tiedon varastointitaso sisältää itse tietokannan sekä yhden tai useamman tiedostotietokannan. Vaikka Aton-järjestelmä lähtee globaalista ajattelumallista, on tietokantoja vain yksi, mikä helpottaa hallittavuutta sekä varmistusten ottoa, mutta myös aiheuttaa hieman ongelmia järjestelmän vasteaikoihin pitkien välimatkojen päästä. Metso Mineralsilla käytössä oleva Aton tukee vain Oracle 9i - tietokannanhallintajärjestelmää, mutta tämänkin työn yhtenä päätarkoituksena on tutustua uudempaan Oracle 10g järjestelmään. Modutek julkaisi maaliskuussa 2008

uuden Aton 5 -version, joka käyttää Oracle 10g -tietokantaa. Myös järjestelmän oikeushallinta suoritetaan tietokantatasolla. /6, s. 1/

Tiedostoarkistoja käytetään dokumenttiedostojen säilyttämiseen. Niitä voi olla yksi tai useampia. Aton mahdollistaa myös tiedostojen replikoinnin eri tiedostoarkistojen välillä, jolloin tiedostot pysyvät mahdollisimman lähellä käyttäjää. Kuvassa 7 on esitetty ajastetun replikoinnin kulkukaavio. /6, s.1/



Kuva 7 Ajastettu tiedstoreplikointi /7/

Kuvassa 7 ajastettu replikointi suoritetaan esimerkiksi kerran yössä. Aton PDM:n Exchange palvelu ohjaa replikoinnin suorittamista. Exchange-palvelusta on kerrottu lisää luvussa 3.2.2. Replikointi-transaktio pyytää tiedostoja toiselta yksiköltä, joille replikointia on pyydetty, ja kun tiedostot saapuvat, se tallentaa ne tiedostoarkistoon.

3.2.2 Looginen toiminta

Aton-ohjelmiston toiminta nojautuu hyvin pitkälti Javaan sekä CORBA-teknoologiaan. CORBA-teknoologia mahdollistaa jaetun sovelluspalvelinarkkitehtuurin (kuva 8) käyttäen IIOP-protokollaa TCP/IP-protokollan päällä. Java-teknoologia puolestaan mahdollistaa erilaisten palvelinratkaisujen käytön. Palvelinten ei siis kaikkien tarvitse välttämättä olla Windows-palvelimia. /6, s. 2, 5/

Aton PDM:n sovelluspalvelinarkkitehtuuri koostuu erilaisista palvelumoduuleista. Näitä ajetaan sovelluspalvelimella Java-virtuaalikoneissa (*Java VM*). Yksi virtuaalikone voi käsittää yhden tai useamman palvelumoduulin. Seuraavaksi esitellään palvelumoduulit, jotka ovat Metso Mineralsilla käytössä.

Tietokantapalvelu (*Database Service*) kontrolloi käyttäjäsessioita Oracle-kantaan sekä hoitaa kaikkien muiden palvelumoduulien yhteyden tietokantaan. Pohjana on käytännössä yksi JDBC-rajapinta.

Sovelluspalvelu (*Application Service*) hoitaa käyttäjän paikannuksen sekä Java-palvelut käyttäjälle globaalissa ympäristössä.

Tiedostopalvelu (*File service*) hoitaa kaikki tiedostoihin liittyvät asiat kuten uuden tiedoston tuomisen tai viemisen järjestelmästä, replikoinnin yhdessä Exchange-palvelun kanssa sekä tiedostojen pakkaamisen.

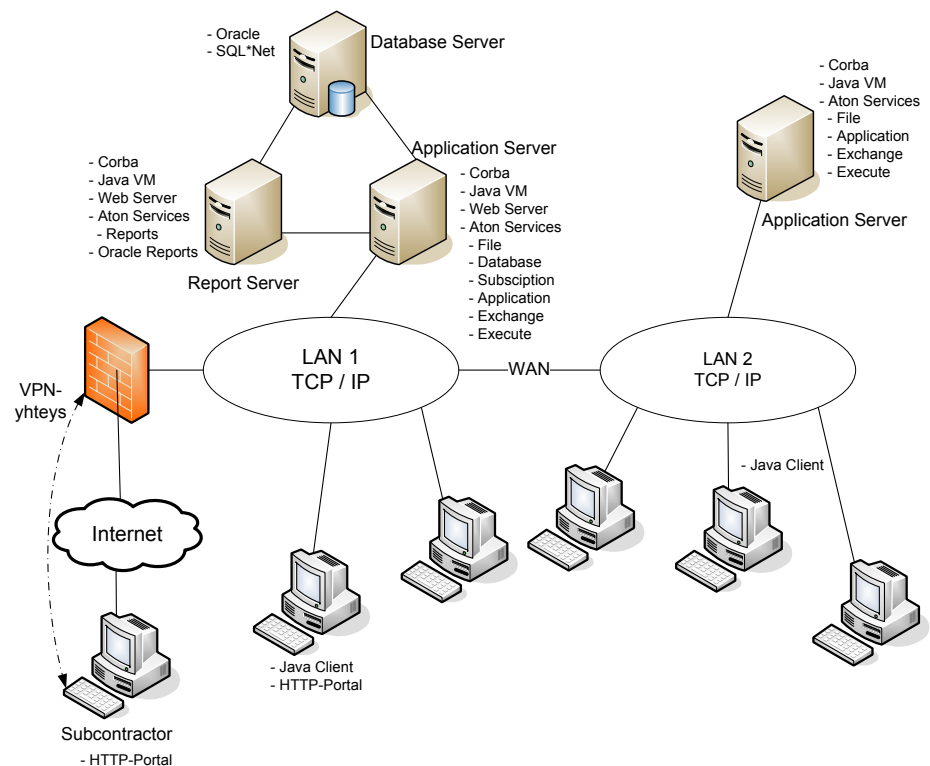
Raporttipalvelun (*Report Service*) avulla saadaan järjestelmästä erilaisia listauksia kuten esimerkiksi osaluetteloita. Pohjana tälle on *Oracle Reports* -ohjelma

Ilmoituspalvelu (*Notification Service*) hoitaa ilmoituksen tietyille käyttäjille sähköpostitse, kun järjestelmässä tehdään tietty toiminto, esimerkiksi luodaan tietynlainen dokumentti.

Exchange-palvelu (*Exchange Service*) hoitaa tiedonsiirtoa muiden tietojärjestelmien ja Aton PDM:n välillä.

Execute-palvelu (*Execute Service*) toimii Exchange-palvelun kanssa yhdessä. Tällä palvelulla voidaan suorittaa käyttöjärjestelmäkomentoja, kuten tiedostojen siirtoja paikasta toiseen.

Kuvassa 8 on esitetty Metso Mineralsilla oleva palvelinratkaisu kahden yksikön näkökulmasta. LAN 1 on ns. pääyksikkö (*Mainsite*). Se on toteutettu kolmella eri palvelimella resurssien tasaamiseksi. Lisäksi alihankkijat, jotka eivät ole Metson verkossa, voivat käyttää järjestelmää VPN:n avulla. LAN 2:ssa sijaitseva toinen sovelluspalvelin ottaa suoraan yhteyden pääyksikössä sijaitsevan sovelluspalvelimen tietokantapalveluun, jonka kautta se pääsee varsinaiselle tietokantapalvelimelle. Näin ollen yhdelle sovelluspalvelimelle ei tule liikaa kuormaa vaan näitä pystytään hajauttamaan. Myös tiedostopalvelun pystyttäminen toiseen yksikköön on järkevää, sillä näin yksikössä työskenteleville käyttäjille tiedostot avautuvat nopeammin, koska ne liikkuvat vain yhden nopean paikallisverkon yli käyttäjälle.



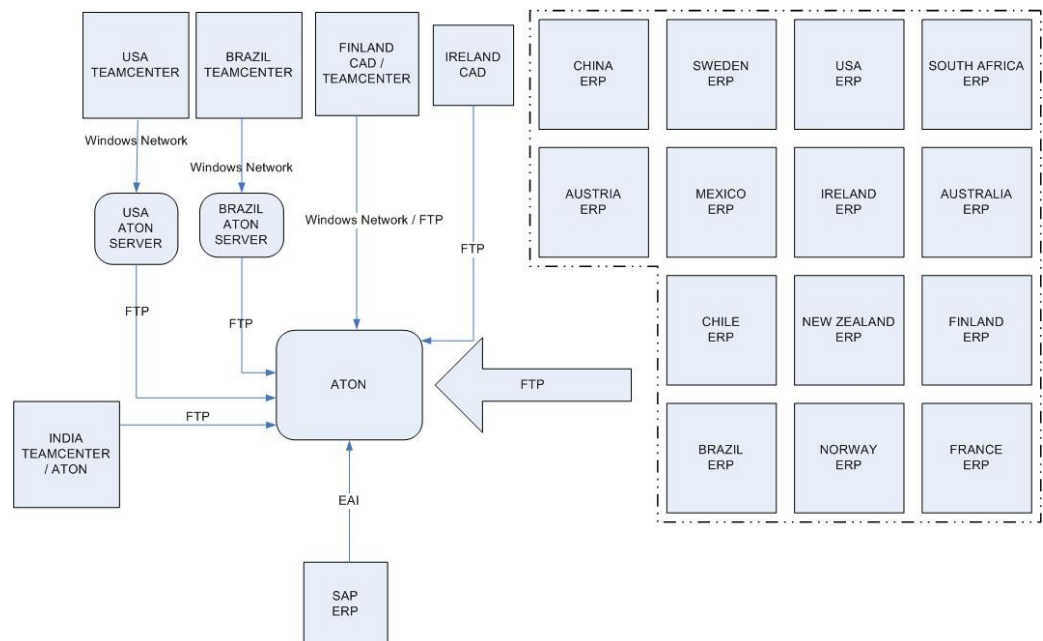
Kuva 8 Aton palvelinarkkitehtuuri /6/

3.2.3 Käyttöliittymä

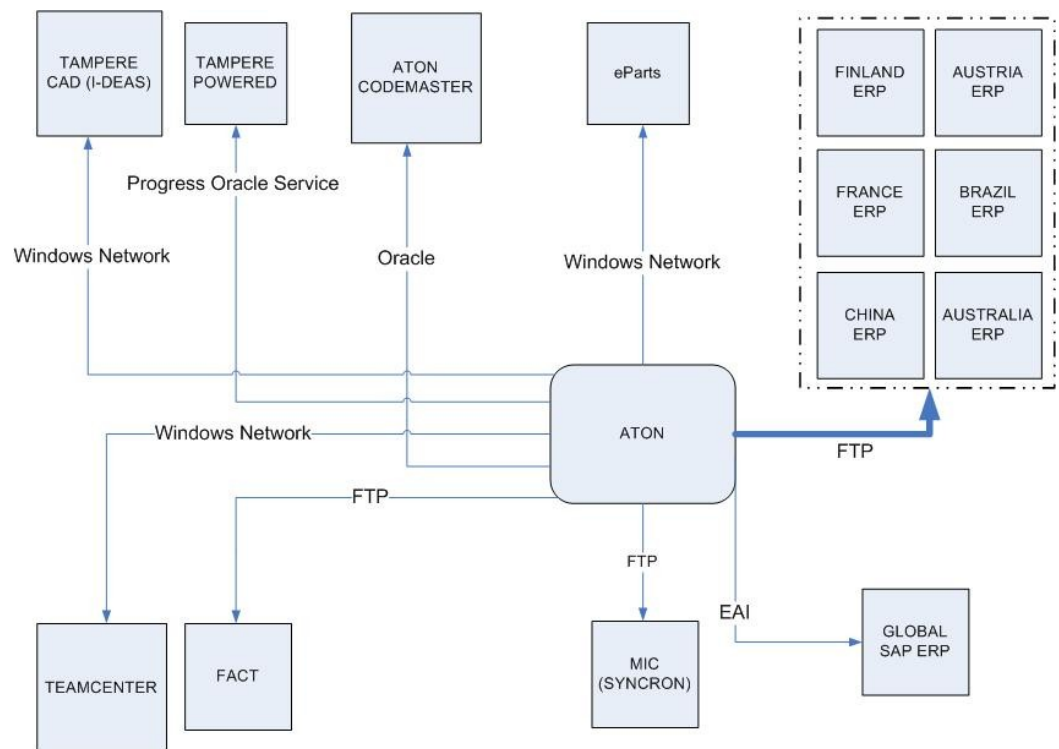
Käyttöliittymiä Aton PDM:ssä on kahdenlaisia. Java-pohjainen asiakasohjelmisto sekä Internet-selaimessa käytettävä Aton Portal. Molemmissa on kuitenkin Java-teknologia pohjana. Asiakasohjelmisto käyttää JFC-komponentteja, kun taas web-käyttöliittymä hyödyntää JSP-teknologiaa. Molempia käyttöliittymiä voidaan muokata suhteellisen helposti. Asiakasohjelmiston käyttöliittymäkonfiguraatiot on tallennettu Atonin rekisteriin, joka on käytännössä yksi tietokantataulu. Web-käyttöliittymää on mahdollista muokata editoimalla sekä HTML- että CSS-tiedostoja. /6, s. 3/

3.3 Rajapinnat

Järjestelmän tehtävän takia Aton PDM on integroitu useaan eri tietojärjestelmään. Näitä ovat esimerkiksi EDMS, CAD- sekä ERP-järjestelmät. Kuvassa 9 on esitetty Aton PDM:n rajapinnat muihin järjestelmiin, kun tietovirta on muista järjestelmistä Atoniin päin. Kuvassa 10 on esitetty rajapinnat, kun tietovirta on Atonista ulospäin.



Kuva 9 Atoniin sisäänpäin menevä tietovirta



Kuva 10 Atonista ulospäin menevä tietovirta

FTP-protokolla on selkeästi hallitsevin menetelmä integraatioiden toteutuksiin. Windows-verkoilla tässä yhteydessä tarkoitetaan toisilla palvelimilla sijaitsevia kansiojakoja, jotka on yhdistetty Aton-palvelimeen verkkolevyksi. Tiedonsiirto tapahtuu näiden jakojen kautta. Globaalin SAP-järjestelmän myötä myös EAI-tiedonsiirtoteknologia on tullut mukaan rajapintoihin. EAI mahdollistaa myös tiedonsiirron monitoroinnin, mihin FTP ja kansiojakojen kautta tapahtuvat siirrot eivät suoranaisesti kykene. EAI:n monitorointi tapahtuu MQ-palvelun kautta.

Kuvissa on katkoviivalla rajattu eri ERP-järjestelmät, joista tulee samantyyppistä tietoa. Kuvassa 10 Powered-tuotannonohjausjärjestelmä sekä Aton CodeMaster -järjestelmä kommunikoivat suoraan Atonin tietokannan kanssa Oraclen omien palveluiden kautta. Tästä syystä siirtomenetelmäksi on merkitty viittaus Oracleen. Muuten integraatiot on toteutettu siten, että Aton generoi tekstipohjaisen tiedoston, joka lähetetään jotakin edellä esiteltyä tapaa käyttäen toiselle järjestelmälle, joka lukee tiedon sisään. Atoniin virtaavassa tiedossa prosessi menee luonnollisesti päinvastaisesti. Ulkopuolinen järjestelmä generoi tiedoston ja Aton lukee ja kirjoittaa tiedon omaan tietokantaansa. Vaihtoehtoisesti Aton voi kirjoittaa siirtotiedoston lisää tietoa ja lähettää eteenpäin, mutta ei kirjoita tietoa omaan tietokantaan.

4 ORACLE

Oracle voidaan käsittää kahtena eri asiana. Se on relaatiotietokanta, mutta samalla myös tietokannanhallintajärjestelmä. Tietokannanhallintajärjestelmän (*Database Management System, DBMS*) on pystyttävä hallitsemaan suurta määrää informaatiota usean käyttäjän järjestelmässä. Käyttäjien on vielä päästävä samanaikaisesti käsiksi samaan dataan. Lisäksi järjestelmän vasteaika informaation hakemiseen, lisäämiseen, muokkaamiseen ja poistamiseen ei saa olla liian pitkä. DBMS-järjestelmän tulee olla tietoturvallinen siinä mielessä, että jokainen kantaan kirjautuva käyttäjä todennetaan esimerkiksi käyttäjätunnuksen ja salasanan avulla. Näin estetään järjestelmän luvaton käyttö. /8/

Oracle on myös uusimpien versioiden myötä alkanut kiinnittää enemmän huomiota isojen yritysten tarpeisiin. Oracle tarjoaa ns. grid-teknologiaa, joka mahdollistaa vikasietoisen ympäristön. Grid-teknologiassa periaate on se, että useampia palvelimia liitetään yhteen tietokantaan, jolloin voidaan suorittaa kuormantasausta tai vaihtoehtoisesti yhden palvelimen hajotessa voidaan toinen ottaa sulavasti käyttöön. Tämä tarjoaa tietojärjestelmään mm. joustavuutta ja vikasietoisuutta. Periaatteessa grid on sama kuin luvussa 5 käsiteltävä klusteri. /9/

Tässä luvussa käydään läpi itse tietokanta kahdella eri tasolla. Tietokantaosiossa käsitellään mm. erilaisia tiedostoja, jotka ovat tärkeitä tietokannan toimivuudelle. Tämän jälkeen käsitellään varsinaista Oracle-prosessia, mikä tuo tietokannan saataville, ja sitä, miten tietokannan suorituskykyä voidaan parantaa. Lopuksi käsitellään varmistusrutiineja sekä tietokannan palautusta.

4.1 Tietokanta

Oracle-tietokanta koostuu fyysisistä ja loogisista tasoista. Fyysiset rakenteet ovat nähtävissä käyttöjärjestelmässä tiedostoina, ja niitä voidaan myös hallita tätä kautta. Loogiset rakenteet näkyvät vain Oraclen omilla hallintaohjelmistoilla ja käyttöjärjestelmä ei ole näistä tietoinen.

4.1.1 Fyysinen taso

Tietokannan fyysinen taso koostuu tiedostoista Oracle-tietokantapalvelimella. Tiedostoja ovat mm. tietokanta-tiedostot (*database file, dbf*), joihin varsinainen tietokannan sisältämä informaatio on tallennettu ja ohjaustiedostoja (*control file, ctl*). Tässä alaluvussa esitellään näiden tiedostojen toimintaa ja sisältöä.

Ohjaustiedosto

Ohjaustiedosto on erittäin kriittinen komponentti tietokannan toiminnan kannalta, koska se sisältää tietoa tietokannan fyysisestä rakenteesta. Tiedosto kuuluu mm. seuraavia asioita /11/:

- Arkistolokihistoria
- Informaatiota taulualueista sekä tietokantatiedostoista
- Tieto käytössä olevasta redo-lokista
- Tietokannan nimi

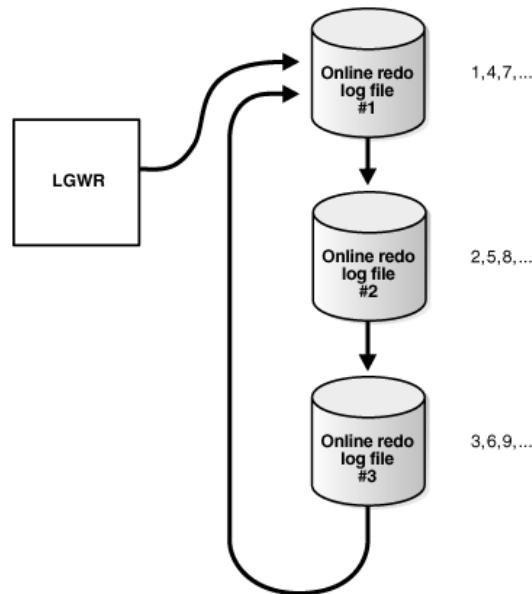
Ohjaustiedoston kriittisyyden vuoksi tulee siitä olla aina useampi kopio. Esimerkiksi voitaisiin pitää kolmea identtistä ohjaustiedostoa, joista vain yhtä kanta käytäisi ja loput kaksi olisivat varalla. Ohjaustiedostojen sijainti määritetään init-tiedostossa, josta edempänä lisää. Kannan varmistuksen yhteydessä on myös hyvä ottaa ohjaustiedostosta kopio trace-tiedostoon käskyllä "ALTER DATABASE BACKUP CONTROLFILE TO TRACE;".

Online redo -lokit

Oracle kirjoittaa muutostietoa *online redo* -tiedostoihin, kun kannan sisältö muuttuu. Tiedostoja on yleensä kaksi tai useampia. Kanta kirjoittaa muutostietoa vuorotellen eri tiedostoihin. Lokitietoja voidaan käyttää hyväksi, jos kanta joudutaan palauttamaan. /10/

Oracle-kannassa on useita taustaprosesseja. Yksi näistä taustaprosesseista on lokin kirjoittaja (*log writer, LGWR*). Tämä on se prosessi, joka kirjoittaa muutostietoja redo-lokeihin *redo log* -puskurista. Lokin kirjoitusprosessi on esitetty kuvassa 11. Kun lokin kirjoittaja saa ensimmäisen lokin kirjoitettua, se siirtyy toiseen lokiin. Tämän jälkeen on vuorossa kolmas loki, jos sellainen on olemassa. Kun kaikki lo-

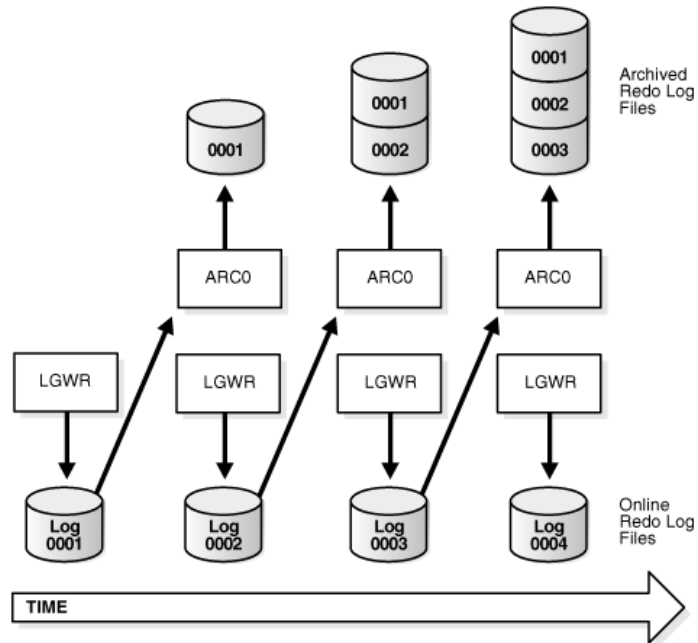
kit on kirjoitettu, niin kierto alkaa alusta. Täynnä oleva loki täytyy olla arkistoitu ja kirjoitettu kantaan (tietokantatiedostoihin) ennen kuin siihen päästään kirjoittamaan uudestaan, jos arkistointimoodi (*archive log mode*) on kannassa käytössä. Jos tätä ei ole käytössä, pelkkä kantaan kirjoittaminen riittää siihen, että lokin kirjoittaja pääsee kirjoittamaan samaan lokiin uudestaan. /12/



Kuva 11 Lokinkirjoittajan toiminta /10/

Archive log -tiedostot

Oraclessa on mahdollista arkistoida redo-lokeja sen sijaan, että lokin kirjoittaja ylikirjoittaisi muutostiedot seuraavalla kiertokerralla. Tässä kohtaa sanotaan kannan olevan arkistointimoodissa (*archive log mode*). Oraclen taustaprosessi ARC0 (numero voi olla muukin kuin nolla) suorittaa redo-lokien arkistointia. Näin ollen kanta voidaan korruptiotilanteessa palauttaa kauemmas ajassa. Kuvassa 12 on esitetty redo-lokien ja arkistoitujen lokien yhteistoiminta. Lokin kirjoittaja siis hoitaa kirjoittamisen, ja ARC0-prosessi arkistoi lokin, kun kirjoittaja on saanut lokin valmiiksi. /12/



Kuva 12 Arkistointiprosessi /12/

Tärkeintä arkistointimoodissa on kuitenkin mahdollisuus ottaa kannasta varmistus, kun järjestelmä on normaalissa käytössä. Tämä on erittäin tärkeää myös tässä työssä käsiteltävässä järjestelmässä, koska se on käytössä periaatteessa 24 tuntia vuorokaudessa ja 7 päivää viikossa.

Tietokantatiedostot

Tietokantatiedostot sisältävät varsinaisen kannan informaation. Ne ovat fyysisiä tiedostoja käyttöjärjestelmässä, jotka voidaan nähdä, mutta niiden sisältöä ei varsinaisesti pääse näkemään. Tietokantatiedostot voidaan jakaa kolmeen osaan.

/10/

Segmentti (*segment*) sisältää tietynlaisia tietokantaobjekteja. Esimerkiksi tietokantataulut kuuluvat taulusegmenttiin ja indeksit indeksisegmenttiin. *Extentit* koostuvat useasta datalohkosta (*data block*). Datalohko taas on pienin yksikkö tietokantatiedostossa. /10/

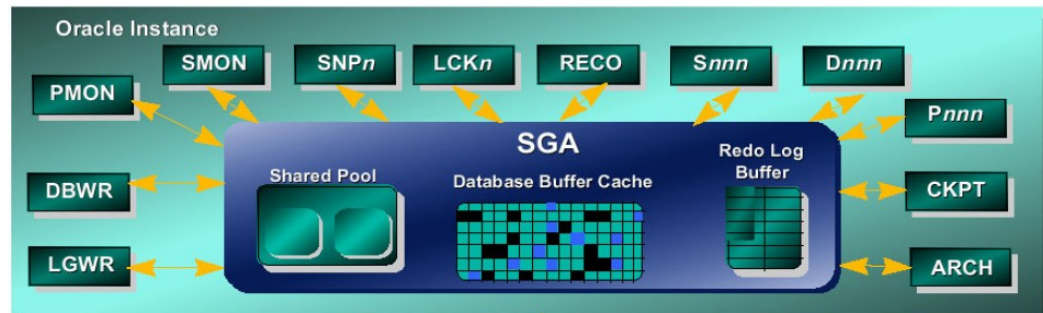
4.1.2 Looginen taso

Oraclen looginen taso koostuu taulualueista (*tablespace*). Niitä voi olla kannassa yksi tai useampi. Taulualue puolestaan koostuu yhdestä tai useammasta tietokantatiedostosta. Taulualueita on kolmea eri tyyppiä. Pysyvä taulualue (*permanent*), kumoustaulualue (*undo*) sekä väliaikainen taulualue (*temporary*). Pysyvä taulualue nimensä mukaisesti säilyttää tiedon pysyvästi. Kumoustaulualueita tarvitaan, kun halutaan kumota kantaan tehdyt muutokset ROLLBACK-komennolla. Kumoustaulualueita voidaan käyttää myös palautuksessa. Väliaikainen taulualue on nimensä mukaisesti tarkoitettu väliaikaisen tiedon tallentamiseen. Jos esimerkiksi SQL-lauseessa halutaan tehdä lajittelu, niin tämä suoritettaisiin väliaikaisessa taulualueessa. /10; 13/

Toinen loogisen tason elementti on skeema (*schema*), joita on yleensä kannassa useampia. Näihin kuuluvat mm. taulut, indeksit, herättimet sekä proseduurit. Toisin sanoen siis skeema on kokoelma erilaisia tietokantaobjekteja. Taulualueet ja skeemat eivät ole yhteydessä toisiinsa, vaan yhden skeeman objekteja voi olla tallennettu useaan eri taulualueeseen. /14/

4.2 Instanssi

Oracle vaatii toimiakseen taustaprosesseja, jotta käyttäjät pääsevät käsiksi tietokantaan. Nämä prosessit varaavat resursseja tietokantapalvelimen keskusmuistista. Taustaprosesseja ja varattua keskusmuistia yhdessä kutsutaan Oracle-instanssiksi (kuva 13). /15/



Kuva 13 Oracle-instanssi /16/

Kuvassa 13 on useita taustaprosesseja. LGWR ja ARCH on jo käsitelty luvussa 4.1.1. Tässä ei käsitellä kaikkia prosesseja, joita kuvassa esiintyy vaan ainoastaan tärkeimmät. DBWR on tietokantaan kirjoittaja eli prosessi, joka kirjoittaa informaatiota tietokantapuskurista (*Database Buffer Cache*) tietokantatiedostoihin. Tietokantapuskuria käsitellään enemmän alaluvussa 4.2.2. PMON on prosessimonitori (*Process Monitor*), joka suorittaa prosessin palauttamisen, jos se epäonnistuu. SMON-prosessi (*System Monitor*) suorittaa palautuksen, kun keskeytynyt instanssi käynnistetään uudestaan. CKPT-prosessi (*Checkpoint*) antaa signaalin tietokantakirjoittajalle kirjoittaa tiedot tietokantapuskurista tietokantatiedostoihin. /15/

4.2.1 Instanssin käynnistyminen

Instanssi vaatii käynnistyäkseen erilaisia parametrejä mm. muistinvarauksesta sekä tiedostojen sijainnista. Nämä parametrit tallennetaan parametritiedostoon. Parametritiedostoja on kahdenlaisia. On olemassa teksti-pohjainen parametritiedosto, joka voidaan avata ja muokata käyttöjärjestelmässä. Tätä kutsutaan p-tiedostoksi (*Parameter file, pfile*). Lisäksi on olemassa binääri-tyyppinen parametritiedosto, joka ei ole selkokielineen. Tätä ei saa avata eikä muokata käyttöjärjestelmässä, vaan kaikki tapahtuu Oraclen kautta. Tätä tiedostoa sanotaan palvelinparametritiedostoksi (*Server Parameter file, spfile*). Erona näissä kahdessa on käytännössä se, että kun tekstipohjaisessa tiedostossa muokataan jotain parametriä, niin instanssi täytyy aina käynnistää uudestaan, jotta muutokset tulevat voimaan. Tämä tarkoittaa luonnollisesti ikävää käyttökatkosta itse tietojärjestelmässä. Palvelinparametritiedostoa käyttämällä sen sijaan on mahdollista muuttaa joitain parametrejä dynaamisesti ilman,

että instanssia tarvitsisi käynnistää uudestaan. Tämä on erittäin käytännöllistä kriittisissä järjestelmissä, joissa esimerkiksi kannan kuormitus vaihtelee paljon vuorokauden eri aikoina. Kuitenkaan lähes kaikkia parametrejä ei tällä binääritiedostolakaan pysty muuttamaan dynaamisesti. Parametritiedostossa olevilla parametreillä on todella suuri vaikutus kannan suorituskykyyn ja toimintaan, koska sillä ohjataan mm. seuraavaksi käsiteltävää instanssin muistirakennetta.

4.2.2 Muistirakenne

Oracle-instanssin muistirakenne rakentuu karkeasti kolmesta osa-alueesta. SGA (*System Global Area*), PGA (*Program Global Area*) sekä UGA (*User Global Area*). Nämä sijaitsevat tietokantapalvelimen keskusmuistissa.

SGA

Taulukossa 4 on esitetty SGA:n tärkeimmät komponentit.

Taulukko 4 SGA:n komponentit /16/

Komponentti	Kuvaus
Buffer Cache	Tietokantakyselyissä esiintyvät taulut ladataan tänne
Shared Pool	SQL-lauseet, paketit, tietoa objekteista (indeksit, taulut, näkymät, proseduurit jne.)
Redo Log Buffer	Insert-, update- ja delete-lauseet (Tietokannan muokkauslauseet)
Large Pool	Tätä käyttää MTS, RMAN.
Java Pool	Oraclen Java-proseduurien puskurialue.

Puskurissa siis säilytetään mm. tietokantatauluja. Sieltä hakeminen on luonnollisesti paljon nopeampaa kuin palvelimen kovalevyllä sijaitsevasta tietokantatiedostosta. SQL-lauseet ladataan jaettuun varantoon (*Shared Pool*), josta niitä on nopea käsitellä. Redo-lokipuskuriin ladataan tietokannan muokkauslauseet. Isoa varantoa (*Large Pool*) käytetään lähinnä, kun käytetään MTS-prosesseja, jolloin jokaiselle käyttäjälle ei varata omaa resurssia PGA:sta niin kuin varatuissa prosesseissa (*Dedicated Server*) tehdään. Java-varanto (*Java Pool*) on varattu Java-proseduureille,

joita yrityksen käyttämässä Aton-versiossa 3 ei ole, mutta tätä uudemmissa versioissa Java-varannolle annetaan suositusarvoksi 150 MB. /16/

Parametritiedostossa määritellään näille komponenteille allokoitavat muistimäärät. Taulukossa 5 on esitetty parametri ja tyypillinen arvo muistin varaukseen. Huomioitavaa on se että, jos MTS-prosesseja ei käytetä, ei isolle varannollekaan (*Large Pool*) ole käyttöä lukuunottamatta sitä, jos DBWR-prosesseja on useita. /16/

Taulukko 5 SGA:n muistiparametrit /16/

Komponentti	Parametri	Koko
Buffer (DB) Cache	Db_cache_size	200 MB - 600 MB
Shared Pool	shared_pool_size	300 MB - 500 MB
Redo Log Buffer	Log_buffer_size	512 kB - 3 MB
Large Pool	large_pool_size	4 MB →
Java Pool	java_pool_size	150 MB - 300 MB

32-bittinen Windows-ympäristö aiheuttaa rajoituksen SGA:n koolle. Se ei saa ylittää 1,2 gigatavua. Tämä tulee muistaa, kun muistiparametrejä säädetään. 64-bittisessä ympäristössä tätä rajoitusta ei ole.

PGA

PGA on muistialue, jolta varataan muistia Oraclen *server*-prosesseille. Toisin sanoen muistia varataan edellä mainituille taustaprosesseille sekä kaikille käyttäjäprosesseille, kun käytetään *dedicated server* -prosesseja. Tässä jokaiselle prosessille varataan oma alue PGA:n muistialueesta. PGA:ta käytetään SQL-lauseiden prosessointiin ja sessioinformaation säilyttämiseen. Parametritiedostossa PGA:n koon määrittää parametri *pga_aggregate_target*, jonka tyypillisesti tulisi olla ainakin 20 % SGA:n koosta /16/. Oracle yrittää pysyä tämän rajan alapuolella. Jos raja on liian pieni, se ylittyy helposti, ja kannan suorituskyky laskee. Jos em. parametrin arvoksi laittaa nollan, Oracle antaa PGA:lle 20 % SGA:n arvosta /15/. Toinen vaihtoehto PGA:n kontrollointiin on muuttaa seuraavia parametreja /50/:

- BITMAP_MERGE_AREA_SIZE
- CREATE_BITMAP_AREA_SIZE
- HASH_AREA_SIZE
- SORT_AREA_SIZE

Tämä on kuitenkin huomattavasti hankalampaa kuin koko PGA:n koon määrittäminen.

UGA

UGA on se muistialue instanssissa, joka pitää sisällään tietoa käyttäjäsessioista. UGA on talletettuna joko SGA:n tai PGA:n sisään riippuen siitä, miten käyttäjäsesioita hallinnoidaan Oraclen sisällä. Jos käytössä on varatut prosessit, on UGA PGA:n sisällä. Jos puolestaan käytössä on MTS, niin UGA on SGA:n sisällä. /49/

4.3 Suorituskyky

PDM-järjestelmä on tiedon määrältään huomattavasti suurempi kuin esimerkiksi ERP- tai tuotannonohjausjärjestelmät. Muissa järjestelmissä kantaa yleensä siivotaan turhista riveistä pois, joten niissä on vain muutama kymmenen tuhatta nimikettä. PDM-järjestelmässä nimikkeitä on miljoonia. Taulukossa 6 on esitetty muutamien yrityksessä käytössä olevien tietojärjestelmien nimikemäärät.

Taulukko 6 Nimikemäärät eri tietojärjestelmissä

Järjestelmä	Nimikemäärät
Mfg / Pro	25 365
Powered CRU	19 379
Powered MOB	27 093
TeamCenter	12 215
SAP	176 098
Aton	1 377 796

Kuten taulukosta 6 huomataan, ovat Atonin datamäärät huomattavasti suurempia verrattuna muihin. Tämä aiheuttaa luonnollisesti tarvetta mm. kannan kattavalle indeksoinnille ja indeksien jatkuvalle analysoinnille. Atonin käyttämistä tietokanta-
tauluista n. 60 %:iin on lisätty indeksit Oraclen omien indeksointien lisäksi nopeutamaan tietokantahakuja. Indeksejä käsitellään tuonnempana.

Kannan virittäminen on olennainen osa tietokannan ylläpitäjän työtä. Kantaa voi joutua virittämään joka päivä kannan kuormituksen mukaan. Käytännössä virittäminen tehdään muistiparametrejä säätämällä. Tämä on huomattavasti helpompaa, jos käytössä on binääriytyyppinen parametritiedosto. Kantaa voidaan kuitenkin nopeuttaa myös taulualueiden uudelleen organisoinnilla sekä indeksoinnilla.

4.3.1 Muistiparametrit

Muistiparametrejä on jo osittain käsitelty alaluvussa 4.2.2. Kannan suorituskykyyn vaikuttaa se, kuinka paljon tietokantaobjekteja on sijoitettuna tietokantapalvelimen keskusmuistiin. Mitä vähemmän tietokoneen keskusmuistia on varattuna Oraclelle, sitä enemmän se joutuu tekemään fyysistä lukua kovalevyiltä, mikä on luonnollisesti hitaampaa kuin suoraan keskusmuistista luku. Kanta toimii näin ollen hitaammin. Fyysinen luku aiheuttaa myös prosessorikuormaa enemmän kuin looginen keskusmuistista luku. Näihin voidaan vaikuttaa juurikin alaluvussa 4.2.2 mainittuja muistiparametrejä säätämällä.

Oraclen 10g -versiossa on myös käytössä automaattinen muistinhallinta (*Automatic Memory Management*). Tämä tarkoittaa käytännössä sitä, että ylläpitäjä määrää vain SGA:n kokonaiskoon. Oracle jakaa tämän muistin automaattisesti SGA:n eri komponenteille (ks. taulukko 4) /17/. Oracle käyttää muistiparametrien määrittämiseen sisäistä statistiikkaa /18/. Tämä helpottaa hallittavuutta, koska muistin kokomäärittäminen tarvitse antaa vain koko SGA:lle ja Oracle osaa itse päätellä tarvittavat arvot SGA-komponenteille. SGA:n kokonaiskoon parametrinä toimii *SGA_TARGET*. Toinen SGA:han liittyvä parametri on *SGA_MAX_SIZE*, joka määrittää kuinka paljon *SGA_TARGET* voi olla maksimissaan.

32-bittisessä Windows-pohjaisessa ympäristössä on rajoitus, jossa Windows varaa itsensä käyttöön aina puolet keskusmuistista. Tämä aiheuttaa luonnollisesti suuria rajoituksia kannan viritykseen. Ongelma käy hyvin ilmi suurissa ympäristöissä, joissa on paljon käyttäjiä kannassa. Jossain vaiheessa Oracle ei vain yksinkertaisesti saa allokoitua lisää muistia PGA:lle ja käyttäjät eivät pysty kirjautumaan kantaan. Tämä ongelma saadaan korjattua uudemmissa Windows-käyttöjärjestelmissä laittamalla *boot.ini*-tiedostoon ns. */3GB switch*, joka tarkoittaa sitä, että tietylle prosessille annetaan resursseja 3 gigatavua keskusmuistista. Windowsille jää 1 gigatavu, kun käytetään 32-bittisen Windows 2003 Server -ympäristön maksimikeskusmuistia (4 GB). Tässä tapauksessa prosessille oracle.exe annettaisiin 3 gigatavua keskusmuistia. Windows 2003 Server -perheen edistyneimmissä versioissa tuettu fyysisen RAM:n määrä on suurempi. Windows Server 2003 Enterprise Edition (SP1) ja Datacenter Edition tukee 32 gigatavua. Toisena vaihtoehtona on käyttää Intelin kehittämää PAE-optiota (*Physical Address Extension*), jolla muistia voidaan laajentaa jopa 64 gigatavuun. Tämä on tuettu useimmissa 32-bittisissä Windows-käyttöjärjestelmissä pois lukien Windows 2000 Server -käyttöjärjestelmä. Myöskään 64-bittisillä alustoilla tätä ei tueta. /46/

4.3.2 Taulualueiden uudelleenorganisointi

Oraclen tietokanta pirstoutuu (*fragmentation*) ajan myötä aivan samalla tavalla kuin tietokoneen kovalevykin. Pirstoutuminen aiheuttaa myös ongelmia tietokannan suorituskyvyssä. Tietokannan pirstoutumisen yhteydessä puhutaan sellaisesta termistä kuin ketjuuntumismäärä (*chain count*). Ketjuuntuminen tarkoittaa sitä, että taulun rivi on liian iso mahtumaan yhteen tietokantalohkoon (*Block*, ks. 4.1.1), jolloin Oraclen on pakko laittaa tieto useampaan lohkoon. Tämä ketjuuntuminen näkyy SQL:n select-lauseiden suoritus suunnitelmassa. Suoritus suunnitelmasta on kerrottu lisää alaluvussa 4.3.3. Kannan on pakko lukea useammasta lohkoista tietoa (*multiblock read*). Lohkon koko määritellään parametritiedossa parametrillä *db_block_size*, joka on tyypillisesti 4 - 8 kilotavua. /18; 19/

Tietokannan taulu voi pirstoutua kahdella eri tavalla. Ensimmäinen tapa on taulun *extenttien* lisääntyminen. Ilman ketjuuntumista tämä ei kuitenkaan aiheuta ongelmia suorituskyvyssä. Toinen, joka aiheuttaa suuren ongelman on juuri tietokantariivien pirstoutuminen. Tämä voi tapahtua esimerkiksi SQL:n UPDATE-lauseen kautta. Taulualueen sisällä pirstoutuminen tapahtuu taulujen tuhoamisella ja uudelleen luonnilla. Myös taulujen viennillä (*export*) ja tuonnilla (*import*) on vaikutusta tähän asiaan. /18/

Uudelleen organisointi voitaisiin tehdä koko kannan viemisellä (*export*) ja tuomisella takaisin (*import*). Tämä kuitenkin vaatisi koko kannan luomista uudestaan. Uudelleen organisointi on järkevämpää suorittaa siihen tehdyllä ohjelmistolla. Ohjelmisto on nimeltään *Capacity Manager for Oracle*. Taulukossa 7 on esitetty, miten Aton PDM -järjestelmässä taulualueiden uudelleenorganisointi voitaisiin suorittaa. Varsinaisille tauluille luodaan kolme taulualueutta. Samoin myös indekseille luodaan kolme taulualueutta. Taulujen koko määrää taulualueen. Jos taulun koko on alle 10 megatavua, tulisi se sijoittaa MSTDATA1-taulualueeseen. Jos taulun koko on yli 10 megatavua, mutta pienempi kuin 100 megatavua, tulisi se sijoittaa MSTDATA2-taulualueeseen. Jos taulun koko on yli 100 megatavua, tulee se taulualueeseen MSTDATA3. Indeksit sijoittuvat vastaaviin MSTIND-taulualueisiin sen mukaan, mihin tauluun indeksi on sidoksissa. Pienimmissä taulualueissa käytetään automaattista laajennustyyppiä (*auto-allocation*). Tämä tarkoittaa sitä, että Oracle pääättelee itse laajentamisen määrän. Kooltaan suuria tauluja sisältävät taulualueet tulevat saamaan laajennustypiksi tasaisen (*uniform*). Tämä tarkoittaa sitä, että Oracle laajentaa taulualueutta aina ennalta määrätyllä tavumäärällä. Tässä tapauksessa määrät olisivat 10 MB ja 100 MB. Tuleva koko -sarakeessa kerrotaan tarvittava tilamäärä, mitä vaaditaan objektien (taulujen tai indeksien) tallentamiseen. On muodostunut tavaksi tehdä tietokantatiedostoista 2 gigatavun kokoisia. Tämä juontaa juurensa FAT32-levyjärjestelmästä, jossa suurin sallittu tiedoston koko on n. 2 gigatavua.

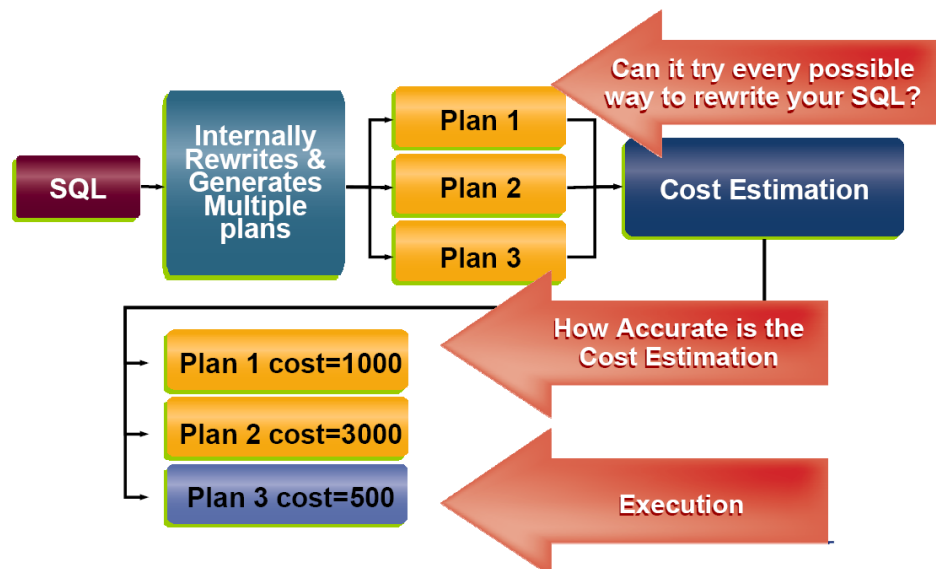
Taulukko 7 Taulualueet uudelleen organisoinnissa

TABLESPACE	Extent-tyyppi	Extent	Objektin koko	Objektien määrä	Tuleva koko	Todellinen koko
MSTDATA1	auto-allocation		< 10 MB	519	250 MB	2 GB
MSTDATA2	uniform	10 MB	10 MB - 100 MB	24	592 MB	2 GB
MSTDATA3	uniform	100 MB	> 100 MB	15	7,61 GB	8 GB
MSTIND1	auto-allocation		< 10 MB	789	650 MB	2 GB
MSTIND2	uniform	10 MB	10 MB – 100 MB	49	885 MB	2 GB
MSTIND3	uniform	100 MB	> 100 MB	99	12 GB	14 GB
Yht.				1495		30 GB

4.3.3 Indeksit ja tilastot

Indeksien käyttö tietokannassa parantaa merkittävästi haun nopeutta. Indeksi voidaan ajatella kirjan sisällysluettelona tai kirjan takana olevalla hakemistolla. Olisi todella hankalaa hakea kirjasta tietoa ilman jonkinasteista hakemistoa. Kirja pitäisi lukea kokonaan läpi tai ainakin siihen asti kunnes haluttu tieto löydetään. Näin toimii myös Oracle-tietokanta, jos indeksit eivät ole käytössä. Yksi taulu voidaan siis pahimmassa tapauksessa käydä kokonaan läpi (*full table scan*), jolloin haku kestää kauemmin. Tämä ei tosin ole mitenkään harvinaista.

Oraclessa on käytössä erityinen optimoija (*Cost Based Optimizer, CBO*), joka etsii parhaimman tavan suorittaa SQL-lause. CBO laskee kustannuksen (*cost*) erilaisille suoritussuunnitelmille ja valitsee näistä pienimmällä kustannuksella olevan suunnitelman. Tämä toiminta on esitetty kuvassa 14. /16/



Kuva 14 Optimoijan toiminta /16/

SQL-lauseen suorittaminen edellyttää aina jonkun taulun lukemista. Oracle voi tehdä taulun lukemisen monella eri tapaa. Tärkeimmät tavat ovat em. koko taulun lukeminen, ja toisena on nopea, indeksiä käyttävä suoritus.

CBO vaatii toimiakseen tilastotietoja eli statistiikkaa tauluista. Tilastot rakennetaan analysoinnilla, mikä tulisi suorittaa vähintään kerran viikossa. Jos kannan sisältämä informaatio ei muutu paljon, niin analysointia ei ole tarvetta suorittaa näin usein.

Tilastot pitävät sisällään taulusta mm. seuraavia tietoja /30, s. 360 - 361/:

- Taulun rivimäärä
- Käyttämättömien ja käytössä olevien lohkojen määrä (*block*)
- Vapaa tila käytössä olevissa lohkoissa
- Rivin keskimääräinen pituus
- Ketjuuntuneet rivit eli rivit, jotka eivät ole mahtuneet yhteen lohkoon.

Jos analysointi jätetään tekemättä, voimassa olevat tilastot vanhenevat. Tämä aiheuttaa sen, että CBO tekee huonoja päätöksiä suorituksen toteuttamisesta ja vaikuttaa näin ollen nopeuteen. /16/

4.4 Varmistukset ja palautus

Varmistusten ottaminen on missä tahansa tietojärjestelmässä erittäin tärkeää. Varmistaminen tarkoittaa sitä, että tietojärjestelmän informaatiotiedostot otetaan talteen tietyin väliajoin varmistetulle levyille tai nauhalle. Oraclen tapauksessa informaatiotiedostoihin kuuluisi tietokantatiedostot, ohjaustiedosto, arkistoidut lokit sekä parametritiedosto. Tätä kutsutaan Oracllessa fyysiseksi varmistukseksi. Toinen vaihtoehto on ottaa looginen varmistus, joka tarkoittaa eräänlaisen valokuvan ottamista tietokannasta.

4.4.1 Fyysinen varmistus

Fyysinen varmistus siis pohjautuu fyysisten tiedostojen kopioimiseen levyltä toiselle. Tämä voidaan tehdä kuumavarmistuksena (*online backup*) tai kylmävarmistuksena (*offline backup*). Kylmävarmistuksessa kanta ajetaan alas ja tämän jälkeen tiedostot kopioidaan toiselle levyille. Tämän jälkeen kanta nostetaan takaisin ylös. Isossa globaalissa järjestelmässä tämä menettely ei tule kysymykseen, koska järjestelmän täytyy olla koko ajan saatavilla. Kylmävarmistuksen toisena huonona puolella on se, että järjestelmää ei voida palauttaa kuin varmistushetkeen. Näin ollen tulee käyttää kuumavarmistusta. Kuumavarmistuksessa varmistus suoritetaan taulualue kerrallaan. Jos käytetään komentokieltä, niin jokaiselle taulualueelle annetaan käsky varmistuksen aloittamisesta, jolloin tietokantatiedostoihin ei kirjoiteta ennen kuin annetaan käsky varmistuksen lopettamisesta. Kuumavarmistus vaatii arkistointimoodin (*archive log mode*) kannassa. /16/

Toinen vaihtoehto tehdä kuumavarmistus on käyttää RMAN-apuohjelmaa (*Recovery Manager*). Koko tietokannan varmistus voidaan RMAN:ssa suorittaa BACKUP-komennolla. Tämä yksinkertainen komento varmistaisi tietokantatiedostot, ohjaustiedoston, ja parametritiedoston. Todellisuudessa varmistuskomennot eivät ole näin yksinkertaisia. Verrattuna perinteisiin komentokieli-pohjaiseen varmistukseen, RMAN-varmistus on monimutkaisempi ja vaatii paljon perehtymistä. Hyvänä puolella RMAN:in käytössä voidaan mainita ylläpidon vähyys varmistusrutiinissa.

Komentokielipohjaisessa varmistusrutiinissa tarvitsee tehdä muutoksia aina, kun uusi tietokantatiedosto lisätään kantaan, jolloin uusi tietokantatiedosto tarvitsee lisätä kopioitavaksi toiseen paikkaan. /20/

4.4.2 Looginen varmistus

Looginen varmistus otetaan export-työkalulla. Tämän yhteydessä puhutaan *dumpin* ottamisesta. Oracle luo yhden tai useamman dmp-päätteisen tiedoston, joissa kanta sijaitsee riippuen siitä, halutaanko tietokanta useampaan tiedostoon vai pelkästään yhteen. Yhden tiedoston käyttö on järkevämpää, koska se helpottaa varmistuksen hallintaa, jolloin ei tarvitse koko ajan seurata tuleeko tiedostot täyteen ja lisätä kommentoihin uusia tiedostoja.

Huonona puolena loogisessa varmistuksessa on sama kuin kylmävarmistuksessa. Tietokanta voidaan palauttaa vain tiettyyn hetkeen. On kuitenkin hyvä ottaa myös looginen varmistus kuumavarmistuksen lisäksi, koska loogisesta varmistuksesta voidaan palauttaa yksittäinen taulu esimerkiksi toiseen skeemaan. Tämä tulee hyödylliseksi sellaisessa tilanteessa, jossa SQL:n UPDATE-lause on tehty huonosti ja päivitys on ollut virheellinen. Taulu voidaan palauttaa edellisen yön varmistuksesta toiselle käyttäjälle (toiseen skeemaan), jolloin tätä tietoa hyödyntäen voidaan varsinainen käytössä oleva taulu päivittää edelliseen muotoonsa.

4.4.3 Palautus

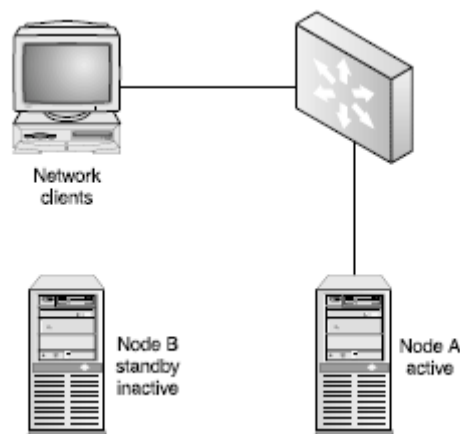
Tietokannan korruptoitua voidaan palautus siis suorittaa joko fyysisestä tai loogisesta varmistuksesta. Loogisesta varmistuksesta palautus vaatii tietokannan uudelleen luonnin, ja kantaa ei voida palauttaa kuin varmistushetkeen kuten aikaisemmin tuli jo mainittua. Palautus on parasta näin ollen tehdä fyysisestä kuumavarmistuksesta. Palautus voidaan tehdä tiettyyn aikaan sekunnin tarkkuudella tai vaihtoehtoisesti palauttaa jonkin arkistoidun redo-lokin perusteella. Myös ei-

arkistoidun redo-lokin kautta palautus onnistuu. RMAN-apuohjelmaa käyttäen palautusrutiini on hyvin samantyyppinen kuin, jos se tehtäisiin komentokielellä /21/.

Palautusrutiineissa on tärkeätä tietää, kuka palautuksen tekee ja miten tekee. Tässä kohtaa dokumentointi nousee tärkeään rooliin. Palautuksesta on hyvä olla dokumentti siitä, miten palautus tehdään. Sitä tulisi myös säännöllisin väliajoin harjoitella.

5 KLUSTEROINTI

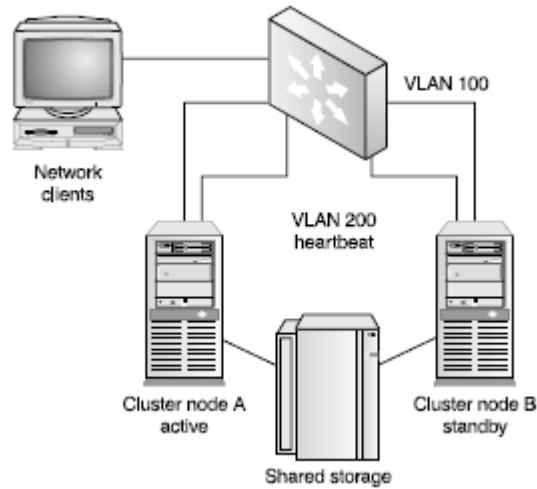
Klusteri koostuu joukosta tietokoneita (noodi), jotka työskentelevät yhdessä muodostaen yhden järjestelmän. Ennen klusterointi suoritettiin siten, että vain yksi palvelin oli käytössä ja toinen palvelin oli valmiina ottamaan tehtävät haltuunsa, jos toinen palvelin kaatui. Tämä välittömästi käyttöön otettavissa oleva varalaite on englanninkieliseltä termiltään *hot spare* (kuva 14). Siinä muutokset peilataan inaktiiviselle noodille. Tämä malli ei ennen estänyt käyttökatkosta järjestelmään. Asiakasohjelmat menettävät yhteyden primäärin noodin kaatuessa ja varanoodi pitää asettaa manuaalisesti primäärin noodin tilalle. /22, s. 6/



Kuva 14 Hot Spare /22, s. 6/

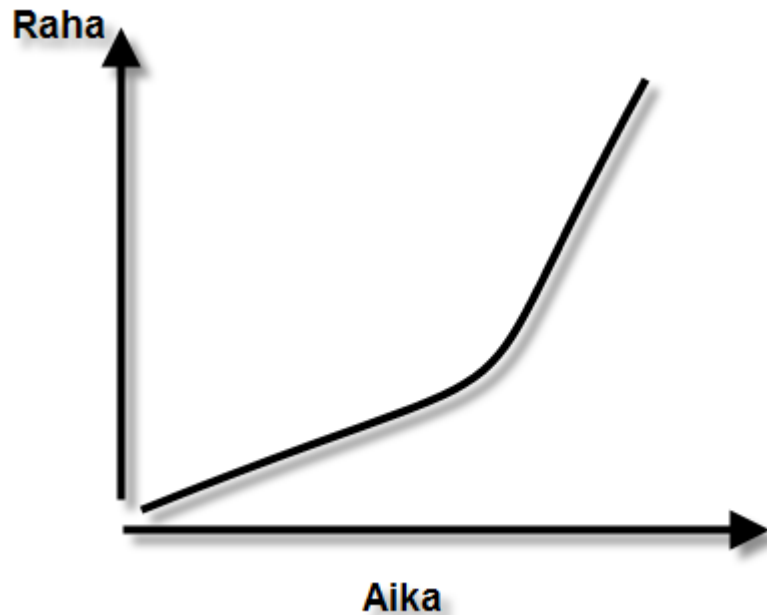
Nykyään tavoitteena on myös suorittaa kuormantasausta, jolloin voidaan käyttää kahden tai useamman noodin palveluja yhtäaikaaisesti. Kuorman kasvaminen tietojärjestelmässä aiheuttaa yleistä hitautta tai jopa sitä, että tietoihin ei pääse käsiksi lainkaan. /22, s. 3/

Kuvassa 15 on esitetty se, miten toinen noodi tietää toisen noodin kaatuneen. Tämä suoritetaan erityisellä liikenteellä noodien välillä (*heartbeat traffic*). Tätä samaa liikennöintiä käytetään myös kuormantasauksessa siihen tarkoitukseen, jos yksi noodeista kaatuu, tulee kuormantasaus laskea uudelleen jäljelle jääville noodeille. /22, s. 5/



Kuva 15 Klusterointi ja kuormantasaus /22, s. 5/

Klusteroinnilla ja kuormantasauksella haetaan sitä, että järjestelmä on koko ajan saatavilla (*High Availability*). Usein puhutaan viiden yhdeksikön (*five nines*) periaatteesta. Tämä tarkoittaa sitä, että järjestelmä on saatavilla 99,999 % ajasta. On selvää, että tietojärjestelmän virhetilanteessa yritys alkaa menettää rahaa ja mitä kauemmin tämä käyttökatkos kestää sitä enemmän rahaa menetetään. Järjestelmän virhetilanne ei heti vaikuta merkittävästi rahan menetykseen, mutta jossain vaiheessa menetetty raha aikayksikköä kohden nousee tuntuvasti. Kuvassa 16 on esitetty tämä ilmiö.



Kuva 16 Menetty raha ajan funktiona järjestelmän virhetilanteessa.

Aton PDM -järjestelmässä suunnittelematon käyttökatkos ei saisi kestää yli 15 minuuttia. Tänä aikana käyttökatkos vaikuttaa vain satunnaisiin käyttäjiin, joilla on tarvetta käyttää järjestelmää juuri sillä hetkellä. Tosin näitäkin voi olla enimmillään useita satoja. Kun aletaan puhua tunteista käyttökatkoksen kestona, Atonin rajapinnat alkavat näyttämään isoa roolia. Esimerkiksi nimiketietoutta ei saada menemään ERP-järjestelmiin tai varastonhallintaan. Varaosakirjoja ei saada tehtyä, mikä voi pahimmillaan aiheuttaa sen, että tuote ei pääse lähtemään tehtaalta eteenpäin varaosakirjan puuttuessa.

Käyttökatkoksen sijoittumisella vuorokauden aikaan on suuri merkitys. Käyttökatkos aamulla ei ole niin kriittinen kuin iltopäivällä. Tämä johtuu siitä, että aamulla vain noin kolmella tuotantoyksiköllä on tarve Atonin käyttöön. Iltopäivällä puolestaan noin kahdeksalla yksiköllä on tarve Atonin käyttöön.

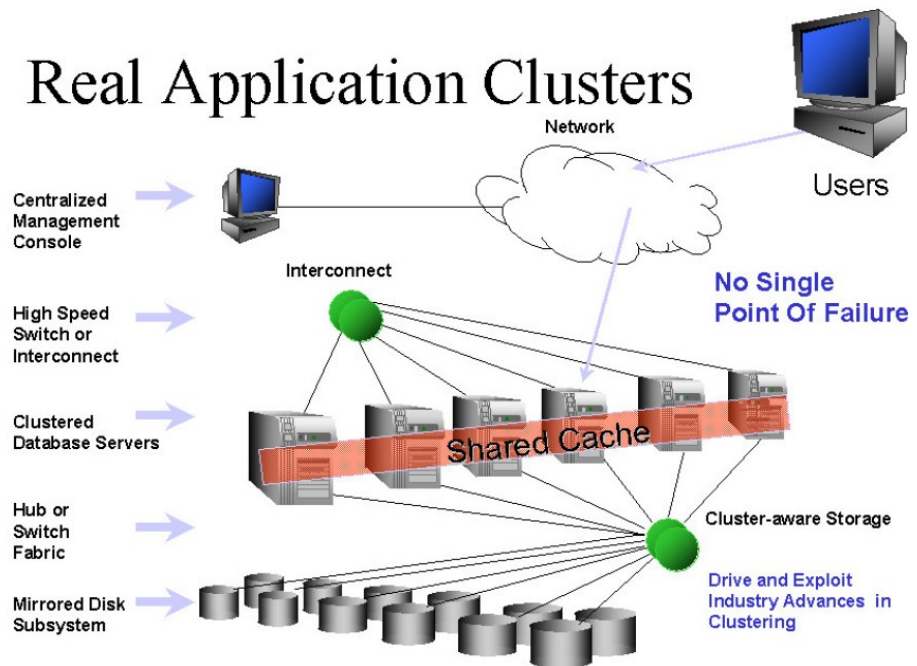
Järjestelmän saatavuus ei rajoitu pelkästään siihen, että noodeja järjestelmässä on useampi kuin yksi. Käyttäjännitteet noodeille tulisi tulla useasta virtalähteestä, jolloin virransyöttö voidaan ns. kahdentaa. Myös kriittisen verkon aktiivilaitteen viikaantuessa järjestelmä ei ole saatavissa, jolloin nämäkin tulisi kahdentaa. Lisäksi

palvelimien tulisi sijaita eri telineissä (*rack*), jolloin toisen telineen vikaantuessa, toinen teline on vielä saatavilla.

Klusterijärjestelmät voidaan jakaa kahteen eri osa-alueeseen. Sovellustason klusteroinnin toimintaperiaate on sama kuin mitä yllä tuli mainittua. Useita palvelimia laitetaan yhteen järjestelmään, ja jokaiseen asennetaan klusterointiohjelmisto, jolla voidaan hallita ja monitoroida esimerkiksi kuormantasausta ja noodien vikatilanteita. Erona laitteistotason klusterointiin on se, että tällä menetelmällä ei ole erikoislaitteistoa asennettuna yhdelle klusterinoodille. Verrattuna laitteistotason klusterointiin, tämä on huomattavasti helpompaa hallita ja konfiguroida. Klusterointijärjestelmistä ORACLE RAC sekä Windows Server 2003 ovat sovellustason klusterointijärjestelmiä. HP:n kehittämä HP-UX on puolestaan laitteistotason klusterointijärjestelmä. /23/

5.1 Oracle RAC

Oracle RAC (*Real Application Cluster*) on klusterijärjestelmä, joka on tehty Oracle DBMS:ää varten. Se noudattaa samoja periaatteita, mitä yllä on mainittu. Kuvassa 17 on esitetty tämä arkkitehtuuri. Järjestelmää siis pystyy hallitsemaan yhdellä päätelaitteella. Tietokantapalvelimet yhdistetään yhdeksi klusteriksi käyttäen virtuaalisia IP osoitteita. Jokaisella palvelimella on käynnissä oma Oracle instanssi, jotka ottavat yhteyden erilliseen levyjärjestelmään käyttäen esimerkiksi SCSI-väylää. Itse levyjärjestelmän tulisi olla vähintään RAID5-tasoa, jolloin yhden levyn hajoaminen ei vielä aiheuta käyttökatkosta.

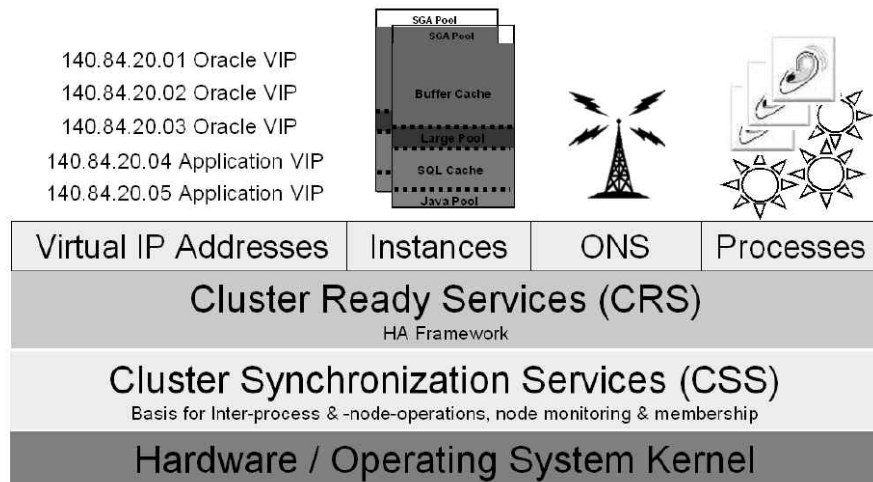


Kuva 17 Oracle RAC -arkkitehtuuri

Oracle RAC tarvitsee toimiakseen klusteriohjelmiston (*clusterware*), jaetun levyjärjestelmän sekä klusterinoodit eli palvelimet /25/. Näitä käsitellään seuraavaksi. /24, s. 4 - 6/

5.1.1 Klusteriohjelmisto

Klusteriohjelmisto monitoroi klusteria. Kun noodi käynnistetään, kaikki palvelut noodissa käynnistyvät automaattisesti. Noodissa olevan instanssin kaatumisen johdosta instanssi nostetaan automaattisesti uudelleen pystyyn. Kaikilla noodeilla tulee olla klusteriohjelmisto asennettuna. Ohjelmistolla on kolme eri tasoa. Nämä on esitetty kuvassa 18.



Kuva 18 Klusteriohjelmiston tasot /26/

Synkronointipalvelut (CSS) monitoroivat noodeja sekä huolehtivat nooiden välisestä liikennöinnistä. Kaikkia klusteriohjelmiston resursseja hallintaan valmiuspalvelulla (CRS) /26, s. 6/. CRS-palvelu hallinnoi muita prosesseja klusterirekisterin kautta (*OCR, Oracle Cluster Register*). Rekisterissä sijaitsee tieto miten muita palveluja voidaan pysäyttää, monitoroida sekä käynnistää /27, s. 28/. CRS hallinnoi mm. seuraavia palveluja /26, s. 6/:

- Globaalipalvelu (*Global Service Daemon, GSD*)
- Ilmoituspalvelu (*Oracle Notification Service, ONS*)
- Virtuaaliset IP-osoitteet (*Virtual IPs, VIPs*)

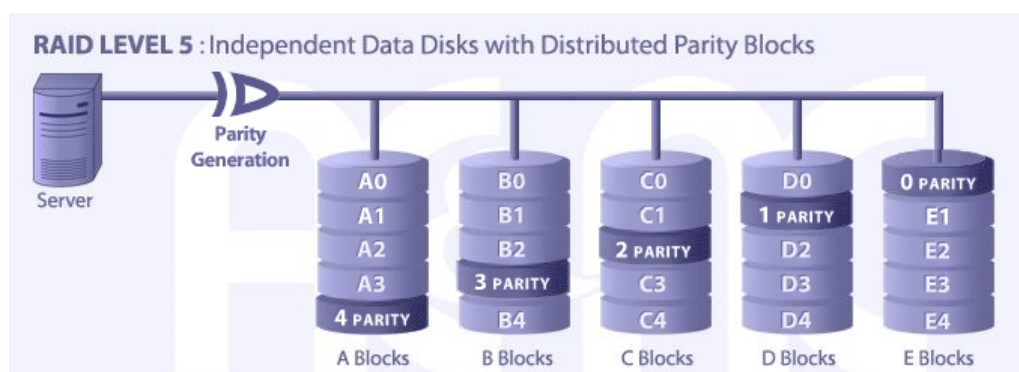
Näitä palveluja kutsutaan yhteisnimellä noodisovellukset. Globaalipalvelu huolehtii pyyntöjen vastaanottamisesta muilta Oracle-apuohjelmilta esimerkiksi instanssin uudelleenkäynnistämiseksi /28/. Ilmoituspalvelu puolestaan voi lähettää SMS- tai sähköpostiviestejä /29/. RAC vaatii yhden virtuaalisen IP osoitteen, jonka tulee olla samassa aliverkossa kuin lähiverkon. Tämä IP osoite annetaan yhdelle noodille kerrallaan. Jos noodi ei ole enää saatavilla, niin virtuaalinen IP osoite annetaan toiselle noodille. Näin saadaan lisää vikasietoisuutta järjestelmään. Noodeille täytyy luonnollisesti antaa myös normaalit IP osoitteet /25, s. 6/.

5.1.2 Levyjärjestelmä

Kaikki RAC-noodit ovat yhteydessä yhteen levyjärjestelmään. Tämä voi olla esimerkiksi verkkoon yhdistetty levy, SCSI-levy tai SAN. /25, s. 5/

Oracle suosittelee käyttämään automaattista muistin hallintaa (*Automatic Storage Management, ASM*). Tämä on sisällytetty Oracle 10g tietokanta-asennukseen. Oracle tukee myös Oracle Cluster -levyjärjestelmää (*Oracle Cluster File system, OCFS*), joka on saatavilla sekä Windows että Linux-ympäristöön. Tämä estää jae-
tussa levyjärjestelmässä olevien tiedostojen korruptoitumisen usean noodin lukies-
sa ja kirjoittaessa sinne.

Levyjärjestelmänä olisi hyvä käyttää vikasietoista levyjärjestelmää. RAID5-järjestelmä sopii hyvin tietokannan levyjärjestelmäksi. RAID5-järjestelmä vaatii vähintään kolme fyysistä levyä. Kahdelle levyllä kirjoitetaan data ja kolmannelle kirjoitetaan pariteettia. Näin ollen järjestelmä kestää yhden levyn rikkoutumisen. Kuvassa 19 on esitetty RAID5:n toiminta. Kuvassa datalohko kirjoitetaan neljälle eri levyllä (A0, B0, C0, D0) ja viidennelle kirjoitetaan pariteetti (0 parity). Toisella kirjoituskerralla pariteetti kirjoitetaan D-levylle ja niin edelleen. Useasta levystä koostuva RAID5-järjestelmä näkyy tietokoneen käyttöjärjestelmässä yhtenä asemana, jos siitä ei olla tehty loogisia osituksia (*partition*). /31/



Kuva 19 RAID5 /31/

5.1.3 Klusterinoodit

Metso Mineralsin Aton PDM -ympäristössä vaatimukset ovat noodien kannalta korkealla. Alustana tulee olla 64-bittinen prosessoriympäristö. Tämän on todettu laskevan prosessorikuormaa verrattuna 32-bittiseen ympäristöön. 1,2 gigatavun SGA-muistirajoitusta ei myöskään 64-bittisellä alustalla ole. Käyttöjärjestelmän tulee olla vähintään Windows 2003 Enterprise Edition, jolla voidaan ottaa käyttöön muistin allokointi suoraan Oraclelle (*/3GB-switch*). Tosin tämän hyödyllisyys riippuu siitä, että kuinka paljon keskusmuistia palvelimeen laitetaan. */32/*

Noodit tarvitsevat rajapinnan lähiverkkoon asiakasohjelmayhteyksiä varten. Lisäksi klusteri tarvitsee toisen yksityisen verkon, jolla tapahtuu noodien välinen kommunikointi (ks. Kuva 15). Tämä aiheuttaa noodeille tarpeen kahdesta verkkokortista. Toisen verkkokortin kautta kommunikoidaan asiakasohjelmien kanssa ja toisella toisten noodien kanssa. Noodien väliin tarvitsee sillan tai kytkimen, koska Oracle ei tue ristikytkettyä (*cross-over*) kaapelia noodien välissä. Ethernet-tekniikan tulisi olla ns. gigabit-ethernet tekniikkaa noodien välillä. RAC käyttää UDP-protokollaa noodien välisessä kommunikoinnissa.

5.1.4 Kuormantasaus

RAC:ssa on käytössä kaksi eri kuorman tasausmenetelmää. Asiakasohjelman kuormantasaus (*client-side load balancing*) sekä palvelimen kuormantasaus (*server-side load balancing*). */39, s. 8/*

Asiakasohjelman kuormantasaus

Asiakasohjelman puolella otetaan käyttöön lisäämällä `LOAD_BALANCING`-parametri `TNSNAMES.ora`-tiedostoon. Menetetyn tietokantayhteyden uudelleenmuodostamiseen voidaan käyttää samaisessa tiedostossa parametrilla `FAILOVER`. Tämä aiheuttaa ongelman Aton PDM -järjestelmässä, koska Aton-sovelluspalvelimelle ei asenneta Oraclea ja näin ollen myös `TNSNAMES.ora`-tiedosto puuttuu palvelimelta */47/*.

Vaikka Oracle RAC toipuukin nopeasti yhden noodin kaatumisesta, se ei välttämättä estä käyttäjäkatkosta. Yhden noodin vikaantuessa saattaa asiakasohjelma menettää yhteyden tietokantaan. Tämä voidaan korjata ottamalla TAF-optio (*Transparent Application Failover*) käyttöön Oracle RAC:ssa. Noodin vikatilanteessa TAF reitittää yhteydet uudelleen jäljellä jääneille noodeille ja käyttäjä ei näe virheilmoitusta yhteyden katkeamisesta tietokantaan. TAF tukee myös kuormantasausta sillä tavalla, että yhteydet jaetaan useimmalle noodille, jolloin kuorma noodia kohden pienee. Tämä optio ei tosin tue *Oracle JDBC thin* -ajuria, joka on Aton PDM -järjestelmässä suositeltava ajuri tietokantayhteyden muodostamiseen. TAF tukee OCI-ajuria, joka on saatavana Atonin tietokantayhteyteen. Tämä taas aiheuttaa sen, että Oracle tulee olla asennettua myös Aton-sovelluspalvelimelle, mikä vaatii yhden Oracle-lisenssin lisää. /40/

Toinen mahdollisuus on käyttää FCF-menetelmää (*Fast Connection Failover*), joka tukee myös *JDBC*-ajuria. Tämä puolestaan vaatii implisiittisen yhteysvarannon (*Implicit Connection Cache*) käyttöönottamista Oracle-tietokantayhteyttä hoitavassa Java-koodissa. Lisäksi Java-koodissa tietokanta tulee määrittää palvelunimenä eikä SID:nä /41, s. 105/. Kuvassa 20 on karsittu esimerkki tietokantayhteyden toteuttamisesta Javalla. Muutokset tietokantayhteyteen tulee ohjelmistotoimittajan toteuttaa ohjelmistoräätälöintinä.

```
OracleDataSource ods = new OracleDataSource()
...
ods.setUser("Scott"); //käyttäjätunnus
ods.setPassword("tiger"); //salasana
ods.setConnectionCachingEnabled(true); //Implisiittinen yhteysvaranto käyttöön
ods.setFastConnectionFailoverEnabled(true); //FCF käyttöön
ods.setConnectionCacheName("MyCache"); //Implisiittisen yhteysvarannon nimi
ods.setConnectionCacheProperties(cp); //implisiittisen yhteysvarannon ominaisuudet
ods.setURL("jdbc:oracle:thin:@(DESCRIPTION=
(Load_Balance=on)
(Address=(Protocol=tcp)(Host=vip1)(Port=1521))
(Address=(Protocol=tcp)(Host=vip2)(Port=1521))
(Connect_Data=(Service_Name=service_name)))");
//@-merkin jälkeen annetaan service name, joka koostuu mm. kuormantasausparametrista (LOAD_BALANCE)
//sekä noodien nimistä
```

Kuva 20 Esimerkki Java-tietokantayhteydestä /42, s. 15/

Jotta FCF voidaan ottaa käyttöön JDBC-rajapinnan yhteydessä, tulee myös ONS olla konfiguroituna. JDBC:n käynnistyksen parametreissa tulee olla viittaus ONS-

clientin sijaintiin, jolloin täytyy luoda yhteys Aton-sovelluspalvelimen ja tietokantapalvelimen välille. /42, s. 15/

Noodin vikatilanteessa JDBC saa RAC:lta ilmoituksen instanssin kaatumisesta ja sovellus voi yrittää yhteyttä tietokantaan uudestaan ja suorittaa uudestaan haluttu toiminto tietokantaan /42, s. 16; 41, s. 119 - 120, 123/

TAF:n ja FCF:n käytännön erona on se, että FCF:ia käytettäessä käyttäjä huomaa yhteyden tietokantaan katkenneen, mutta FCF yrittää muodostaa yhteyttä uudelleen instanssin noustessa takaisin ylös. Oracle ei kuitenkaan suosittele käyttämään TAF:ia ja FCF:ia yhtäaikaisesti. /41, s. 124 - 125/

Palvelimen kuormantasaus

Palvelimen kuormantasaus vaatii kaksi kuuntelijaa (*listener*). Yksi molemmille noodeille / Oracle-instansseille. Lisäksi, parametri REMOTE_LISTENER tulee asettaa parametritiedostoon (ks. luku 4.2.1), että tietokanta tietää hakea parametrin arvoksi annettua tekstiä TNSNAMES.ora tiedostosta. Kuuntelijat kommunikoiivat keskenään instanssin PMON-prosessin välityksellä. Tämän avulla saadaan tietoa kuinka monta käyttäjää on kirjautunut kuhunkin instanssiin. Uudet yhteydet voidaan ohjata niille noodeille, joilla on vähemmän käyttäjiä. Seuraavaksi on esitetty esimerkki palvelinkuormantasauksen käyttöönotosta /48/.

```
LISTENERS_RACDB =
```

```
(ADDRESS_LIST =
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = racora1-vip)(PORT = 1521))
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = racora2-vip)(PORT = 1521))
```

```
)
```

```
SSLB =
```

```
(DESCRIPTION =
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = raclinux1-vip)(PORT = 1521))
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = raclinux2-vip)(PORT = 1521))
```

```
(LOAD_BALANCE = ON)
```

```
(FAILOVER = ON)
```

```
(CONNECT_DATA =
```

```
(SERVER = DEDICATED)
```

```
(SERVICE_NAME = racdb)  
)  
)
```

Ensiksi määritellään palveluun kuuluvat kuuntelijat virtuaalisilla IP osoitteilla sekä Oraclen käyttämällä TCP-portilla. Tämän jälkeen määritellään tietokannan parametrit alkaen SID:stä, joka tässä tapauksessa on SSLB. Parametrejä kuormantasauksesta ja virhetilanteesta toipumiseen voidaan asettaa samalla tavalla kuin asiakasohjelman kuormantasauksessa.

5.2 Windows Server 2003

Windows Server 2003 käyttöjärjestelmäperheessä on myös omat sisäänrakennetut klusterointivaihtoehdot. Vaihtoehtoja on kaksi, palvelinklusteri sekä verkon kuorman tasaus (*Network Load Balancing, NLB*). Palvelinklusterointia suositellaan käytettäväksi mm. tietokannoissa. Verkon kuorman tasausta puolestaan voitaisiin käyttää web-palveluissa sekä palomuuereissa. Palvelinklusteri on saatavissa ainoastaan *Enterprise Editioniin* sekä *Datacenter Editioniin*, kun taas NLB on saatavissa kaikissa *Windows Server 2003* -tuoteperheen käyttöjärjestelmissä. /33, s. 5/

Windowsin sisäinen klusterointi ei ehkä sovi klusterointimenetelmäksi Oraclelle, mutta pelkkä tietokannan klusterointi ei riitä, sillä jos etupään palvelut vikaantuvat, ollaan taas samassa tilanteessa kuin ennen. Järjestelmä ei ole saatavilla. Tämän takia myös Aton-palvelut olisi hyvä klusteroida. /33, s. 12/

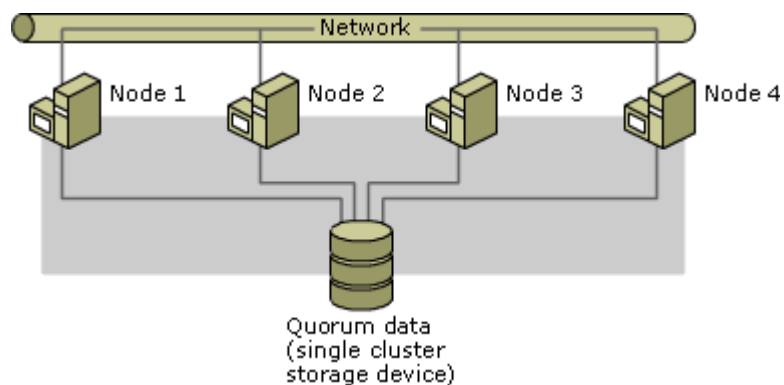
5.2.1 Palvelinklusteri

Myös Windows-palvelinklusteri lähtee siitä ajatuksesta, että yhteen järjestelmään lisätään useita noodeja. Yksi noodi on ns. primääri noodi, jonka virhe aiheuttaa sen, että varmistusnoodi ottaa primäärin noodin tehtävät hoitaakseen. Asiakasohjelmien

pyynnöt ohjataan välittömästi varmistusnoodille. On kuitenkin myös mahdollista käyttää useampaa noodia yhtäaikaaisesti järjestelmässä. /33, s. 6/

Kuten muutkin klusterijärjestelmät, Windows-klusteri käyttää jaettua muistilaitetta. Se, mikä noodi tämän *quorumin* omistaa, määrittää primäärin klusterinoodin. Kun omistajanoodiin tulee vika, *quorumin* omistajuus siirtyy varmistusklusterille. Tämä on esitetty kuvassa 21, jossa klusteri on toteutettu neljällä noodilla. /33, s. 6/

Quorumiin liittyy olennaisena osana *quorum log*, joka sisältää klusteriin liittyvät konfiguraatiot. Tämä tiedosto sisältää noodit, jotka ovat osana klusteria, ja mikä näiden noodien tila on. Lokitiedosto tulee olla kaikkien noodien nähtävillä. Siksi se on syytä laittaa jaetulle kiintolevyille. /44, s. 21/



Kuva 21 Neljän noodin klusteri /33, s. 6/

Tämä luonnollisesti aiheuttaa yhden heikon kohdan järjestelmään. Entä, jos itse muistilaitteeseen tulee vika? Silloin koko järjestelmä on pois käytöstä. Tämä on tärkeätä ottaa huomioon IT-infrastruktuuri -suunnittelussa. 33, s. 6/

Yksi vaihtoehto kiertää yhden muistilaitteen aiheuttama ongelma on käyttää MNS-klusterointia (*Majority Node Set*). Tässä teknologiassa jokaisella noodilla on oma muistilaitteensa. Yksi noodi on edelleen primääri noodi, jonka muistilaitteelta data replikoidaan (kopioidaan) muiden noodien muistilaitteille verkon välityksellä. Noodien ei tarvitse olla samassa lähiverkossa, mutta käytettäessä hitaita WAN-yhteyksiä, voi kaistanleveys tulla ongelmaksi datan replikoinnissa. /33, s. 6/

Molemmat palvelinklusterointia tukevista käyttöjärjestelmistä tukevat maksimissaan kahdeksaa noodia. Näiden käyttöjärjestelmien 64-bittiset versiot tukevat myös palvelinklusterointia, mikä mahdollistaa suurten keskusmuistien käytön. Käytössä voi olla jopa 4 Teratavua muistia per noodia. /34, s. 6/

5.2.2 Kolmannen osapuolen sovellukset

Ennen Windows-pohjainen klusterointi oli suunniteltu ainoastaan Microsoftin omille tuotteille kuten esimerkiksi *SQL Serverille* tai *Exchange Serverille*. Server 2003 tarjoaa kuitenkin tuen yleisille ohjelmistoille. Tämä antaa mahdollisuuden myös varsinaisen Aton PDM -järjestelmän klusterointiin. Atonin palvelumoduulit (ks. 3.2.2) tulee saattaa Windows-palveluiksi. Tämä aiheuttaa sen, että palvelimen hallitsemattoman uudelleenkäynnistyksen yhteydessä palveluita ei tarvitse uudelleen käynnistää manuaalisesti, vaan ne käynnistyvät automaattisesti, jos palvelut on näin konfiguroitu. Toinen positiivinen asia palvelumoduulien siirtäminen Windows-palveluiksi on tietoturva. Enää ei välttämättä tarvitse käyttää yhteistä käyttäjätunnusta, vaan jokainen voi käyttää omia tunnuksiaan kirjautuessaan palvelimelle. Tämä toisaalta aiheuttaa ongelman verkkolevyjen kanssa, joita käytetään rajapintoina tiedonsiirrossa muihin järjestelmiin. Verkkolevyt eivät ole enää saatavilla sen jälkeen kun se käyttäjä, jolla verkkolevyt on yhdistetty palvelimeen, on kirjautunut järjestelmästä ulos. Tämä voitaisiin ratkaista niin, että SAN:sta allokoitaisiin kahdelle eri palvelimelle sama lohko. Tämä ratkaisu ei kuitenkaan ole mahdollinen, jos klusterina on Windows Server 2003. Jos SAN:sta allokoidaan sama osio usealle palvelimelle, jotka eivät ole samassa klusterissa, johtaa tämä tiedon korruptoitumiseen. Tiedonsiirtokäskyissä onkin parasta käyttää UNC-polkuja verkkolevyjen asemesta. /43, s. 25/

Atonin kannalta palveluiden oikeaan klusterointiin vaadittaisiin ohjelmistotoimittajalta klusterointia varten tehdyt DLL:t. Tämä puolestaan aiheuttaa Atonin alustariippumattomuuden katoamisen. Atonia on tämän jälkeen mahdollisuus käyttää vain Windows-ympäristöissä. Mahdollisuus on kuitenkin käyttää Windows Server

2003 -perheen omia yleisiä DLL:iä. Nämä eivät todennäköisesti tule tarjoamaan riittävän hyvää vikasietoisuutta. /45/

5.3 HP-UX

HP-UX on HP:n kehittämä UNIX-pohjainen käyttöjärjestelmä, jota voidaan yhdessä HP:n toimittamien palvelimien kanssa käyttää järjestelmien klusterointiin. HP:n ja Oraclen vuosien yhteistyön takia HP-UX on yhteensopiva myös Oraclen kanssa sekä *Oracle Database* -ohjelmiston että *Oracle RAC* -ohjelmiston osalta. /35/

Aton PDM:n ohjelmistotoimittaja ei tue ko. käyttöjärjestelmää, jolloin käyttöönotto käsiteltävään tietojärjestelmään ei ole suositeltavaa, mutta ei täysin poissuljettua. Metson palvelinhallinnan siirtyminen HP:lle aiheuttaa kuitenkin tietotarvetta HP-UX-käyttöjärjestelmästä

HP-UX 11i v3 -versiossa on neljä erilaista käyttöympäristöä /36/.

- *HP-UX 11i v3 Data center OE (DC-OE)*
- *HP-UX 11i v3 High Availability (HA-OE)*
- *HP-UX 11i v3 Virtual Server (VSE-OE)*
- *HP-UC 11i v3 Base OE (BOE)*

5.3.1 Pääkomponentit

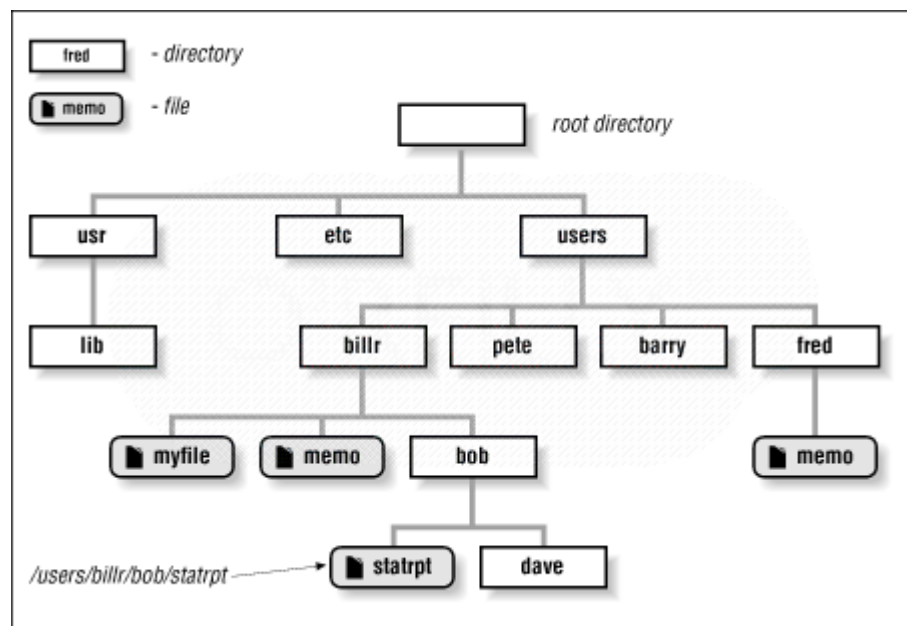
Kernel

HP-UX:n toiminnan pohjana toimii ydin (*kernel*). Kaikki käyttöjärjestelmässä tapahtuva toiminta on jollain tavalla sidoksissa tähän. Käynnistettäessä käyttöjärjestelmää eräs ensimmäisistä asioista mitä tapahtuu, on ytimen lataaminen kovalevyllä keskusmuistiin. Ydin koostuu useista moduuleista, jotka hoitavat mm. muistin hallintaa ja ajureita. Näitä moduuleja voidaan lisätä ja poistaa tarpeen mukaan. Ydintä

voidaan myös virittää tarpeen mukaan. Esimerkiksi kuinka paljon prosesseja voi olla aktiivisena samalla hetkellä, ja miten muistia allokoidaan ytimen sisällä. /37, s. 47 - 48/

Kansiorakenne

Käyttöjärjestelmän kansiorakenne perustuu hierarkkiseen malliin aivan kuten muissakin UNIX- sekä Windows-käyttöjärjestelmissä. Juurikansio on ylimpänä hierarkiassa ja sen polku on "/". Kaikki muut kansiot ovat tämän alla. Kuvassa 22 on esitetty hakemistohierarkia.



Kuva 22 HP-UX:n kansiorakenne /38/

Kuvassa 22 näkyy juurikansio ylimpänä. *etc*-kansio on tarkoitettu käyttöjärjestelmän konfiguraatitiedostojen säilyttämiseen. Tämä voi sisältää esimerkiksi tiedostoja käyttöjärjestelmän käynnistämisen ja sammutusrutiineista. Lisäksi kansio voi sisältää tietoa käyttäjistä ja tietokoneen verkkoasetuksista. *Uusr*-kansio sisältää sellaista tietoa, jota ei tarvita esimerkiksi käyttöjärjestelmän käynnistykseen yhteydessä. Täällä sijaitsevat tiedostot ovat usein vain lukua varten, joten niitä ei pääse muokkaamaan. *Users*-kansio sisältää luonnollisesti käyttäjien henkilökohtaisia tiedostoja. /37 s. 50 - 51/

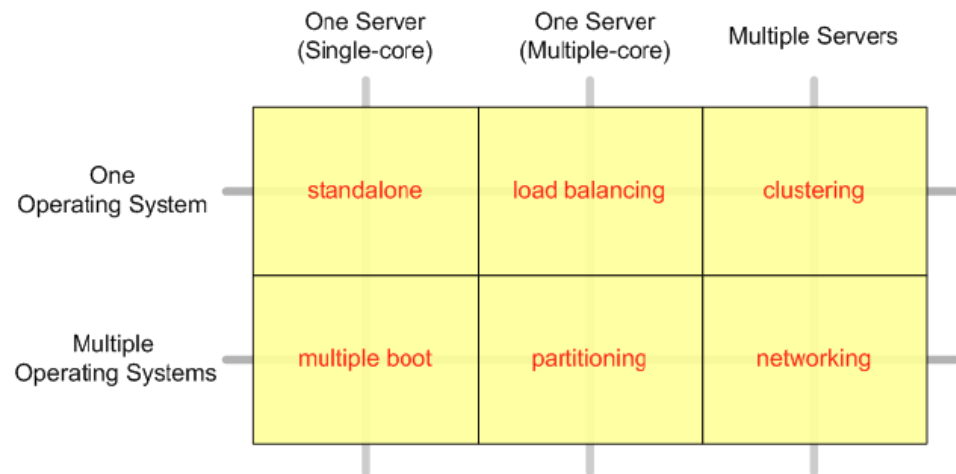
Tiedon varastointi

Tietoa HP-UX-pohjaiseen tietokoneeseen voidaan varastoida monella tavalla. Voidaan käyttää lokaaleja kovalevyjä, USB-levyjä, SCSI-levyjä, RAID-levyjä, SAN-levyjä tai magneettinauhoja jne. Näitä voidaan käyttää käyttäjien tiedostojen tallentamiseen, sivutukseen, varmistuskohdan luomiseen tai tietokannan tallentamiseen kovalevylle. /37, s. 52 - 53/

Fyysiset kovalevyt yhdistetään usein isoksi varastointitulaksi. Tämä voidaan jakaa useisiin loogisiin varastoihin. Yksi looginen varasto voi sijaita useammalla fyysisellä levyllä. HP-UX:ssä tiedon varastoinnin hallintaan on olemassa kaksi ohjelmaa, joilla tehdään loogisia varastoja. Ensimmäinen on *Logical Volume Manager (LVM)*, joka on oletushallintatyökalu 11i v3 -versiossa. Toinen hallintatyökalu on *VERITAS Volume Manager (VxVM)*, jonka ominaisuudet on hieman paremmat kuin LVM:ssä. Loogisia volyymejä voidaan tarvittaessa pienentää tai suurentaa. Huomioitavaa on, että HP-UX ei tue NTFS-tiedostojärjestelmää, joskin vastaavia HP:n kehittämiä tiedostojärjestelmiä käyttöjärjestelmä tukee. Myös vanha FAT32 on tuettu. /37 s. 57 - 58/

5.3.2 Virtuaalisointi ja klusterointi

HP jakaa virtuaalisoinnit useaan eri kategoriaan riippuen prosessoriytimien, palvelinten sekä käyttöjärjestelmien määrästä. Kuvassa 23 on esitetty tämä jako. /37, s. 27 - 28/



Kuva 23 Virtuaalisointitekniikat /37, s. 28/

Kuvassa 23 yhden käyttöjärjestelmän sekä yhden ytimen prosessori edustaa perinteistä palvelinajattelua. Yhdellä käyttöjärjestelmällä, mutta moniytimisellä prosessorilla voidaan suorittaa kuormantasausta eri ytimille. Klusterointitekniikkaa edustaa se, että käytössä on useampi palvelin, mutta kaikissa on sama käyttöjärjestelmä. Jos käytössä on yksiytiminen prosessori, mutta palvelimeen on asennettu useampi käyttöjärjestelmä, voidaan palvelin käynnistää useaan käyttöjärjestelmään tarpeen mukaan tosin luonnollisesti vain yksi kerrallaan. Moniytimisellä prosessorilla ja useammalla käyttöjärjestelmällä voidaan suorittaa ositusta. Useat käyttöjärjestelmät ja palvelimet muodostavat palvelinverkon. Tässä aliluvussa selvitetään kuorman tasausta yhdellä käyttöjärjestelmällä, moniytimisellä prosessorilla sekä klusterointia yhdellä käyttöjärjestelmällä, mutta usealla palvelimella. /37 s. 28/

Kuormantasaus

Kuormantasauksella tässä kohtaa tarkoitetaan laskentakuorman jakamista eri prosessoriytimille. Kuormantasauksella voidaan suorittaa resurssien allokointia tärkeille sovelluksille. Tähän liittyy useita apuohjelmia ja teknologioita, jotka on lueteltu seuraavaksi. /37 s. 30 - 31/

- *Process Resource Manager (PRM)*
- *Processor Sets (PSETS)*
- *Workload Manager (WLM)*
- *Instant Capacity (iCAP)*
- *Pay per Use*

PRM hallinnoi resursseja silloin, kun joko prosessorin kuorma, muistinkäyttö tai kovalevyn I/O on 100 %. PRM:ään luodaan erilaisia ryhmiä prosesseille ja käyttäjille. Nämä ryhmät sidotaan tiettyihin resursseihin kuten esimerkiksi prosessoriytimeen tai muistialueeseen. Näin estetään toisten ryhmien resurssien käyttö. /37 s. 31/. PRM on asennettu oletuksena BOE-käyttöympäristössä ja voidaan hankkia myös muihin käyttöympäristöihin. /37 s. 30/

PSETS on teknologia, jolla voidaan ryhmitellä usea prosessori(ydin) yhdeksi resurssiksi, jota voi käyttää vain yksi sovellus. Tämä on tärkeää kriittisissä sovelluksissa, jolloin vähemmän tärkeät sovellukset eivät pääse käsiksi varattuun resurssiin. Teknologiaa käytetään usein yhdessä PRM:n kanssa. /37 s. 33/

Jos kuorma palvelimella muuttuu usein, tarvitaan WLM:ia suorittamaan dynaamista resurssien allokointia resursseille tarpeen mukaan. PRM:ssä resurssien jako suoritetaan staattisesti ja WLM:ssä tämä siis suoritetaan dynaamisesti kuorman mukaan.

ICAP ja *Pay per Use* ovat HP:n tarjoamia palveluja, jossa palvelimet toimitetaan tietyllä alustalla ja hinta määräytyy palvelinten kuorman ja käytössä olevien komponenttien suhteen.

Klusterointi

Seuraavaksi on lueteltu neljää erillistä klusteritekniologiaa, jota HP tarjoaa. Tässä keskitytään lähinnä ensimmäisenä mainittuun teknologiaan. /37, s. 38/

- *Serviceguard Clusters*
- *Extended Diastance Clusters*
- *Metropolitan Clusters*
- *Continental Clusters*

Serviceguard-klusteri

Serviceguard-pohjaiset klusterit koostuvat ryhmästä HP:n Integrity-palvelimia tai 9000-sarjan palvelimia. Palvelimissa on kahdennettu ratkaisu sekä sovellustasolla että laitteistotasolla niin, että yhden komponentin vikaantuminen ei estä järjestelmän toimintaa. Tämä klusteriteknologia toimii usein yhdessä RAID-levyjen kanssa. Palvelimissa on myös HP:n keskeyttömät virtalähteet (*HP Powertrust power supplies*). Klusterissa määritellään paketteja noodeille, joita on kolme kappaletta serviceguard-klusterille. /37 s. 38/

- *Failover*
- *Multi-node*
- *System Multi-node*

Failover-paketti lähtee *hot spare* -mallista aivan kuten Windows Server 2003 -käyttöjärjestelmä alaluvussa 5.2.1. Yksi noodeista on primääri noodi ja sen vikaantuessa toinen noodi ottaa tehtävät hoitaakseen. Tämä tunnetaan ottonoodina (*adaptive node*). Tehtävien siirto toiselle noodille aiheuttaa pienen katkon käyttöön tai ei ollenkaan. /37 s. 38/

Multi-node -paketissa jokaisella noodilla ajetaan samanaikaisesti kopiota samaisesta paketista. Klusteri toimii niin kauan kun yksikin paketti on toiminnassa. HP tukee tätä tapaa vain tietyissä sovelluksissa. /37 s. 38-39/

System Multi-node -pakettia käytetään kaikilla noodeilla samanaikaisesti. Järjestelmä on toiminnassa niin kauan kuin kaikissa noodeissa on paketti käynnissä. HP tukee tätä tapaa vain tietyissä applikaatioissa. /37 s. 39/

Serviceguard-klusterissa kommunikointi tapahtuu luotettavan TCP/IP -protokollapinon avulla. Verrattuna Oraclen RAC:n käyttämään UDP-liikenteeseen, tämä on luotettavampi mutta hitaampi menetelmä. /37 s. 39/

Etäklusterit

Edellä käsitelty klusteri lähtee siitä ajatuksesta, että noodit sijaitsevat samassa tilassa. HP:n laitteilla on myös mahdollista tehdä ns. etäklusterointia. Näitä ovat edellä mainitut pitkän välimatkan, metropoliitan ja mantereen klusterit. Näissä tekniikoissa datan replikointi tulee tarpeeseen ja siksi noodien välinen verkko täytyy olla kaistanleveydeltään suuri. Noodien välinen välimatka voi olla jopa tuhansia kilometrejä. /37 s. 39 - 40/

6 TULOKSET JA YHTEENVETO

Työn alkupuoli toimii dokumentointina tietojärjestelmän sekä tietokannan arkkitehtuureihin. Tämä antaa yleisen kuvan siitä, miten työssä esitelty tietojärjestelmä toimii sekä tietokannan että varsinaisen Aton PDM -järjestelmän osalta.

Työn loppupuoli toimii esikartoituksena tietojärjestelmän vikasietoisuuden sekä vasteajan parantamiseen tulevan versiopäivityksen yhteydessä. Vikasietoisuutta voidaan parantaa käyttämällä moderneja klusterointimenetelmiä. Vasteaika voidaan parantaa päivittämällä järjestelmäpalvelinten laitteisto, käyttöjärjestelmä sekä muokkaamalla Oraclen muistiparametrejä ja taulualueita. HP:n osallistuminen palvelinhallintaan vähentää riskitekijöitä, koska sieltä löytyvä tietotaito palvelin- ja tietokanta-asioihin on kattavaa.

Päivitys on suunniteltu tapahtumaan kahdessa eri vaiheessa. Ensimmäisessä vaiheessa päivitetään laitteisto sekä käyttöjärjestelmä. Samalla palvelinhallinta siirtyy HP:lle. Toinen vaihe tulee sisältämään varsinaisen versiopäivityksen sekä Aton PDM:n että Oraclen osalta. Toisessa vaiheessa päivitetään myös palvelinlaitteistoa, koska sekä Oraclen että Aton PDM:n uudet versiot tukevat mm. 64-bittistä palvelinalustaa, joka omalta osaltaan parantaa järjestelmän suorituskykyä.

Vasteajan parantamiseksi painotettiin Oracllessa tehtäviä konfigurointeja kuten muistiparametrien muuttamista, indeksoinnin käyttöä ja analysointia sekä taulualueiden uudelleen organisointia. Muistiparametreillä ei voida merkittävästi parantaa järjestelmän vasteaika ennen kuin tietokantapalvelimen käyttöjärjestelmä on päivitetty uudempaan versioon. Tässä kohtaa on syytä huomioda myös Aton-liikenteen priorisointi Metson tietoverkossa, vaikka tätä näkökulmaa ei olla enempää käsitelty. Tämä parantaa järjestelmän käyttöä niillä käyttäjillä, jotka eivät ole järjestelmän tietokantapalvelimen kanssa samassa lähiverkossa.

Työ käsitteli erilaisia klusterijärjestelmiä, joista voidaan valita käyttöön sopivin vaihtoehto, kun klusterointia tehdään. Klusterointia ei tulla ottamaan suoraan käyttöön versiopäivityksen yhteydessä, koska tämä on liian radikaali ratkaisu tämän

kokoisessa ympäristössä. Jos klusterijärjestelmäksi tulisi Oracle RAC, aiheuttaisi tämä välittömiä ohjelmistoräätälöintejä tietokantayhteyteen, jos halutaan suorittaa kuormantasausta ja tietokantayhteyden palauttamista ilman asiakasohjelman uudelleenkäynnistystä. Kuormantasausta voidaan ottaa myös käyttöön palvelimen puolella, jolloin ratkaistaan JDBC:stä johtuva ongelma ainakin osittain. On kuitenkin mielenkiintoista nähdä miten pelkällä palvelimen virhesietoisuudella saadaan Atonin palvelumoduulit säilyttämään tietokantayhteys virhetilanteessa.

HP-UX:ää käsiteltäessä on syytä huomioida, että ohjelmistotoimittaja ei tue käyttöjärjestelmää, jolloin klusterointi tällä teknologialla on riskialtista. Koska HP-UX tarjoaa hyvää resurssienhallintaa, mitä esimerkiksi Windows ei tue, voidaan tätä kuitenkin pitää varteenotettavana vaihtoehtona. Aton on Java-pohjainen, joka tarkoittaa sitä, että sitä voidaan suorittaa kaikilla alustoilla.

Jos halutaan klusteroida Aton PDM:n palvelumoduuleja, on tämä ehkä järkevintä suorittaa Windows Server 2003 -käyttöjärjestelmässä. Atonin palvelumoduulien siirtyminen Javan virtuaalikoneista Windows-palveluiksi on väistämätöntä HP:n palvelinhallinnan myötä, mikä antaa mahdollisuuden myös Windows-pohjaiseen klusterointiin. Tämäkin tosin vaatii ohjelmistotoimittajalta toimenpiteitä tekemällä räätälöidyt DLL-tiedostot, joita käytetään klusteroinnissa. Tämä tekee Aton-ohjelmistosta Windows-riippuvaisen.

Järkevin tapa suorittaa klusterointi Oralessa olisi käyttää tähän nimenomaiseen tarkoitukseen tehtyä Oracle RAC -järjestelmää. Aton PDM:n perusasennuksella pääsemme ns. *hot spare* -malliin, josta voidaan kehittää edelleen kuorman tasausta sekä virhetilanteista palautumista järjestelmäräätälöinneillä tai kuormantasauksen käyttöönotolla Oracle-palvelimella.

Kaikissa tässä työssä käsitellyissä järjestelmissä on siis omat huonot puolensa. Oracle RAC:ssa on sen tuki JDBC-rajapinnalle. HP-UX:ssä on sen puuttuminen ohjelmistotoimittajan tuetuista käyttöjärjestelmistä. Windows Server:n rajoitteena on dll-tiedostojen käyttö sovellusten monitoroinnissa. Aton PDM:n järjestelmäarkkitehtuuri aiheuttaa myös omat hankaluutensa. Klusterointi tulee tehdä sekä Aton-

sovelluspalvelimella että Oraclessa, jotta käyttökatkokset saadaan ajaltaan mahdollisimman pieniksi. Tuloksena on siis "tuplaklusteri".

Yhtenä mahdollisuutena suorittaa korkean saatavuuden järjestelmäarkkitehtuuria, on käyttää virtuaalipalvelimia. Tällaisista järjestelmistä voidaan mainita mm. Microsoft Virtual Server 2005 sekä VMWare HA. Työn rajauksen puitteissa nämä on jätetty käsittelemättä. Tämän työn jatkona tullaan kuitenkin selvittämään mahdollisuudet käyttää virtuaalisointitekniologiaa.

Lähteet

1. Pöyry, Pekka, ohjelmistotekniikan lehtori. Luentomateriaali Tietokannat-kurssi (S4233), Tampereen ammattikorkeakoulu
2. Peltola, Sami, projekti-insinööri. Luentomateriaali 14.09.2007. Metso Minerals Oy
3. Aton PDM - Hallittavuutta tuotekehityksestä toimitukseen [verkkodokumentti] Saatavissa:
<http://www.modultek.com/linkkitiedosto.aspx?taso=1&id=121&sid=95> [Viitattu 19.02.2008]
4. Heikkilä Juhamatti, kehityspäällikkö. Luentomateriaali 06.09.2007. Metso Minerals Oy
5. Metso Minerals Intranet. [www-sivu]. [Viitattu 26.02.2008] (ei julkinen)
6. Aton 4 Data Sheet (DOC50418 ver. 2) [esittelylehtinen] Modultek Oy 2007 [Viitattu: 26.02.2008]
7. Maijanen, Janne, Luentomateriaali 30.05.2007. Modultek Oy
8. Chapter 1 - Introduction to ORACLE [www-sivu] Saatavissa:
<http://ugweb.cs.ualberta.ca/~c391/manual/chapt1.html> [Viitattu: 27.02.2008]
9. Introduction to the Oracle Database [www-sivu] Saatavissa: http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/intro.htm [Viitattu: 27.02.2008]
10. Exploring the Storage Structure of Your Database [www-sivu] Saatavissa:
http://download.oracle.com/docs/cd/B19306_01/server.102/b14196/storage001.htm#CHDHAFBA [Viitattu: 27.02.2008]

11. Control files [Oracle] [www-sivu] Saatavissa: <http://www.adp-gmbh.ch/ora/concepts/controlfiles.html> [Viitattu: 27.02.2008]
12. Managing the Redo Log [www-sivu] Saatavissa: http://download.oracle.com/docs/cd/B19306_01/server.102/b14231/onlineredo.htm#i1007497 [Viitattu: 27.02.2008]
13. Managing the Undo Tablespace [www-sivu] Saatavissa: http://download.oracle.com/docs/cd/B19306_01/server.102/b14231/undo.htm#sthref1464 [Viitattu: 27.02.2008]
14. Overview of Schema Objects [www-sivu] Saatavissa: http://download.oracle.com/docs/cd/B19306_01/server.102/b14196/schema001.htm#sthref453 [Viitattu: 28.02.2008]
15. Overview of an Instance and Instance Management [www-sivu] Saatavissa: http://download.oracle.com/docs/cd/B19306_01/server.102/b14196/instance001.htm#CACEIBFA [Viitattu: 28.02.2008]
16. Lindfors, Jani, tuotepäällikkö. Luentomateriaali 30.05.2007. Quest Software Oy
17. Memory Configuration and Use [www-sivu] Saatavissa: http://download.oracle.com/docs/cd/B19306_01/server.102/b14211/memory.htm#i58728 [Viitattu: 28.02.2008]
18. Oracle Data Warehouse Tablespace Reorganization [www-sivu] Saatavissa: http://www.dba-oracle.com/data_warehouse/tablespace_reorganization.html [Viitattu: 28.02.2008]
19. The Secrets of Oracle Row Chaining and Migration [www-sivu] Saatavissa: http://www.akadia.com/services/ora_chained_rows.html [Viitattu: 28.02.2008]

20. 4.1 Overview of RMAN Backups [www-sivu] Saatavissa: http://download-uk.oracle.com/docs/cd/B19306_01/backup.102/b14192/bkup001.htm [Viitattu: 29.02.2008]
21. Using RMAN to Restore and Recover a Database When the Repository and Spfile/Init.ora Files Are Also Lost [www-sivu] Saatavissa: <https://metalink.oracle.com> (Doc ID 372996.1) [Viitattu: 03.03.2008] (ei julkinen)
22. Shimonski, Robert. Windows 2000 And Windows Server 2003 Clustering And Load Balancing, Mcgraw Hill 2003 [e-kirja] 369 sivua
23. What is application clustering? - a definition from Whatis.com - see also: software clustering [www-sivu] Saatavissa: http://searchdomino.techtarget.com/sDefinition/0,,sid4_gci874062,00.html [Viitattu: 03.03.2008]
24. Oracle Real Application Clusters, Oracle Technical Whitepaper, Barb Lundhild 2005 [verkkodokumentti] Saatavissa: http://www.oracle.com/technology/products/database/clustering/pdf/twp_rac10gr2.pdf [Viitattu: 04.03.2008]
25. Administering Database Instances and Cluster Databases [www-sivu] Saatavissa: http://download.oracle.com/docs/cd/B19306_01/rac.102/b28759/adminrac.htm#TDPRC041 [Viitattu: 05.03.2008]
26. Oracle Clusterware 11g, Oracle Technical Whitepaper, Markus Michalewicz 2007. [verkkodokumentti] Saatavissa: http://www.oracle.com/technology/products/database/clusterware/pdf/TWP_Clusterware_11g.pdf [Viitattu: 05.03.2008]

27. Stroupe, Richard, Extending Oracle's High Availability Framework with 10g Cluster Ready Services (CRS). SELECT Journal Q2 2005 Volume 12, nro 4, s. 28 Saatavissa:
http://www.triconsulting.com/html/downloads/SELECTQ405_Stroupe.pdf
[Viitattu: 05.03.2008]
28. Real Application Clusters Software Architecture [www-sivu] Saatavissa:
http://download.oracle.com/docs/cd/B10501_01/rac.920/a96597/pssvarch.htm
[Viitattu 05.03.2008]
29. INSTALLATION [www-sivu] Saatavissa:
<http://www.oracle.com/technology/products/ias/portal/point/ums/installation.html> [Viitattu 10.03.2008]
30. Bersinic, Damir; Watson, John. Oracle Oracle Database 10g OCP Certification All-in-One Exam Guide, Mcgraw Hill 2005 [e-kirja] 892 sivua
31. AC&NC | RAID.edu - RAID Levels - RAID Level 5 - RAID 5 [www-sivu]
Saatavissa: http://www.acnc.com/04_01_05.html [Viitattu 10.03.2008]
32. Sisäinen muistio Metso Minerals Tampere, Aton PDM -klusteri plan
30.01.2008 (ei julkinen)
33. Windows Server 2003 Clustering [verkkodokumentti] Saatavissa:
<http://download.microsoft.com/download/d/d/7/dd75ece7-83de-45da-8bb1-cb233decf595/BDMTDM.doc> [Viitattu: 11.03.2008]
34. Technical Overview of Clustering in Windows Server 2003 [verkkodokumentti] Saatavissa: <http://download.microsoft.com/download/4/d/e/4de815ef-2904-420a-b726-e57de31ae63a/ClusteringOverview.doc> [Viitattu: 11.03.2008]
35. HP United States - Computers, Laptops, Servers, Printers & more [www-sivu]
Saatavissa: <http://www.hp.com> [Viitattu: 14.03.2008]

36. HP-UX 11i on HP Integrity servers [verkkodokumentti] Saatavissa:
<http://h20338.www2.hp.com/hpux11i/downloads/5982-7405EN.pdf> [Viitattu: 17.03.2008]
37. HP-UX System Administrator's Guide Overview [verkkodokumentti] Saatavissa:
<http://docs.hp.com/en/5992-3384/5992-3384.pdf> [Viitattu: 18.03.2008]
38. http://www.oreilly.com/catalog/korn/figs/I_1_tt4.png [kuva] [viitattu: 18.03.2008]
39. Using Oracle Real Application Clusters (RAC) [verkkodokumentti] Saatavissa:
http://www.datadirect.com/developer/odbc/docs/odbc_oracle_rac.pdf [Viitattu: 18.03.2008]
40. High Availability Using Transparent Application Failover On Oracle Real Application Clusters [www-sivu] Saatavissa: http://www.quest-pipelines.com/newsletter-v5/0404_A.htm [Viitattu: 31.03.2008]
41. JDBC Developer's Guide and Reference [verkkodokumentti] Saatavissa:
http://download-west.oracle.com/docs/cd/B14117_01/java.101/b10979.pdf
[Viitattu: 01.04.2008]
42. How to Build End-to-End Recovery and Workload Balancing for your Applications [verkkodokumentti] Saatavissa:
<http://www.oracle.com/technology/products/database/clustering/pdf/oowfan.pdf>
[Viitattu: 01.04.2008]
43. Server Clusters : Storage Area Networks [verkkodokumentti] Saatavissa:
<http://download.microsoft.com/download/7/b/5/7b555ca0-297d-4a04-a7ea-5f8b0089b249/SAN.doc> [viitattu: 07.04.2008]

44. Server Clusters : Architecture Overview [verkkodokumentti] Saatavissa:
<http://download.microsoft.com/download/0/a/4/0a4db63c-0488-46e3-8add-28a3c0648855/ServerClustersArchitecture.doc> [Viitattu: 07.04.2008]
45. Introducing Microsoft Cluster Service (MSCS) in the Windows Server 2003 Family [www-sivu] Saatavissa: <http://msdn2.microsoft.com/en-us/library/ms952401.aspx> [Viitattu: 07.04.2008]
46. Memory Support and Windows Operating Systems [www-sivu] Saatavissa:
<http://www.microsoft.com/whdc/system/platform/server/PAE/PAEmem.msp>
[Viitattu: 10.04.2008]
47. DBAsupport.com : Oracle 10g Central : Oracle 10gR2 RAC Load Balancing Features [www-sivu] Saatavissa:
<http://www.dbasupport.com/oracle/ora10g/LBA01.shtml> [Viitattu: 10.04.2008]
48. DBAsupport.com : Oracle 10g Central : Oracle 10gR2 RAC Load Balancing Features [www-sivu] Saatavissa:
http://www.dbasupport.com/img/LBA_features_Listing.html#List03 [Viitattu: 10.04.2008]
49. UGA - Oracle FAQ [www-sivu] Saatavissa: <http://www.orafaq.com/wiki/UGA>
[Viitattu: 14.04.2008]
50. Tuning PGA_AGGREGATE_TARGET in Oracle 9 [www-sivu] Saatavissa:
http://www.revealnet.com/newsletter-v5/0304_A.htm [Viitattu: 14.04.2008]