Partitioning and Green IT

Kåre Kristoffersen, Product Manager, Lector ApS
March 5, 2009
IT University of Copenhagen
Agenda

• About Lector
• Problems with large sets of data
• Vision: Information Lifecycle Management
• Oracle partitioning (Standard ILM)
• Lector’s DLCM – Data Lifecycle Management
• Templates for partitioning
• Maintenance of partitioning
• Reducing storage expenses
• Partial Restore
• Green IT
• Conclusion
Lector

- Established January 1, 2000
- 35 employees
- Turnover 2008 29 mio. DKK
- Gazelle Company
- Green IT Provider

IT company based on three legs:
- Standardsolutions
- Projects
- Consultants

Standardsolutions:
- TeamShare, ESS, DLCM

Technology Partnerships:
- Microsoft, Oracle
Problems with large sets of data

• **Performance:**
  – Response times get worse.
  – Time consumption for daily/weekly batch executions gets worse.

• **Managebility:**
  – Disk-usage grows unbounded.
  – Backup time grows.
  – Restore/Recovery time grows.

• **Increasing costs:**
  – Buying new disks.
  – Backup price grows when data amounts grows.
  – Down-time in case of disaster grows => increasing operational risk.

• **Environmental issues:**
  – Hosting 1.5 Terabyte data causes the emission 1 ton CO2 each year.
  – 2 percent of CO2 originates IT. This corresponds to emission from all planes.
  – 37 percent of companies keeps data in storage forever although not necessary.
Goal

- Reduce down-time
- Prioritize data and handle each set/group individually
  - Disk location
  - Compression
  - Read write / read only
  - Online / offline
- Improved economy in data management
- Ease administration and managebility of data.
- Reduce negative impact on environment from datacenters.

*ILM: Information Life Cycle Management.*
Vision: ILM
ILM: The process
Partitioning in Oracle

- Split tables and indexes into smaller pieces.
- Avoids impractical “history” tables.
- Partitioning methods in Oracle: RANGE, LIST and HASH.
- Two types of objects: Tables and indexes.
<table>
<thead>
<tr>
<th>Order_ID</th>
<th>office</th>
<th>item_type</th>
<th>items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Herning</td>
<td>Pump</td>
<td>3</td>
<td>04.01.2009</td>
</tr>
<tr>
<td>2</td>
<td>Herning</td>
<td>Pump</td>
<td>7</td>
<td>27.11.2008</td>
</tr>
<tr>
<td>3</td>
<td>Aalborg</td>
<td>Valve</td>
<td>4</td>
<td>02.02.2007</td>
</tr>
<tr>
<td>4</td>
<td>Herlev</td>
<td>Pipe</td>
<td>1</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>5</td>
<td>Roskilde</td>
<td>Pump</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>6</td>
<td>Vejle</td>
<td>Pump</td>
<td>14</td>
<td>28.10.2006</td>
</tr>
<tr>
<td>7</td>
<td>Aalborg</td>
<td>Pipe</td>
<td>10</td>
<td>10.01.2009</td>
</tr>
<tr>
<td>8</td>
<td>Herlev</td>
<td>Valve</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>9</td>
<td>Vejle</td>
<td>Valve</td>
<td>3</td>
<td>29.10.2006</td>
</tr>
<tr>
<td>10</td>
<td>Roskilde</td>
<td>Pump</td>
<td>8</td>
<td>10.06.2008</td>
</tr>
</tbody>
</table>

**Example:**

LIST Partitioning

```sql
create table orders(
  order_id number not null,
  items number not null,
  vare_nummer number not null,
  office varchar2(30 byte) not null,
  Order_Date date not null
partition by list (office)
(
  partition VestDK
  VALUES ('Herning','Vejle','Aalborg'),
  partition ØstDK
  VALUES ('Roskilde','Herlev'),
  partition OTHERS VALUES (DEFAULT)
);
```

Table Orders: Partition VestDK values ('Aalborg','Herning', 'Vejle')

<table>
<thead>
<tr>
<th>Order_ID</th>
<th>office</th>
<th>item_type</th>
<th>items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Vejle</td>
<td>Pump</td>
<td>14</td>
<td>28.10.2006</td>
</tr>
<tr>
<td>11</td>
<td>Aalborg</td>
<td>Pipe</td>
<td>10</td>
<td>10.01.2009</td>
</tr>
<tr>
<td>9</td>
<td>Vejle</td>
<td>Valve</td>
<td>3</td>
<td>29.10.2006</td>
</tr>
</tbody>
</table>

Table Orders: Partition ØstDK values ('Herlev', 'Roskilde')

<table>
<thead>
<tr>
<th>Order_ID</th>
<th>office</th>
<th>item_type</th>
<th>items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Roskilde</td>
<td>Pump</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>8</td>
<td>Herlev</td>
<td>Valve</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>10</td>
<td>Roskilde</td>
<td>Pump</td>
<td>8</td>
<td>10.06.2008</td>
</tr>
</tbody>
</table>

Tabel Orders: Partition Others values (DEFAULT)
<table>
<thead>
<tr>
<th>Order_ID</th>
<th>office</th>
<th>item_type</th>
<th>items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Herning</td>
<td>Pump</td>
<td>3</td>
<td>04.01.2009</td>
</tr>
<tr>
<td>2</td>
<td>Herning</td>
<td>Pump</td>
<td>7</td>
<td>27.11.2008</td>
</tr>
<tr>
<td>3</td>
<td>Aalborg</td>
<td>Valve</td>
<td>4</td>
<td>02.02.2007</td>
</tr>
<tr>
<td>4</td>
<td>Herlev</td>
<td>Pipe</td>
<td>1</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>5</td>
<td>Roskilde</td>
<td>Pump</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>6</td>
<td>Vejle</td>
<td>Pump</td>
<td>14</td>
<td>28.10.2006</td>
</tr>
<tr>
<td>7</td>
<td>Aalborg</td>
<td>Pipe</td>
<td>10</td>
<td>10.01.2009</td>
</tr>
<tr>
<td>8</td>
<td>Herlev</td>
<td>Valve</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>9</td>
<td>Vejle</td>
<td>Valve</td>
<td>3</td>
<td>29.10.2006</td>
</tr>
<tr>
<td>10</td>
<td>Roskilde</td>
<td>Pump</td>
<td>8</td>
<td>10.06.2008</td>
</tr>
</tbody>
</table>

Table Orders

<table>
<thead>
<tr>
<th>Order_ID</th>
<th>office</th>
<th>item_type</th>
<th>items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Herning</td>
<td>Pump</td>
<td>3</td>
<td>04.01.2009</td>
</tr>
<tr>
<td>2</td>
<td>Herning</td>
<td>Pump</td>
<td>7</td>
<td>27.11.2008</td>
</tr>
<tr>
<td>5</td>
<td>Roskilde</td>
<td>Pump</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>6</td>
<td>Vejle</td>
<td>Pump</td>
<td>14</td>
<td>28.10.2006</td>
</tr>
<tr>
<td>10</td>
<td>Roskilde</td>
<td>Pump</td>
<td>8</td>
<td>10.06.2008</td>
</tr>
</tbody>
</table>

Table Orders: Partition Pumps values ('Pump')

<table>
<thead>
<tr>
<th>Order_ID</th>
<th>office</th>
<th>item_type</th>
<th>items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Herlev</td>
<td>Pipe</td>
<td>1</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>7</td>
<td>Aalborg</td>
<td>Pipe</td>
<td>10</td>
<td>10.01.2009</td>
</tr>
</tbody>
</table>

Table Orders: Partition Pipes values ('Pipe')

<table>
<thead>
<tr>
<th>Order_ID</th>
<th>office</th>
<th>item_type</th>
<th>items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Aalborg</td>
<td>Valve</td>
<td>4</td>
<td>02.02.2007</td>
</tr>
<tr>
<td>8</td>
<td>Herlev</td>
<td>Valve</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>9</td>
<td>Vejle</td>
<td>Valve</td>
<td>3</td>
<td>29.10.2006</td>
</tr>
</tbody>
</table>

Table Orders: Partition Valves values ('Valve')

-- create table orders(
    order_id number not null,
    items number not null,
    vare_nummer number not null,
    office varchar2(30 byte) not null,
    Order_Date date not null
) partition by list (item_type)
(
    partition Pumps VALUES ('Pump'),
    partition Pipes VALUES ('Pipe'),
    Partition Valves VALUES ('Valve'),
    partition OTHERS VALUES (DEFAULT)
);
<table>
<thead>
<tr>
<th>Order_ID</th>
<th>Office</th>
<th>Item_type</th>
<th>Items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Herning</td>
<td>Pump</td>
<td>3</td>
<td>04.01.2009</td>
</tr>
<tr>
<td>2</td>
<td>Herning</td>
<td>Pump</td>
<td>7</td>
<td>27.11.2008</td>
</tr>
<tr>
<td>3</td>
<td>Aalborg</td>
<td>Valve</td>
<td>4</td>
<td>02.02.2007</td>
</tr>
<tr>
<td>4</td>
<td>Herlev</td>
<td>Pipe</td>
<td>1</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>5</td>
<td>Roskilde</td>
<td>Pump</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>6</td>
<td>Vejle</td>
<td>Pump</td>
<td>14</td>
<td>28.10.2006</td>
</tr>
<tr>
<td>7</td>
<td>Aalborg</td>
<td>Pipe</td>
<td>10</td>
<td>10.01.2009</td>
</tr>
<tr>
<td>8</td>
<td>Herlev</td>
<td>Valve</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>9</td>
<td>Vejle</td>
<td>Valve</td>
<td>3</td>
<td>29.10.2006</td>
</tr>
<tr>
<td>10</td>
<td>Roskilde</td>
<td>Pump</td>
<td>8</td>
<td>10.06.2008</td>
</tr>
</tbody>
</table>

Tabel Ordrer

<table>
<thead>
<tr>
<th>Order_ID</th>
<th>Office</th>
<th>Item_type</th>
<th>Items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Herning</td>
<td>Pump</td>
<td>3</td>
<td>04.01.2009</td>
</tr>
<tr>
<td>7</td>
<td>Aalborg</td>
<td>Pipe</td>
<td>10</td>
<td>10.01.2009</td>
</tr>
</tbody>
</table>

Tabel Ordrer: Partition Ordrer_2009 values less than 01-01-2010

<table>
<thead>
<tr>
<th>Order_ID</th>
<th>Office</th>
<th>Item_type</th>
<th>Items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Herning</td>
<td>Pump</td>
<td>7</td>
<td>27.11.2008</td>
</tr>
<tr>
<td>10</td>
<td>Roskilde</td>
<td>Pump</td>
<td>8</td>
<td>10.06.2008</td>
</tr>
</tbody>
</table>

Tabel Ordrer: Partition Ordrer_2008 values less than 01-01-2009

<table>
<thead>
<tr>
<th>Order_ID</th>
<th>Office</th>
<th>Item_type</th>
<th>Items</th>
<th>Order_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Aalborg</td>
<td>Valve</td>
<td>4</td>
<td>02.02.2007</td>
</tr>
<tr>
<td>4</td>
<td>Herlev</td>
<td>Pipe</td>
<td>1</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>5</td>
<td>Roskilde</td>
<td>Pump</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>6</td>
<td>Vejle</td>
<td>Pump</td>
<td>14</td>
<td>28.10.2006</td>
</tr>
<tr>
<td>8</td>
<td>Herlev</td>
<td>Valve</td>
<td>2</td>
<td>05.03.2007</td>
</tr>
<tr>
<td>9</td>
<td>Vejle</td>
<td>Valve</td>
<td>3</td>
<td>29.10.2006</td>
</tr>
</tbody>
</table>

Tabel Ordrer: Partition old_orders values less than 01-01-2008

create table orders(
  order_id number not null,
  items number not null,
  vare_nummer number not null,
  office varchar2(30 byte) not null,
  Order_Date date not null
) partition by RANGE (Order_date) (
  partition Orders_2009 VALUES LESS THAN to_date('01-jan-2010'),
  partition Orders_2008 VALUES LESS THAN to_date('01-jan-2009'),
  Partition old_orders VALUES LESS THAN to_date('01-jan-2008'),
  Partition OTHERS VALUES LESS THAN (MAXVALUE)
);
The DLCM Solution:

<table>
<thead>
<tr>
<th>Standard ILM</th>
<th>Lector DLCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformation to partitioning is a tedious and errorprone task:</td>
<td></td>
</tr>
<tr>
<td>• Create new table partitioned</td>
<td></td>
</tr>
<tr>
<td>• Copy data to new table</td>
<td></td>
</tr>
<tr>
<td>• Drop old table</td>
<td></td>
</tr>
<tr>
<td>• Rename</td>
<td></td>
</tr>
<tr>
<td>• Recreate all depending objects.</td>
<td></td>
</tr>
<tr>
<td>No highlevel support for maintenance of partitioning</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Single point of entry.</td>
<td></td>
</tr>
<tr>
<td>Templates for RANGE, LIST and HASH partitioning.</td>
<td></td>
</tr>
<tr>
<td>Automatic transformation to partitioned tables.</td>
<td></td>
</tr>
<tr>
<td>Transformation from one partitioning to another.</td>
<td></td>
</tr>
<tr>
<td>Rebuild related objects.</td>
<td></td>
</tr>
<tr>
<td>Validering af tilgængelighed af ressourcer (tablespaces and space) before</td>
<td></td>
</tr>
<tr>
<td>partitioning is executed.</td>
<td></td>
</tr>
<tr>
<td>Handling materialized views.</td>
<td></td>
</tr>
<tr>
<td>Partial Restore.</td>
<td></td>
</tr>
<tr>
<td>Maintenance of partitioning.</td>
<td></td>
</tr>
<tr>
<td>Automatic archiving of oldest partitions.</td>
<td></td>
</tr>
<tr>
<td>Customized partitioning operations.</td>
<td></td>
</tr>
</tbody>
</table>
DLCM Architecture

Web-browser

DLCM Repository
- Template mgt
- Planning
- Scripting
- Validation

Oracle 10g

DLCM Agent
- Scheduling
- Execution
- Logging

Target Database 1

Oracle/
DB2/
MS SQL

DLCM Agent
- Scheduling
- Execution
- Logging

Target Database 2
Range templates and life cycles

Example: RANGE template:

- 1st year partition into MONTHS tablespace TS1
- 2nd year partition into QUARTERS tablespace TS2, use compression
- 3rd year partition into HALF_YEARS tablespace TS3, compression + read only
- 4th year partition into YEARS tablespace TS4, offline

A concrete data lifecycle:

- Year 2009 partition into MONTHS tablespace TS1
- Year 2008 partition into QUARTERS tablespace TS2, use compression
- Year 2007 partition into HALF_YEARS tablespace TS3, compress + read only
- Year 2006 partition into YEARS tablespace TS4, offline
LIST and HASH templates

Example: LIST template

```
partition VestDK VALUES ('Herning','Vejle','Aalborg') tablespace TS1,
partition ØstDK VALUES ('Roskilde','Herlev') tablespace TS2,
partition OTHERS VALUES (DEFAULT) tablespace USERS
```

Example: HASH template

```
partition1 tablespace TS1,
partition2 tablespace TS2,
partition3 tablespace TS3,
Partition4 tablespace TS4
```

Defined in GUI, stored in tables
Partitioning maintenance

Move a partition to a tablespace:
alter table Orders move partition VestDK tablespace MY_TS

Split a LIST partition:
Alter table Orders split partition VestDK VALUES ('HERNING') into
  (partition Herning, partition AalborgVejle) update global indexes;

Rebuild an index partition:
Alter index Orders_IX rebuild partition Herning;

Split a RANGE partition:
alter table Orders split partition OTHERS at (to_date('01-jan-2011')) into
  (partition orders_2010 tablespace MY_TS, partition OTHERS) update global indexes;

Merge partitions:
Alter table Orders merge partitions Pipes, Valves into partition PipesAndValves;
Example

Example revealing business secrets removed. Cannot distribute!
DLCM implements new Oracle functionalities:

- Templates for RANGE, LIST and HASH partitioning
- Single Command for complete partitioning

Template: BIGEMP_TEMPLATE

<table>
<thead>
<tr>
<th>Partition Range</th>
<th>Tablespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>from 01-01-2008 to 31-12-2008</td>
<td>TS1</td>
</tr>
<tr>
<td>from 01-01-2007 to 31-12-2007</td>
<td>TS2</td>
</tr>
<tr>
<td>from 01-01-2006 to 31-12-2006</td>
<td>TS3</td>
</tr>
<tr>
<td>from 01-01-2005 to 31-12-2005</td>
<td>TS4</td>
</tr>
</tbody>
</table>

Please note: Such a template cannot be defined in Oracle, but in DLCM it is possible!!

Partitioning as a single command:

```
alter table BIGEMP partition by range (HIREDATE) template BIGEMP_TEMPLATE
including indexes rebuild all depending objects compute statistics with validation;
```

Please note: Such a command does not exist in Oracle, but DLCM implements this!!
- 50 Gigabytes data added to table each month.
- Storage prize: 8 Euro pr Gigabyte pr month.
- Without archiving, the monthly pay increases beyound all bounds.
- With DLCM data may be archived after ex. 3, 6 or 12 months.
- With DLCM the monthly storage expences become constant.
Partial Restore

- In Oracle tablespaces may be backed up, restored and recovered individually using Oracle Recovery Manager, RMAN.

- If tables/partitions are spread over one or more tablespaces in an ad-hoc manner, all tablespaces must be restored and recovered in case of disaster.

- A prioritized distribution of partitions over a set of tablespaces theoretically opens for the possibility of a co-called *partial restore*, in which tablespaces and their partitions are restored and recovered in a *prioritized order*.

- In a partial restore the most important data/partitions are recovered first, thus minimizing down-time, getting production available.

- Older or less important data/partitions only important for reporting and revision are recovered at a later stage.

- A database is divided into *containers*, each having a priority in a partial restore process. A container is a set of partitions from several tables.

- Technical: Global index rebuild, and simulation of off-line partitions. All scripted by DLCM. *Analogy: Lifeboat.*
Customer cases

• Finance
  – RANGE Partitionering on prod_date on all transaction tables.
  – Archiving/disposal after 15-18 months.
  – Improved performance of batch jobs.
  – Reduced expenses for storage, backup and mirroring.
  – Partial Restore after disaster.
  – 17 Terabyte data is disposed of.

• Insurance
  – LIST partitionering on large TIA tables.
  – Time consumed on batch reduced by 2 hrs pr night.
Green IT

Environmental impact:

• A Serial Attached SCSI disk for 4.5 Terabyte (1 Terabyte = 1000 Gigabyte) has an effect of 666 Watt (666 Joule pr sekund). The energy consumption for this disk annually will be:
  
  \[\text{666 Joule pr sec} \times \text{3600 seconds pr hour} \times 24 \text{ timer} \times \text{365 dage/3600000 joule pr kwH} = 6000 \text{ kwH.}\]

• 1 kwH causes the emission of \(\frac{1}{2}\) kg CO\(_2\), so the production of energy for this disk will cause the emission of 3 ton CO\(_2\) annually.

• In other words, the hosting of 1.5 terabyte data causes 1 ton CO\(_2\) annually.
Green IT

- IT is not "clean" and immaterial like mathematics.
- Production of data is an industry!
- Old data must be collected and be disposed of rather than kept forever.
- The information society is not as clean as we hoped. But we have tools to make it greener!
Green IT Perspectives

• If 200 companies or organizations reduce data capacity with 100 terabyte
  5 % of CO2 from all IT may be removed using Lectors DLCM.
• Storage becomes a ”sliding window”: Old data are removed in the same
  order as new data is born.
• From 2005 to 2006 Great Britain doubled its storage capacity in server
  parks.
• Lectors DLCM is by IT- and Telestyrelsen promoted as ”best practice” in
  guiding companies and organizations to introduce green IT.
• DLCM will appear in a new Idea- and knowledge catalogue for companies
  regarding green IT.
Other approaches to green IT

- Server consolidation.
- Improved cooling of serverparks (use cold seawater).
- \textit{Claim: DLCM addresses the CO2 problem from datacenters at its very core!}
- Work from home / virtual meetings
- Use power down equipment: printers/scanners/pc screens/etc.
Conclusion

• Automated process for data management, as opposed to manually cleaning up old data.
• Restore after disaster reduced from several days to few hours using partial restore.
• Up to 70% reduction in expenses for storage and backup.
• Information Lifecycle Management and partitioning gives green IT.
• Reduced expenses on resources to manual database administration, and reduced risk of human errors.
• ROI in less than one year.
Oracle 11g features

• Referential partitioning
• Composite partitioning
• Interval partitioning