Targets

✔ How To Help Yourself
  ✔ Overview on most important performance issues
  ✔ Performance Tools in SAP Service Marketplace

✔ Please do ...
  ✔ Measure the progress of your own performance tuning
    ✔ Measure before tuning
    ✔ Tune the system according to the tips and via using provided tools
    ✔ Measure after tuning
  ✔ Provide Feedback
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Settings and Checklist</td>
</tr>
<tr>
<td>Modeling</td>
</tr>
<tr>
<td>Query Performance</td>
</tr>
<tr>
<td>Extraction / Data Load Performance</td>
</tr>
<tr>
<td>Analysis Methods</td>
</tr>
</tbody>
</table>
Performance Tuning

- **OLTP Systems**
  - Application Development and performance tuning separated
  - Performance tuning by basis experts

- **BW**
  - Performance Tuning as holistic process over application design and database configuration
General Performance Rules

Basic Rules for Performance Optimization:

Be economic: Eliminate all unnecessary processes

Keep it small: Reduce the data volume to be processed

Do it parallel: Deploy parallelism on all available levels

Parallel processes are fully scalable
Basis Parameter

- SAP Recommendations according to SAP note 192658
- Also consider process settings: SAP BW does not need an UP2 process and at most 2 UPD processes

Database Parameters

- Check of most important parameters via transaction RSRV → database parameters (not available for all platforms)
- ORACLE DB parameter recommendations: SAP note 180605
- Informix DB parameter recommendations: SAP note 181945
- IBM DB2/UDB DB parameter recommendations: SAP notes 302429 (BW 2.x), 546262 (BW 3.x)
- IBM DB2/OS/390 parameter recommendations: SAP note 390016
- IBM DB2/iSeries parameter recommendations: SAP notes 307077, 501572, 541508
- SQL Server DB recommendations: SAP notes 327494, 28667
✓ **ORACLE-specific Database Tuning**
  ✓ ORACLE 8.1.x-scripts for improved meta data access as described in SAP note 565075
  ✓ Locally managed tablespaces: if you run BW 2.x, be sure that SAP notes 359835 and 387946 are applied. Be sure that your BW 3.x temporary tablespace is locally managed.
  ✓ Database statistics: SAP notes 351163 and 428212

✓ **Database Partitioning**
  ✓ Use DB partitioning, especially for large InfoCube tables
  ✓ ORACLE, Informix and DB2/OS390 support range partitioning, DB2/UDB supports hash partitioning
Checklist – 3 –

✓ **Web Application Tuning in BW 3.x**
  ✓ Use HTTP Compression to reduce network traffic: see SAP note 550669
  ✓ Use Client (browser) caching of images to reduce unnecessary network traffic: see SAP note 561792

✓ **BW 3.0B SP9 / BW 3.1 Content SP3 or higher**
  ✓ We strongly recommend these support packages
  ✓ Internal measurements have proven an average improvement of over 40% (compared to 3.0B SP4) in Web Application Performance

✓ **Sizing**
  ✓ Decisive Factors: Number and Speed of CPU’s, Main Memory, Disk Volume and Architecture (e.g. RAID), I/O-Controller
  ✓ Rough Sizing available in SAP Service Marketplace alias quicksizer
  ✓ In cooperation with hardware partner
Checklist – 4 –

✓ **Sound Data Model**
   ✓ The Data Model is one of the most important issues

✓ **BW Reporting Performance**
   ✓ Aggregates
   ✓ Pre-calculated Web Templates
   ✓ OLAP Cache

✓ **Extraction / Data Load Performance**
   ✓ Recommendations in SAP notes 130253 and 417307
Database (or Physical) Partitioning

- Database tables are cut into smaller chunks (partitions)
- One logical database table
- Transparent for user
- Available for the following database management systems
  - Range Partitioning: ORACLE, Informix, IBM DB2/390
  - Hash Partitioning: IBM DB2/UDB
Database (or Physical) Partitioning

- Benefits
  - Parallel accesses to partitions
  - Read smaller sets of data
  - Fast Deletion of partitions (DROP PARTITION instead DELETE FROM WHERE)

Automatically Partitioned Database Tables (for Range Partitioning)
  - InfoCube F-Fact table: partitioned by request
  - PSA table: partitioned by request
  - ODS Change Log: similar to PSA table

User Defined Partitioning Criteria (for Range Partitioning)
  - InfoCube E-Fact table
  - Partition Criteria: Time characteristics like month or fiscal period
  - Note: Both fact tables are extended by the SID of the chosen time characteristic
Database (or Physical) Partitioning

- Query Access (Partition Pruning)

Example:
Month = August 2003

Master Data

Dimension

Apply Restrictions to Dimensions

Fact Table

Apply Restrictions to fact table

Discard irrelevant partitions

F table has to be read, too
Distribution of workload

- **Application Server groups / Load Balancing**
  - Logon load balancing (via group login): this allows you to distribute the workload of multiple query/administration users across several application servers
  - Distribution of web users across application servers can be configured in the BEx service in SICF
  - ODS Object Data Activation is definable for specific server groups (BW 3.x)
  - Process Chains can be processed on specified server groups (BW 3.x)
  - Extraction in the mySAP source system can be processed on specified server groups: RFC destination in BW to source system must be defined accordingly (transaction SM59)
  - Data load in BW can be processed on specified server groups: RFC destination in source system to BW must be defined accordingly (transaction SM59)
  - Data staging via XML over HTTP/SOAP can be processed on specified server groups (BW 3.x)

- **Operation modes**
  - Different modes for mainly query processing (many dialogue jobs) and data load (several batch jobs)
Other Aspects

- Reorganization of Log Tables
  - SLG2 (SAP note 195157)
  - EDI40 (SAP note 179046)
  - RSDDSTAT-tables

- Check for unnecessary traces and logs that impact system performance
  - Authorization log (RSSMTRACE)
  - User Trace (RSRTRACE)
  - SQL Trace (ST05)
  - ABAP Trace (SE30)
  - Statistics Trace
Other Aspects

- Archiving (as of BW 3.x)
  - Reduction of number of records in InfoCube fact tables / ODS object tables

BW Functionality with ADK / 3rd party products
Performance Guidelines

- General Settings and Checklist
- Modeling
- Query Performance
- Extraction / Data Load Performance
- Analysis Methods
Checklist

✓ Design of Operational Store, Data Warehouse Layer and Multidimensional Model
✓ Dimensions of InfoCubes
✓ Logical (MultiProvider) Partitioning
✓ Time-Dependent Master Data
✓ Non-Cumulative Key Figures
Operational Data Store and Data Warehouse Layer

Data Warehouse
- Non volatile
- Granular
- Integrated
- Historical foundation
- Built with ODS Objects

Operational Data Store
- Operational Reporting
- Near Real-Time / Volatile
- Granular
- Built with ODS Objects

Multidimensional Models
- Multidimensional analysis
- Aggregated view
- Integrated
- Built with InfoCubes
InfoCubes

SAP-Extended Star Schema
- ONE Normalization of Dimensions
- Master Data is shared, not part of the InfoCube

Diagram:
- Fact Table
- Dimension
- Master Data
- Surrogate Key
- Dimension ID
- Dimension
- Dimension
- Dimension
- Master Data

© SAP AG 2002, BW Performance Tuning, Alex Peter, 21
InfoCubes

Master Data

- Hierarchies
- Attributes
- Texts (Multi-Language Support)

SID Table

Dimension

Fact Table

Hierarchies

Attributes

Texts

Customer | Street | Account

Plant UK | Plant Germany | Plant Australia
Line Item Dimensions

- Direct Link from InfoCube Fact Table to Master Data SID Table – without dimension table
- Only one characteristic is possible in a line item dimension
- Recommended for large dimensions (e.g. invoice number, order number, but also reasonable for large material or customer dimensions)
Line Item Dimensions

**Benefits**
- Saves one table join at query runtime
- Saves determination of dimension ID’s at data load time

**Disadvantages**
- F4 – Help not possible on dimension values (only on master data values)
MultiProvider (or Logical) Partitioning

- Possible partitioning criteria: year, plan/actual, regions, business area
- Parallel sub-queries are started automatically to basic InfoCubes

✔ Use MultiProvider Partitioning to cut large amounts of data in chunks
**MultiProviders (BW 3.x)**

- **Definition**
  - Combination of InfoCubes, ODS objects, InfoObjects and InfoSets
MultiProviders

Benefits

◆ No additional data storage
◆ Single InfoProviders smaller, less complex and less sparsely filled than one big InfoProvider
◆ (Parallel) Data load into individual InfoProviders
◆ Queries are split automatically and distributed to InfoProviders (parallel execution)
◆ Transparent usage for Reporting
◆ Local queries on each InfoProvider possible
◆ Archiving of single basic InfoProvider is very easy

Disadvantages

◆ Administration (with aggregates)
◆ Additional I/O
MultiProviders

Example for sparsely filled single InfoCube

<table>
<thead>
<tr>
<th>Order Number</th>
<th>Customer</th>
<th>Product</th>
<th>Order Date</th>
<th>Delivery Date</th>
<th>Billing Date</th>
<th>Order qty</th>
<th>Delivery Qty</th>
<th>Billing Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Imunex</td>
<td>Pen</td>
<td>2001</td>
<td>*</td>
<td>*</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Pasag</td>
<td>Ruler</td>
<td>2001</td>
<td>*</td>
<td>*</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

..... and the distribution to single InfoCube with consolidating MultiProvider:

InfoCube Order

<table>
<thead>
<tr>
<th>Order Number</th>
<th>Customer</th>
<th>Product</th>
<th>Order Date</th>
<th>Order qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Imunex</td>
<td>Pen</td>
<td>2001</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Pasag</td>
<td>Ruler</td>
<td>2001</td>
<td>8</td>
</tr>
</tbody>
</table>

InfoCube Delivery

<table>
<thead>
<tr>
<th>Order Number</th>
<th>Customer</th>
<th>Product</th>
<th>Delivery Date</th>
<th>Delivery Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Imunex</td>
<td>Pen</td>
<td>2002</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Pasag</td>
<td>Ruler</td>
<td>2002</td>
<td>8</td>
</tr>
</tbody>
</table>

InfoCube Billing

<table>
<thead>
<tr>
<th>Order Number</th>
<th>Customer</th>
<th>Product</th>
<th>Billing Date</th>
<th>Billing Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Imunex</td>
<td>Pen</td>
<td>2002</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Pasag</td>
<td>Ruler</td>
<td>2003</td>
<td>7</td>
</tr>
</tbody>
</table>
Time-dependent master data

- Master Data is assigned to a validity period
- Key date in query defines the valid characteristic values
- Example:

<table>
<thead>
<tr>
<th>Country</th>
<th>Valid From</th>
<th>Valid To</th>
<th>Sales Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK</td>
<td>2002/01/01</td>
<td>2002/08/31</td>
<td>Miller</td>
</tr>
<tr>
<td>UK</td>
<td>2002/09/01</td>
<td>2003/02/28</td>
<td>Stevenson</td>
</tr>
<tr>
<td>USA</td>
<td>2002/01/01</td>
<td>2002/10/31</td>
<td>Smith</td>
</tr>
<tr>
<td>USA</td>
<td>2002/11/01</td>
<td>2003/02/28</td>
<td>Miller</td>
</tr>
<tr>
<td>Australia</td>
<td>2002/01/01</td>
<td>2002/04/30</td>
<td>Leask</td>
</tr>
<tr>
<td>Australia</td>
<td>2002/05/01</td>
<td>2003/02/28</td>
<td>Cassidy</td>
</tr>
</tbody>
</table>

Example: Keydate 2002/09/15 delivers sales persons Stevenson, Smith and Cassidy for UK, USA and Australia, respectively, for all records within the result set
Time-dependent master data

- Note that a query on time-dependent attributes does NOT perform a join between the validity period and the InfoCube time dimension.
- Aggregates on time-dependent master data possible as of BW 3.x.
- Adjustment Process necessary when key date changes.
  - Usually expensive.

✔ Sometimes it is reasonable to model time-dependent attributes ALSO as time-independent to achieve a good response time for „today is yesterday“ - reports.
Non-cumulative key figures

- Non-cumulative key figures cannot be cumulated meaningfully over time, e.g. inventory, number of employees
- Storage: Historical movements and reference point (in E fact table)
- Reference point is updated when InfoCube is compressed; if all requests are compressed, the reference point represents e.g. the current stock
- Example:

<table>
<thead>
<tr>
<th>Month</th>
<th>Material</th>
<th>Plant</th>
<th>Material flow</th>
<th>Reference point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 02</td>
<td>4711</td>
<td>1000</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td><strong>Compressing</strong></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Feb 02</td>
<td>4711</td>
<td>1000</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Mar 02</td>
<td>4711</td>
<td>1000</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td><strong>Compressing</strong></td>
<td></td>
<td></td>
<td>35</td>
</tr>
</tbody>
</table>

- Note: the reference points are only stored in the compressed E-table; they have the time value “infinity”, e.g. for day 31.12.9999
Performance Tips & Tricks:

- Compress InfoCubes with non-cumulative key figures as soon as possible
- Very granular characteristic value combinations generate lots of reference points in the E table
  - Aggregate rebuild is significantly influenced
  - Delete unused values (e.g. old seasonal material) from the InfoCube by selective deletion without timely restriction
- LAST- and FIRST- key figures can use aggregates; AVG-key figures cannot use them. Try to split these key figures in separate queries, if semantically possible
- At query time use tight restrictions on time characteristics; ideally request only current values
- Suppress sum lines if not needed
- Do not use partial time characteristics (e.g. FISCPER3) if not needed
Validity Objects

- Validity Objects are characteristics for which a validity period can be defined
- The validity is stored within the validity table
- Example: If plant 1 is closed at the end of March, it should not show up any value (even no zero) in April and May

<table>
<thead>
<tr>
<th>Plant</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Reference Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plant 1</td>
<td>+10</td>
<td>+20</td>
<td>+30</td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Plant 2</td>
<td>+20</td>
<td>+10</td>
<td>+10</td>
<td>+20</td>
<td>+30</td>
<td>150</td>
</tr>
</tbody>
</table>

- Every entry in the validity table generates an individual SELECT at query runtime.

✔ Use as few validity objects as possible
✔ Do not misuse validity objects for termination (e.g. insurance contract)
Checklist – Query Design

✓ Multi-dimensional Query Design
✓ Inclusion / Exclusion
✓ MultiProvider Query
✓ Cell Calculation
✓ Customer Exits
✓ Query Read Mode
Query Design – Multi-Dimensional Navigation

Multi-dimensional Navigation

- SAP BW provides very powerful multi-dimensional reporting capabilities like slice & dice, drill-down, drill-through

☑ Every Query should start with a relatively small result set; let the user drill down to more detailed information

☑ Do not use ODS objects for multi-dimensional reporting

Not Recommended as first result
Inclusion/Exclusion

- Selections can include and exclude characteristic values
- Exclusion cannot access database indices

Better use inclusion and avoid exclusion of char values

Recommended
Include in Selection
Exclude from Selection

Not recommended
Swap Interval Limits
Specify Variable Offsets
Select Variable Hierarchy
Query Design – MultiProvider

Query on MultiProvider

Queries on MultiProviders usually access all underlying InfoProviders, even if some cannot be hit as no key figures within the query definition are contained in this InfoProvider.

Use 0INFOPROVIDER to restrict the query to a selection of individual InfoProviders.
Other Query Design Tips

- Cell calculation by means of the cell editor generates separate queries at query runtime
  
  ✔ Be cautious with cell calculations

- Customer-specific code is necessary for virtual key figures and characteristics
  
  ✔ Check Code in Customer Exits

- Read Mode For Queries
  
  ✔ Usually “Read when navigating and expanding hierarchies” is recommended
Different Types of Users in OLAP Reporting

- **Consumer** (~70%) - Low analytical ability, self-explanatory, predefined navigational paths, static reports.
- **Analyst** (~20%) - Moderate analytical ability, multi-dimensional analysis, predefined data collections.
- **Power User** (~8%) - High analytical ability, report creation, ad-hoc reporting.
- **Stand-Alone Query Designer** (~2%) - Only needed for global report creation.

~98% Potential web reporting users

- Report creation
- Ad-hoc reporting
- Multi-dimensional analysis
- Self-explanatory, predefined navigational paths
- Predefined data collections
- Static reports

*Only needed for Global Report Creation*
Checklist – Query and Web Performance – Overview

Which component contributes most?

**Database**
1. Data Model
2. Query Definition
3. Aggregates
4. OLAP Cache
5. Pre-Calculated Web Templates
6. Compressing
7. Indices
8. DB Statistics
9. DB and basis (Buffer) Parameter

**OLAP**
1. Data Model
2. Query Definition (including OLAP features)
3. Aggregates
4. OLAP Cache
5. Virtual Key Figures / Characteristics
6. Authorizations

**Frontend**
1. Query Definition
2. Network
3. WAN and BEx
4. Client Hardware
5. VBA / Java
6. Documents
7. Formatting
8. ODBO / 3rd party

**Tools**
- SQL Trace (ST05)
- RSRV
- RSRT, RSRTRACE
- RSRT, RSRTRACE

© SAP AG 2002, BW Performance Tuning, Alex Peter, 41
Checklist – Query and Web Performance – 1 –

1. Data Model
2. Query Definition
3. Aggregates
4. OLAP Cache
5. Pre-Calculated Web Templates
6. Compressing
7. Indices
8. DB Statistics
9. DB Parameter, AppBuffer

ST03 / Technical Content

Database

Tools
- Query Monitor (RSRT)
- Query Trace (RSRTRACE)
- InfoCube Maintenance
- SQL Trace (ST05)
- Analysis and repair of BW objects (RSRV)
- DB / Buffer Monitors (ST04/ST02)

Check these points
Checklist – Query and Web Performance – 2 –

1. Data Model
2. Query Definition (including OLAP features)
3. Aggregates
4. OLAP Cache
5. Virtual Key Figures / Characteristics
6. Authorizations

Query Monitor (RSRT)
Query Trace (RSRTRACE)
ABAP Trace (SE30)
SQL Trace (ST05)
Authorization Trace (RSSMTRACE)

Tools

Check these points
Checklist – Query and Web Performance – 3 –

1. Query Definition
2. Network
3. WAN and BEx
4. Client Hardware
5. VBA / Java
6. Documents
7. Formatting
8. ODBO / 3rd party

Check these points

Tools

- OS Monitor (ST06)
- IEMON
- RSRTRACE (with “web ping”)
- Frontend system monitor
- IEMON
- Query Monitor (RSRT)
Performance Layers

- For specific Scenarios

Offline Analysis

Pre-Calculation

OLAP Cache

Aggregates

InfoCubes

Performance

Reuse
Aggregates – 1 –

Aggregate Definition
- Materialization of aggregated subsets of InfoCube fact table data
- Independent structures where summary data is stored within separate, transparent InfoCubes
- Transparency: Users do not notice if aggregate is hit or not
- Improved query performance by reducing the amount of data to be read from DB

Aggregates can be created
- Only on top of basic InfoCubes
- For dimension characteristics
- For navigational attributes
- On hierarchy levels
- Using time-dependent navigational attributes (as of BW 3.x)
- Using hierarchy levels where the structure is time-dependent (as of BW 3.x)

Note: Aggregates cannot be built on MultiProviders, SAP Remote Cubes, Remote Cubes or ODS Objects

Aggregates can improve query performance considerably, but keep in mind that they also impact the load performance. For more information, see chapter on data load performance.
### Aggregates – 2 – Example

**Example for flow with and without aggregates**

#### Flow without aggregate

- **Database**
  - Month | Material | Revenue
  - July  | Hammer   | 10
  - July  | Nail     | 20
  - August| Hammer   | 10
  - August| Nail     | 20

- **Selection**
  - Month | Material | Revenue
  - July  | Hammer   | 10
  - July  | Nail     | 20
  - August| Hammer   | 10
  - August| Nail     | 20

- **OLAP Engine**
  - Month | Revenue
  - July  | 30
  - August| 30

#### Flow with aggregate

- **Database**
  - Month | Revenue
  - July  | 30
  - August| 30

- **Selection**
  - Month | Revenue
  - July  | 30
  - August| 30

- **OLAP Engine**
  - Month | Revenue
  - July  | 30
  - August| 30

**Number of records read on the database**

**Records transferred to BW instance after being summarized on the database**
### Aggregates – 3 – Example

- Summarize by characteristic customer
- Abbreviation: customer = space

#### Fact Table: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Internetworks</td>
<td>15</td>
</tr>
<tr>
<td>USA</td>
<td>Funny Duds Inc.</td>
<td>5</td>
</tr>
<tr>
<td>Austria</td>
<td>Internetworks</td>
<td>10</td>
</tr>
<tr>
<td>Austria</td>
<td>Thor Industries</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Funny Duds Inc.</td>
<td>20</td>
</tr>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>25</td>
</tr>
</tbody>
</table>

#### Aggregate Tables: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>40</td>
</tr>
<tr>
<td>Germany</td>
<td>35</td>
</tr>
<tr>
<td>Austria</td>
<td>20</td>
</tr>
</tbody>
</table>
### Aggregates – 4 – Example

- Fixed Value (Germany) for country (subset of data)
- Abbreviation: country = F, Germany

#### Fact Table: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Internetworks</td>
<td>15</td>
</tr>
<tr>
<td>Germany</td>
<td>Funny Duds Inc.</td>
<td>5</td>
</tr>
<tr>
<td>USA</td>
<td>Internetworks</td>
<td>10</td>
</tr>
<tr>
<td>Austria</td>
<td>Thor Industries</td>
<td>10</td>
</tr>
<tr>
<td>Austria</td>
<td>Funny Duds Inc.</td>
<td>20</td>
</tr>
<tr>
<td>Germany</td>
<td>Winsoft Inc.</td>
<td>25</td>
</tr>
</tbody>
</table>

#### Aggregate Tables: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germany</td>
<td>Internetworks</td>
<td>15</td>
</tr>
<tr>
<td>Germany</td>
<td>Funny Duds Inc.</td>
<td>20</td>
</tr>
</tbody>
</table>

© SAP AG 2002, BW Performance Tuning, Alex Peter, 49
Aggregates – 5 – Example

- Summarize by navigational attribute industry
- Abbreviation: customer__industry = *

Navigational Attribute for Characteristic Customer

<table>
<thead>
<tr>
<th>Customer</th>
<th>Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winsoft Inc.</td>
<td>Technology</td>
</tr>
<tr>
<td>Funny Duds Inc.</td>
<td>Consumer Products</td>
</tr>
<tr>
<td>Internetworks</td>
<td>Technology</td>
</tr>
<tr>
<td>Thor Industries</td>
<td>Chemical</td>
</tr>
</tbody>
</table>

Fact Table: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Internetworks</td>
<td>15</td>
</tr>
<tr>
<td>USA</td>
<td>Funny Duds Inc.</td>
<td>5</td>
</tr>
<tr>
<td>Austria</td>
<td>Internetworks</td>
<td>10</td>
</tr>
<tr>
<td>Austria</td>
<td>Thor Industries</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Funny Duds Inc.</td>
<td>20</td>
</tr>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>25</td>
</tr>
</tbody>
</table>

Aggregate Tables: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer__Industry</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>*</td>
</tr>
<tr>
<td>Germany</td>
<td>winsoft Inc.</td>
<td>*</td>
</tr>
<tr>
<td>USA</td>
<td>Internetworks</td>
<td></td>
</tr>
<tr>
<td>Austria</td>
<td>Funny Duds Inc.</td>
<td></td>
</tr>
<tr>
<td>Austria</td>
<td>Internetworks</td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>Thor Industries</td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>Funny Duds Inc.</td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Industry</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>60</td>
</tr>
<tr>
<td>Consumer Products</td>
<td>25</td>
</tr>
<tr>
<td>Chemical</td>
<td>10</td>
</tr>
</tbody>
</table>
Aggregates – 6 – Example

- Summarize on hierarchy level 2 of country hierarchy
- Abbreviation: country = H, Level 2

Hierarchy for Country

Time-independent hierarchies are stored outside the dimension, in a table called /BI0/ICOUNTRY

Fact Table: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Internetworks</td>
<td>15</td>
</tr>
<tr>
<td>USA</td>
<td>Funny Duds Inc.</td>
<td>5</td>
</tr>
<tr>
<td>Austria</td>
<td>Internetworks</td>
<td>10</td>
</tr>
<tr>
<td>Austria</td>
<td>Thor Industries</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Funny Duds Inc.</td>
<td>20</td>
</tr>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>25</td>
</tr>
</tbody>
</table>

Aggregate Tables: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer</th>
<th>H, Level 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>America</td>
<td>Winsoft Inc.</td>
<td>40</td>
</tr>
<tr>
<td>Europe</td>
<td>Internetworks</td>
<td>55</td>
</tr>
</tbody>
</table>

© SAP AG 2002, BW Performance Tuning, Alex Peter, 51
Aggregates – 7 – Tips to Find Aggregates

- Use Technical Content (=BW Statistics) or Workload Monitor (transaction ST03)
  - Based on query statistics (table RSDDSTAT)
Aggregates – 8 – Tips for Creating Aggregates

Do the following steps

- Call query or InfoCube overview in technical content or ST03
- Sort by mean overall time to find queries/InfoCubes with highest runtimes
- Calculate the KPI ‘aggregation ratio’ = number of records read from DB / number of records transferred
- Check quota of database time to total runtime
- As a rule of thumb, an aggregate is reasonable and may be created, if:

  - **Aggregation ratio > 10**, i.e. 10 times more records are read than are displayed, **AND**
  - **Percentage of DB time > 30%**, i.e. the time spent on database is a substantial part of the whole query runtime
Sort by mean overall time to find InfoCubes with queries having the highest runtimes.
How to build Aggregates – 2 –

Database time ~40% of total runtime (6771 of 17 025 seconds)

Ratio records selected / records transferred: 24

Aggregates will probably improve query performance
How to build Aggregates – 3 –

Database time ~3.5% of total runtime (1011 of 28 540 seconds)

Ratio records selected / records transferred: 32

Aggregates will not improve query performance
How to build Aggregates – 4 –

Database time ~90\% of total runtime (1420 of 1572 seconds)

Ratio records selected / records transferred: 1.1

Aggregates will not improve query performance
How to build Aggregates – 5 – (ST03)

Highest contributor in Total Run Time per InfoCube

High % DB Time

Expert Mode!!

High ratio: Selected / transferred records
How to build Aggregates – 6 –

Utilizing OLAP per Query

Sort by overall time to find queries that have the highest total runtime
How to build Aggregates – 7 –

Utilizing OLAP per Query

Database time ~60% of total runtime (2.798 of 4.685 seconds)

Ratio records selected / records transferred: 54

Aggregates will improve query performance
How to build Aggregates – 8 –

Database time ~89% of total runtime (1064 of 1162 seconds)

Ratio records selected / records transferred: 2.1

Aggregates will not improve query performance
PARALLEL Query on MultiCubes in BW 2.x

Queries on MultiCubes are split up over the different BasisCubes with parallel access to the InfoCube or maximal one aggregate per Cube

Query on MultiCube => 40 sec runtime

Sub-Query I

Sub-Query II

Sub-Query III

BasisCube I

15 sec

BasisCube II

40 sec

Aggregate BasisCube III

2 sec
Using the NOPARALLEL–option in table RSADMIN:
the BasisCubes are accessed in serial,
but more than one aggregate per Infocube can be used

Query on MultiCube

Sub-Query I

Sub-Query II

Sub-Query III

1 sec

3 sec

2 sec

15 sec

=>23 sec runtime
Query on MultiProvider in BW 3.X

Query on MultiProvider

Sub-Query I

Aggregate I

BasisCube I

1 sec

Sub-Query II

Aggregate II

BasisCube II

3 sec

Sub-Query III

Aggregate III

BasisCube II

2 sec

Aggregate BasisCube III

2 sec

15 sec runtime

© SAP AG 2002, BW Performance Tuning, Alex Peter, 64
Aggregates Features

Further Performance Features in BW 3.x

- Block Size
  - Aggregate Build for large InfoCubes are resource-consuming (e.g. temporary tablespace, memory)
  - Data of large InfoCubes can be read in several blocks to prevent resource problems when filling an aggregate.
  - The block size can be customized (system-wide) – optimum depends on system sizing and on underlying DBMS
  - Blocks are internally distinguished by characteristic values

- Flat Aggregates
  - If an aggregate has less than 15 components, each component is put into a separate dimension ("Flat Aggregates")
  - The dimensions (except the package and unit) are marked as “Line Item”
  - Flat Aggregates are filled / rolled up without loading the data into the application server (performance gain)
  - Line item / high cardinality dimensions already specified in the InfoCube are also used for aggregates for corresponding dimensions (also for non-flat aggregates)
OLAP Cache – 1 –

- Caching in BW 2.x
  - One Local Cache for each session
  - No cache access across sessions
  - No possibility to set size of cache

- Caching in BW 3.x → OLAP Cache
  - Additional global cache which is accessible from all sessions
  - OLAP Cache is part of application buffer (Imp/Exp SHM) and can also be stored in a DB table or a file
  - The Cache stores query results and navigation statuses as highly compressed cluster data
  - Query Cache is used for equal queries or subsets of cached queries
  - Benefits: OLAP Cache reduces workload on database and application server
  - Invalidations:
    - Data Load into underlying DataTarget
    - Query Generation
OLAP Cache

- OLAP Cache size can be adjusted
- (In 3.x there is also an adjustable size for the local cache)
- Cache can be switched off entirely, for InfoCubes and for individual queries
- Local Cache is used, when OLAP Cache is inactive.
- Specific instance profile parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsdb/esm/buffersize_kb</td>
<td>4096 (kB)</td>
<td>Size of exp/imp SHM buffer</td>
</tr>
<tr>
<td>rsdb/esm/max_objects</td>
<td>2000</td>
<td>Max. number of objects in the buffer</td>
</tr>
<tr>
<td>rsdb/esm/large_object_size</td>
<td>8192 (byte)</td>
<td>Estimation for the size of the largest object</td>
</tr>
<tr>
<td>rsdb/esm/mutex_n</td>
<td>0</td>
<td>Number of mutexes in Exp/Imp SHM buffer</td>
</tr>
</tbody>
</table>

Buffer small by default!
OLAP Cache – 3 –

**OLAP Cache Usage**

- **Front-end**
  - Query Definition (BEx)
  - Analyzer Display and Manipulation

- **Memory**
  - OLAP Processor
  - Query Execution
  - 1st

- **Application Server**
  - Data Manager

- **Database Server**
  - 2nd
  - 3rd
  - InfoCube
  - Aggregates

- **Current Query View**
OLAP Cache – 4 –

- OLAP Cache Monitor
  - Accessible via RSRT
  - Check status of cache
  - Delete cache entries

Delete Cache
Pre-Calculated Web Templates

Pre-calculation is a set of techniques where you can distribute the workload of running the report to off-peak hours, and have the report result set ready for very fast access to the data.

Data Pre-Calculation or HTML Pre-Calculation
- HTML Pre-Calculation can be used for Offline Analysis

Benefits
- Fast response time
- System workload shifted to off-peak hours
- Re-use data that goes to many queries
- Reports also available offline
Scenario

- Supports “Newspaper” scenario
  - Quick access to many reports
  - Reports are requested by many users
  - Static reporting, usually little navigation requirements
  - Active Reporting Authorisations

- Restrictions
  - No invalidation when new data is loaded
  - Restricted navigation: only filtering possibles
Pre-Calculated Web Templates – 3 – Access Modes

**DATA MODE**

- **NEW**
- **STORED**
- **HYBRID**
- **STATIC**
- **STATIC_HYBRID**

**Web Browser**

http://myserver:myport/sap/bw/BEx?cmd=ldoc&template_id=myTemplate&data_mode=new

**BW Server**

- **BEx Web Service**
- **Doc Web Service**

**WEB FRAMEWORK**

- **Web Framework**

**Reporting Agent**

**OLAP Engine**

**Log**

**Web Application Server**

- **WebDAV Handler**
- **Content Management Framework**

**InfoProvider**

**RA Data Storage**

**CMF Repository**
Pre-Calculated Web Templates – 4 – Access Modes

**DATA MODE**

- NEW
- STORED
- HYBRID
- STATIC
- STATIC_HYBRID

```
http://myserver:myport/sap/bw/BEx?cmd=ldoc&template_id=myTemplate&data_mode=stored
```
Pre-Calculated Web Templates – 5 – Access Modes

**DATA MODE**

- NEW
- STORED
- HYBRID
- STATIC
- STATIC_HYBRID

**Web Browser**

http://myserver:myport/sap/bw/BEx?cmd=ldoc&template_id=myTemplate&data_mode=static


**Web Application Server**

- BEx Web Service
- Doc Web Service
- BW Server
- Web Framework
- OLAP Engine
- Reporting Agent
- Log
- WebDAV Handler
- Content Management Framework

**InfoProvider**

**RA Data Storage**

**CMF Repository**
Indices

- If you report on ODS objects with a restrictive selection key, check if indices are defined for this key.
- Check if the InfoCube Indices exist.
  - In ORACLE, you can choose between bitmap index and B-tree index.
  - Use B-tree index, if dimension size exceeds 10% of the fact table size.
- If you select on navigational attributes, be sure that an appropriate index is available.
Reporting Performance – ODS Objects

Indexing ODS Objects

- Indices can be defined in the 3.x ODS object maintenance; in BW 2.x indices must be defined via SE11

✓ Use indices on characteristics which you are accessing regularly (for reporting or DataMart interface)

Example: If you look for sales figures with customer Birch, a full table scan is necessary when there is no index (see picture above); if the index has been created, only two DB accesses are necessary (see picture on the left).
Index types (selection)

- **Bitmap Index**
  - Well-suited for few characteristic values
  - Several bitmap indexes can be used for one table (bitmap merge join)

  - Miller: 00110000111
  - Smith: 10000011100
  - Trapp: 01001100000

- **B-Tree Index**
  - Good for range queries and for high cardinality

```
Miller 01001100000
Smith 10000011100
Miller 00110000111
```

![Diagram of B-Tree Indexes]

© SAP AG 2002, BW Performance Tuning, Alex Peter, 77
Indexing in BW

Fact Table

- **Non-Unique Index on every dimension ID**
  - ORACLE: Bitmap (except for “high cardinality” dimensions)
  - ORACLE: B-tree on F-fact table for transactional InfoCubes (guarantees parallel read&write)
  - ORACLE: additional bitmap index on time characteristic SID for partitioned InfoCubes
  - DB2/AS400: EVI
  - Other DBMSs: B-Tree
  - DB2/OS390: B-Tree not on package dimension ID

- **Non-unique B-Tree on all dimension IDs (Primary Key)**
  - Used for compression
  - ORACLE/Informix: only on E-fact table
Indexing in BW

Dimensions
- Dimension ID (primary key)
  - Unique B-Tree
- Master Data SIDs
  - Non-Unique B-Tree
  - ORACLE / DB2/AS400: not on first SID (this is included in the index on all SIDs)
- On all master data SIDs
  - Non-Unique B-Tree

Master Data
- X- and Y- tables (SIDs for all navigational attributes)
  - Unique B-Tree on <SID, Objectversion> (primary key)
  - Additional indices could be defined for individual SID-columns to avoid full table scans (for selective accesses)
- S-table (SID)
  - Unique B-Tree on master data key (primary key)
  - Unique B-Tree on SID
Cost-Based Optimizer (CBO)

- Calculation of the optimal execution plan
  - E.g. full table scan vs. index
  - E.g. Index range scans vs. unique scan (only one result)
  - E.g. nested loop vs. hash join (vs. sort merge join)
- Based on table and index statistics
  - Number of records
  - Number of different keys
- Based on histograms (ORACLE 9i)
  - Detailed information about data distributions which are non-uniform
- Recalculate statistics when there is a major change in the table structure (e.g. data load)
InfoCube – Compression – 1 –

Double Fact Table

- “F” Table
  - Request Information
  - Usually small
  - Optimised for Loading and Deleting

- “E” Table
  - Optimised for Queries
  - Usually large
  - User-defined DB Partitioning

As InfoPackages are added, F fact table partitions are created

Upload

Compression

F - Table

REQUEST No. | Time | Material | Sales

Fact Table

E - Table

REQUEST No. | Time | Material | Sales
Compression

We recommend to compress as soon as possible

- Compression usually reduces the number of records by combining records with the same key that has been loaded in separate requests
- The user-defined DB partitioning (depending on the DBMS) is only affecting the compressed E-table

Advantage
  - Query performance is usually improved significantly

Disadvantage
  - Note that you lose the request information after the compression
Aggregates and Compression

Request Handling within BW 3.x Aggregates

- In BW 3.x, InfoCubes can be marked such that the request is kept in its aggregates.
- Requests which are not compressed can then be deleted out of the InfoCube and its aggregates, without completely rebuilding the aggregates.
- Aggregates of marked InfoCubes are compressed together with the InfoCube.

✔ Compress aggregates as soon as possible to avoid query performance penalty.
Reporting Authorizations

- Reporting Authorizations restrict critical data to certain user/user groups
- Different authorization levels:
  - InfoCube
  - Query
  - Key figures
  - Characteristic values
  - Hierarchies
- Complex authorization checks on large hierarchies including several hierarchy levels can slow down query performance
Web Applications vs. BEx Analyzer

- Advantages of Web Applications
  - HTTP Compression, MIME Compression
  - Browser Caching of images
  - Significantly less roundtrips than BEx Analyzer

Browser-based web applications, using http, represents a significant gain in query response time – especially in a WAN!
Documents

- Transferring large documents (e.g. as document items) can cause significant network traffic

Formatting

- Many Different formatting definition (especially in BEx Analyzer) can impact the frontend performance of queries displaying many lines
- If this is considered to be a problem, try to reduce sum lines (as they require a new format definition transfer) or switch off formatting completely
Performance Guidelines

- General Settings and Checklist
- Modeling
- Query Performance
- Extraction / Data Load Performance
- Analysis Methods
Checklist – Data Load Performance – Overview 1 –

Which component contributes most?

- Extraction
- Transfer
- Load Into PSA

Check these points:

1. Customer Exits
2. Resource Utilization
3. Load Balancing
4. Data Package Size
5. Indices on tables
6. Flat File format
7. Content vs. generic extractor

Tools:

- Extractor Checker (RSA3), ABAP Trace (SE30), SQL Trace (ST05)
- SM50
- SQL Trace (ST05)
- OS Monitor (ST06)
- DB Monitor (ST04)

- 1. Resource Constraint
- 2. CPU / Memory Bottleneck
- 3. Network
- 4. Application Buffer Synchronization

- 1. I/O Contention
Checklist – Data Load Performance – Overview 2 –

Which component contributes most?

- Transfer Rules
  - 1. Transformation Rules / ABAP Coding
  - 2. Transformation Library Formulas
- Update Rules
- Load Into Data Targets
  - 1. Buffering Number Ranges
  - 2. Change Run
  - 3. Compression
  - 4. Indices
  - 5. Load Master Data before Transaction Data
  - 6. Buffering Number Ranges
- Master Data
- InfoCubes
- ODS Objects
  - 1. Parallel ODS activation
  - 2. Unique Data Records
  - 3. Flag BEx Reporting
  - 4. Indices

Check these points:

- Debugger within Monitor
- ABAP Trace (SE30)
- SQL Trace (ST05)

Tools

© SAP AG 2002, BW Performance Tuning, Alex Peter, 89
Typical BW Data Load Process

1. **Source System**
   - Business Information Warehouse
   - ALE
   - IDOC
   - PSA
   - Update Rules
   - Transfer Rules
   - ODS

2. **Loading Process**
   - Scheduler
   - InfoCube

3. **Extractor**
   - BW S-API
   - ALE
   - IDOC
   - tRFC

**Business Information Warehouse**

---

© SAP AG 2002, BW Performance Tuning, Alex Peter, 90

**THE BEST-RUN BUSINESSES RUN SAP**
### Performance Guidelines – Extraction / Staging –

#### Data Load Monitor: Transaction RSMO

**Extraction**

<table>
<thead>
<tr>
<th>Data Load Monitor: Transaction RSMO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Header</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Details</strong></td>
</tr>
</tbody>
</table>

- **Requests (messages): Everything OK**
  - Data request arranged
  - Confirmed with: OK

- **Extraction (messages): Everything OK**
  - Data request received
  - Data selection scheduled
  - 90 Records sent (90 Records received)
  - Data selection ended

- **Transfer (Idocs and TRFC): Everything OK**
  - Request IDoc: Application document posted
  - Info IDoc 2: Application document posted
  - Info IDoc 1: Application document posted
  - Info IDoc 3: Application document posted
  - Data Packet 1: 90 Records Arrived in BW
  - Info IDoc 4: Application document posted

- **Processing (data packet): Everything OK**
  - Data Packet 1 (90 Records): Everything OK

- **Transfer rules: No errors**
  - Transaction data received, Processing being run.
  - Transfer 90 data records in communication status.

- **Update PSA: No errors**
  - Update rules: No errors

- **Update rules: No errors**
  - Update: No errors

- **Processing and runtime of the process**

Select a node and choose the context menu for more information.
Checklist – Data Load Performance – Extraction –

Technical Content, Data Load Monitor

Extraction

1. Customer Exits
2. Resource Utilization
3. Load Balancing
4. Data Package Size
5. Indices on tables
6. Flat File format
7. Content vs. generic extractor

Extractor Checker (RSA3), ABAP Trace (SE30), SQL Trace (ST05)

Process Overview (SM50 / SM51)

SQL Trace (ST05)

Check these points

Tools

© SAP AG 2002, BW Performance Tuning, Alex Peter, 92
Data Extraction Performance – 1 –

- **Customer Exits**
  - Take Recommendations of performance ABAP coding into consideration; see later

- **Resource Utilization**
  - Look for resource competing processes in SM50/SM51

- **Load Balancing**
  - Distribute to different servers to avoid CPU / Memory Bottlenecks on one server and enable greater throughput
  - Configure ROIDOCPRMS
  - Load Balancing can be defined:
    - SAP source system – BW – Connection in SM59
    - InfoPackages, Process Chains
    - XML Data Loads: HTTP/HTTPS processes can be allocated to specific server groups
Data Extraction Performance – 2 –

Data Package Size

- Data Package size defines the number of DB commits
  - In resource constrained systems: small package size
  - In large systems: increase size to speed up collection
- Set ROIDOCPRMS (which can be overwritten by InfoPackage) according to recommendations in OSS note 417307

Indices on Source Tables

- If you decrease the size of data to be extracted by selection criteria in the InfoPackage, consider building indices on the source tables to avoid full table scans
Data Extraction Performance – 3 –

- **Flat File Format**
  - CSV must be internally converted into fixed length format
  - Use fixed-length ASCII format if you load huge volumes of data via flat files
  - Moreover, place the file on the application server rather than on the client; avoid large loads across slow networks, from tape or from I/O critical disks

- **Content vs. Generic Extractor**
  - Most Content extractors support important features like delta capability
  - If a Content extractor meets your business requirements, do not build a generic extractor
  - For logistics, use V3 extractors rather than the old LIS extractors
Use parallel processes

- Data is automatically packaged in the source system; so extraction is parallel
- Load into PSA and data targets in parallel
- Start several InfoPackages
  - Use selection criteria to split data
  - Split flat files
- Use 3.x-Process Chains (or Event Chains or InfoPackageGroups)
Performance Guidelines – Transfer –

Data Load Monitor: Transaction RSMO
Checklist – Data Load Performance – Transfer –

Technical Content, Data Load Monitor

Transfer

1. Resource Constraint
2. CPU / Memory Bottleneck
3. Network
4. Application Buffer Synchronization

Check these points

Tools

SM50
OS Monitor (ST06)
SQL Trace (ST05)
Data Transfer – 1 –

- **SM50 - Processes**
  - With upload method “load PSA and Data Targets in Parallel”, the number of processes in BW is double as much as the extractor jobs in the OLTP system

![Diagram showing data packet transfer between BW and OLTP systems with 6 Dialog WPs on the BW side and 3 Dialog WPs on the OLTP side.]
Resource Constraints

- Many parallel processes can compete for resources
- CPU / Memory / Network resources
  - Check ST06

Application Buffer Synchronization

- During large data loads, swapping of buffers (where e.g. master data resides) can occur on a BW application server
- Synchronization overhead of buffers for several application servers (check SQL trace for log table activities)
- If you run into these sync problems, shut down 2nd application server and set rdisp/bufrefmode = “sendoff, exeauto” during the load phase
Performance Guidelines – Load Into PSA –

Data Load Monitor: Transaction RSMO
Checklist – Data Load Performance – PSA Load –

Technical Content, Data Load Monitor

Load into PSA

1. I/O Contention

OS Monitor (ST06)

Check these points

Tools
Load Into PSA

■ I / O - Contention
  ◆ High number of DB writes during large data loads
  ◆ Disk Layout and Striping
    ● What is located on the same disk or tablespace/DB space etc.?

■ DB Statistics
  ◆ DB statistics are usually not an issue, as PSA tables/partitions are normally read sequentially

■ Load Method
  ◆ If you select sequential loading from PSA to DataTargets, the total runtime of the load process is composed by the (parallel) loads of requests to PSA and the subsequent loading to DataTargets one by one
Performance Guidelines – Transfer Rules –

Data load Monitor: Transaction RSMO

Transfer Rules
Performance Guidelines – Update Rules –

Data load Monitor: Transaction RSMO
Checklist – Data Load Performance – Transformation –

1. Transformation Rules / ABAP Coding
2. Transformation Library Formulas

Check these points

Tools

Transformation Routine Debugger,
ABAP Trace (SE30),
SQL Trace (ST05)

ABAP Trace (SE30)
Transformation Rules – 1 –

- **Identify expensive transfer/update rules**
  - Use ABAP trace (SE30) or SQL trace (ST05) to pinpoint sub-optimal code and/or expensive DB statements
  - Use Debugging facilities within the Data Load Monitor

- Simple tool for debugging of transfer or update rules
- Improves error search and analysis – together with the enhanced error messages
- Debug from one update rule to the next one for each InfoObject
Transformation Rules – 2 –

- **Transformation Rules / ABAP Coding**
  - Check recommendations at the end of the workshop (e.g. start routine)

- **Transformation Rules / Transformation Library**
  - BW 3.x provides powerful transformation rules that can be applied without any ABAP knowledge
    - Library transformations are interpreted at runtime
    - If you notice interpreter overhead in transaction SE30, switch complicated rules to ABAP coding again
Performance Guidelines – Data Targets –

- **Source System**
  - Business Information Warehouse
  - ALE
  - IDOC
  - ALE
  - IDOC
  - ALE

- **Loading Process**
  - Extractor
  - PSI
  - S-API
  - Scheduler

- **Data Load Monitor - RSMO**
  - ODS
  - InfoCube
  - Update rules
  - Transfer rules
  - PSA
  - tRFC

- **Load Into Data Targets**
Checklist – Data Load Performance – Master Data –

1. Buffering Number Ranges
2. Change Run

Tools

- SQL Trace (ST05)
- Aggregate Hierarchy
Loading Master Data – 1 –

- **Number Ranges**
  - SID number range can be buffered instead of accessing the DB for each SID
  - If you encounter massive accesses to DB table NRIV via SQL trace (ST05), increase the number range buffer in transaction SNRO
  - If possible, reset the value to its original state after the load (to avoid unnecessary memory allocation)

<table>
<thead>
<tr>
<th>SID</th>
<th>Material Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4711</td>
</tr>
<tr>
<td>2</td>
<td>0815</td>
</tr>
<tr>
<td>3</td>
<td>4712</td>
</tr>
</tbody>
</table>

New material: 1125

Draw new SID from Buffer
Change Run

- When master data changes, the changes of the navigational attributes/hierarchies must be applied to the depending aggregates; this process is called change run.

- Newly loaded master data is not active until the change run has been applied the changes to all aggregates.

- See more information in the “Load into InfoCube” chapter.
Checklist – Data Load Performance – InfoCubes –

1. Roll-up
2. Change Run
3. Compression
4. Indices
5. Load Master Data before Transaction Data
6. Buffering Number Ranges

Technical Content, Data Load Monitor

Data Targets

InfoCubes

Check these points

Aggregate Hierarchy

SQL Trace (ST05)
InfoCube Maintenance

SQL Trace (ST05)

Tools
Roll-up

- The roll-up process populates all aggregates of an InfoCube with the newly loaded delta load
- Basic Rule: InfoCube and all depending aggregates must be in-sync, i.e. you must see the same data no matter if it has been derived from the InfoCube or the aggregate
- Newly loaded data is not available for reporting until it has been rolled up into the aggregates

Aggregate Hierarchy

- Aggregates can be built out of other aggregates to reduce the amount of data to be read and, hence, to improve the roll-up performance
- Aggregate hierarchy is determined automatically
- General Guideline: define few basis (large) aggregates and many small aggregates that can be built from the hierarchy level before

Display aggregate hierarchy
Example: Optimized Aggregate Hierarchy

Basic InfoCube

Few large basic aggregates

Many small aggregates

Example

\{Material, Material Group, Customer, Day\}

Example

\{Material Group, Customer, Day\}
Why Can Too Many Aggregates Be Harmful?

Balanced aggregates strategy

Time for Rollup and Change Run

Faster Queries
Aggregate Design

Tips for Maintaining Good Aggregates

✓ Relatively small compared to parent InfoCube
✓ Try for summarization ratios of 10 or higher
✓ Find good subsets of data (frequently accessed)
✓ Build on some hierarchy levels, not all
✓ Not too specific, not too general – should serve many different query navigations
✓ Should be frequently used and used recently (except basis aggregates)

Avoid Aggregates

○ That are too big (compared to the parent InfoCube/Aggregate)
○ That are similar to too many other aggregates
○ That are old and that are not frequently used
○ That are built for a specific query (rather build general ones)
Change Run

- Aggregates can contain
  - Dimension Characteristics
  - Navigational Attributes
  - Hierarchy Levels

- When master data changes, the changes of the navigational attributes/hierarchies must be applied to the depending aggregates; this process is called change run.

- Newly loaded master data is not active until the change run has been applied the changes to all aggregates.

- Threshold for delta and new build-up in customizing.

- The Change Run can be parallelized across InfoCubes; see SAP note 534630 for more details.

- Check aggregate hierarchy (see Roll-Up for more details).

- Try to build basis aggregates that are not affected by the change run, i.e. no navigational attributes nor hierarchy levels.

- The following slides show details on the process itself …
# Change Run – 2 – Before Master Data Activation

## Fact Table: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Internetworks</td>
<td>15</td>
</tr>
<tr>
<td>USA</td>
<td>Funny Duds Inc.</td>
<td>5</td>
</tr>
<tr>
<td>Austria</td>
<td>Internetworks</td>
<td>10</td>
</tr>
<tr>
<td>Austria</td>
<td>Thor Industries</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Funny Duds Inc.</td>
<td>20</td>
</tr>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>25</td>
</tr>
</tbody>
</table>

## Aggregate Tables: Sales Data

<table>
<thead>
<tr>
<th>Industry</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>60</td>
</tr>
<tr>
<td>Consumer Products</td>
<td>25</td>
</tr>
<tr>
<td>Chemical</td>
<td>10</td>
</tr>
</tbody>
</table>

**Navigational Attribute for Characteristic Customer**

Customer: Winsoft Inc., Funny Duds Inc., Internetworks, Thor Industries

Industry: Technology, Consumer Products, Chemical

**Changed master data not available for reporting**

Old:
- Winsoft Inc.
- Funny Duds Inc.
- Internetworks
- Thor Industries

New:
- Internetworks
Change Run – 3 – After Master Data Activation

Navigational Attribute for Characteristic Customer

Customer | Industry
---|---
Winsoft Inc. | Technology
Funny Duds Inc. | Consumer Products
Internetworks | Technology
Thor Industries | Chemical

Fact Table: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Customer</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Internetworks</td>
<td>15</td>
</tr>
<tr>
<td>USA</td>
<td>Funny Duds Inc.</td>
<td>5</td>
</tr>
<tr>
<td>Austria</td>
<td>Internetworks</td>
<td>10</td>
</tr>
<tr>
<td>Austria</td>
<td>Thor Industries</td>
<td>10</td>
</tr>
<tr>
<td>Germany</td>
<td>Funny Duds Inc.</td>
<td>20</td>
</tr>
<tr>
<td>USA</td>
<td>Winsoft Inc.</td>
<td>25</td>
</tr>
</tbody>
</table>

Aggregate Tables: Sales Data

<table>
<thead>
<tr>
<th>Country</th>
<th>Space</th>
<th>Industry</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td></td>
<td>Technology</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer Products</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chemical</td>
<td>10</td>
</tr>
</tbody>
</table>
Indices

- Indices are necessary for reporting as they enable fast retrieving of records from DB tables.
- Maintaining index build during load time is expensive, though.
- If the (uncompressed) F fact table is small, it is usually faster to drop the secondary (bitmap) indices before the load and build them up after the load.
- This feature is available in InfoCube maintenance and as Process Type.
Data Load Performance Into InfoCubes – Others

- **Load Master Data BEFORE transaction data**
  - Master Data Load generates SIDs and populates master data tables
  - Avoids expensive operations (SID creation) during transaction data

- **Number Ranges**
  - Dimension ID number range can be buffered instead of accessing the DB for each Dimension ID
  - If you encounter massive accesses to DB table NRIV via SQL trace (ST05), increase the number range buffer
  - If possible, reset the value to its original state after the load (to avoid unnecessary memory allocation)
Checklist – Data Load Performance – ODS Objects –

1. Parallel ODS activation
2. Unique Data Records
3. Flag BEx Reporting
4. Indices

Tools

- Process Monitor (SM50 / SM51)
- ODS Object Maintenance
- SQL Trace (ST05)

Check these points
ODS Objects

- In BW 3.x, data can be loaded and activated in parallel

**Use Parallel Upload and Activation in BW 3.x**

**IMG (or RSCUSTA2)**

- Change View "Maintenance View for ODS Fields (RSADM..."

- Customizing ID: Business Information Warehouse

- Maintenance View for ODS Fields (RSADMINA)
  - No. of Par. Proc.: 1
  - Min. No. Data Recs.: 10000
  - Wait Time in Sec.
  - Server Group

- Load Balancing to Server Groups

© SAP AG 2002, BW Performance Tuning, Alex Peter, 124
Data Packets / Requests cannot be loaded into an ODS object in parallel:
- Overwriting functionality
- Locking on Activation table

Sequential load

Staging Engine
ODS Objects – Data Activation in BW 3.x – 3 –

New queuing mechanism replacing previous Maintenance (M)table

New/modified data

Req1

Req2

Req3

Activation

Staging Engine

Parallel load
ODS Objects – Data Activation in BW 3.x – 4 –

Upload to Activation queue
- Data from different requests are uploaded in parallel to the activation queue.

Activation
- During activation the data is sorted by the logical key of active data plus change log key.
- This guarantees the correct sequence of the records and allows inserts instead of table locks.

Before- and After Image
- Request ID in activation queue and change log differ from each other.
- After update, the data in the activation queue is deleted.
ODS Objects – Data Load

Loading Unique Keys

- The “Unique”-Flag accelerates data activation significantly
  - It directly inserts data to the active data table instead of checking if the key is already there (mass inserts)
  - No sorting
  - No Before Image in change log

- Only usable for documents that cannot be changed, e.g. sales documents

√ Set the “Unique”-Flag if you are sure that all your records are unique
ODS Objects – Reporting

InfoSet on top of ODS Object

- The BEx Reporting flag forces the system to draw SIDs for the master data; although they are determined in parallel, this slows down activation performance.

In BW 3.x, SIDs are drawn during activation, in BW 2.x during data load.

✔ Do not use “BEx Reporting” flag, if not necessary

- InfoSets in BW 3.x provide also fast selective access to individual records, if indices are available

✔ Try to use InfoSet on top of ODS objects instead of BEx-enabled ODS objects

---

Change View to: 5084

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SIC Hierarchy</td>
<td>Revenue</td>
<td>320,674</td>
<td>310,150</td>
<td>3.4%</td>
<td>2,162,278</td>
<td>2,081,074</td>
</tr>
<tr>
<td></td>
<td>Involved Quantity</td>
<td>320,674</td>
<td>310,150</td>
<td>3.4%</td>
<td>2,162,278</td>
<td>2,081,074</td>
</tr>
<tr>
<td>Agriculture, forestry and fishing</td>
<td>Revenue</td>
<td>3,594</td>
<td>6,630</td>
<td>50.0%</td>
<td>66,016</td>
<td>66,016</td>
</tr>
<tr>
<td></td>
<td>Involved Quantity</td>
<td>3,594</td>
<td>6,630</td>
<td>50.0%</td>
<td>66,016</td>
<td>66,016</td>
</tr>
<tr>
<td>Mining</td>
<td>Revenue</td>
<td>4,953</td>
<td>4,537</td>
<td>8.0%</td>
<td>4,953</td>
<td>4,537</td>
</tr>
<tr>
<td></td>
<td>Involved Quantity</td>
<td>4,953</td>
<td>4,537</td>
<td>8.0%</td>
<td>4,953</td>
<td>4,537</td>
</tr>
</tbody>
</table>

Reporting possible via InfoSet

---

© SAP AG 2002, BW Performance Tuning, Alex Peter, 129
Analysis and Repair of BW Objects (RSRV) – 1 –

Analyze DB parameters and specific InfoCubes

- DB Parameters
- Indices & DB Statistics of InfoCubes (and Aggregates)
### Analyze DB parameters and specific InfoCubes

- Database information about the star schema tables of an InfoCube
- Check dimension sizes in relationship to the fact table size

![Image of SAP BW interface showing analysis and repair of BW objects](image)

<table>
<thead>
<tr>
<th>Table</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD_T005_SD_SD_SD001</td>
<td>10 entries, 10% of the InfoCube</td>
</tr>
<tr>
<td>BD_T005_SD_SD_SD002</td>
<td>20 entries, 20% of the InfoCube</td>
</tr>
<tr>
<td>BD_T005_SD_SD_SD003</td>
<td>30 entries, 30% of the InfoCube</td>
</tr>
<tr>
<td>BD_T005_SD_SD_SD004</td>
<td>40 entries, 40% of the InfoCube</td>
</tr>
</tbody>
</table>
Query Monitor (RSRT) – 1 –

Analyze specific queries

Debug Option
### Query Monitor (RSRT) – 2 –

**Analyze specific queries**
- Check if the right aggregate(s) is(are) used by the query

#### Structure to be read

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100005</td>
<td>0CALMONTH</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>100005</td>
<td>0D_MATERIAL</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>100005</td>
<td>0D_PLANT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>100003</td>
<td>0CALMONTH</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>100003</td>
<td>0D_MATERIAL</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>100003</td>
<td>0D_PLANT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>100003</td>
<td>0D_VENDOR</td>
<td>F</td>
<td>0</td>
<td>10</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Available Aggregates

- *: disaggregated
- space: aggregated
- F: fixed value
- H: hierarchy level aggregation

---

© SAP AG 2002, BW Performance Tuning, Alex Peter, 134
Query Trace (RSRTRACE)

Analyze specific queries
- Check all drill-downs and navigation steps

Configure trace tool

Record Trace
Replay trace with transaction RSRCATTTRACE
Analyze high database runtime

- Trace specific process with SQL Trace

- Activate SQL Trace Flag
  - Activate Trace
  - Execute process
  - Deactivate Trace
  - Analyze Trace
### Performance Analysis: SQL Trace (ST05) – 2 –

#### Summarized SQL Statements: Sorted by duration

<table>
<thead>
<tr>
<th>Executions</th>
<th>Identical</th>
<th>Duration</th>
<th>Records</th>
<th>Time/exec</th>
<th>Rec/exec.</th>
<th>AvgTime/R</th>
<th>MinTime/R</th>
<th>Length</th>
<th>BfTp</th>
<th>TabType</th>
<th>Obj. name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>395,897</td>
<td>42,759</td>
<td>790,097</td>
<td>42,759.0</td>
<td>18</td>
<td>18</td>
<td>152</td>
<td>ful</td>
<td>TRANS</td>
<td>RSOLTPSOURCET</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>349,381</td>
<td>3,135</td>
<td>449,301</td>
<td>6,135.0</td>
<td>35</td>
<td>55</td>
<td>350</td>
<td>ful</td>
<td>TRANS</td>
<td>RSTS</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>346,227</td>
<td>24,119</td>
<td>407,227</td>
<td>24,118.0</td>
<td>17</td>
<td>17</td>
<td>152</td>
<td>ful</td>
<td>TRANS</td>
<td>RSIST</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>314,856</td>
<td>4,033</td>
<td>314,860</td>
<td>4,033.0</td>
<td>78</td>
<td>78</td>
<td>235</td>
<td>cust</td>
<td>TRANS</td>
<td>RSUPDINFO</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>255,531</td>
<td>15,649</td>
<td>255,551</td>
<td>16,649.0</td>
<td>15</td>
<td>15</td>
<td>112</td>
<td>ful</td>
<td>TRANS</td>
<td>RSDAREAT</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>156,692</td>
<td>5,267</td>
<td>156,602</td>
<td>5,257.0</td>
<td>38</td>
<td>38</td>
<td>182</td>
<td>ful</td>
<td>TRANS</td>
<td>RSOLTPSOURCET</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>39,732</td>
<td>2,557</td>
<td>39,732</td>
<td>2,557.0</td>
<td>10</td>
<td>10</td>
<td>236</td>
<td>ful</td>
<td>TRANS</td>
<td>RSLDP10</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>37,486</td>
<td>785</td>
<td>37,486</td>
<td>785.0</td>
<td>48</td>
<td>48</td>
<td>112</td>
<td>ful</td>
<td>TRANS</td>
<td>RSQISETT</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>34,781</td>
<td>300</td>
<td>34,781</td>
<td>300.0</td>
<td>116</td>
<td>116</td>
<td>358</td>
<td>sng</td>
<td>TRANS</td>
<td>RSIGSTAT</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>20,026</td>
<td>291</td>
<td>20,026</td>
<td>291.0</td>
<td>91</td>
<td>91</td>
<td>230</td>
<td>ful</td>
<td>TRANS</td>
<td>PAT03</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>11,207</td>
<td>1</td>
<td>11,207</td>
<td>1.0</td>
<td>11,207</td>
<td>11,207</td>
<td>41</td>
<td>sng</td>
<td>TRANS</td>
<td>TSTCC</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>4,891</td>
<td>58</td>
<td>2,446</td>
<td>29.0</td>
<td>84</td>
<td>77</td>
<td>34</td>
<td>ful</td>
<td>TRANS</td>
<td>RSQISETLOC</td>
</tr>
<tr>
<td>4</td>
<td>75</td>
<td>4,458</td>
<td>4</td>
<td>1,117</td>
<td>1.0</td>
<td>1,117</td>
<td>908</td>
<td>915</td>
<td>ful</td>
<td>TRANS</td>
<td>RSSTART</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4,292</td>
<td>133</td>
<td>4,292</td>
<td>133.0</td>
<td>32</td>
<td>32</td>
<td>350</td>
<td>cust</td>
<td>TRANS</td>
<td>RSDDA66F0IR</td>
</tr>
<tr>
<td>3</td>
<td>67</td>
<td>3,535</td>
<td>0</td>
<td>1,178</td>
<td>0.0</td>
<td>3,535</td>
<td>1,278</td>
<td>12</td>
<td>ful</td>
<td>TRANS</td>
<td>RSEND</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3,299</td>
<td>65</td>
<td>3,299</td>
<td>65.0</td>
<td>51</td>
<td>51</td>
<td>182</td>
<td>ful</td>
<td>TRANS</td>
<td>RSCLASS</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2,282</td>
<td>3</td>
<td>754</td>
<td>1.0</td>
<td>754</td>
<td>573</td>
<td>94</td>
<td>sng</td>
<td>TRANS</td>
<td>TPDIR</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2,187</td>
<td>29</td>
<td>2,187</td>
<td>29.0</td>
<td>75</td>
<td>75</td>
<td>112</td>
<td>ful</td>
<td>TRANS</td>
<td>RSQIGETT</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2,051</td>
<td>2</td>
<td>1,026</td>
<td>1.0</td>
<td>1,026</td>
<td>857</td>
<td>121</td>
<td>sng</td>
<td>TRANS</td>
<td>D347T</td>
</tr>
</tbody>
</table>
Performance Analysis: SQL Trace – Explain

1. Find all bitmap streams for package restriction
2. Find all bitmap streams for fiscal periods
3. Calculate “AND” of bitmap streams
4. Access fact table partition by partition through local index
5. Join other requested dimensions to the result via hash join

© SAP AG 2002, BW Performance Tuning, Alex Peter, 138
Basic Help to Read the Execution Plan

- **Read from inside out of the hierarchy**
  - Read a set of operations at the same level from top to bottom

- **Important node names**
  - **Hash Join**
    - Smaller Table is used to build up hash table on the join key (optimal: in memory)
    - Then scanning the larger table with hash table
  - **Nested Loop**
    - For each relevant row in the first table, find all matching rows in the other table
  - **Bitmap Merge (=Or) / Bitmap And**
  - **Partition Range Iterator**
ABAP Runtime Analysis (SE30) – 1 –

Analyze high ABAP Runtime

- Particularly useful for Customer Exits

Trace
transaction, program or function module ......

... or enable tracing in parallel session

Analyze results
Runtime Analysis Evaluation: Overview

Transaction: RSA1 = BW Administrator
Program: RSA1
User: PETERA

Execution time in micro:
- ABAP: 0%
- Database: 50%
- System: 50%

Runtime Analysis Evaluation: Hit List

<table>
<thead>
<tr>
<th>Call</th>
<th>No.</th>
<th>Gross</th>
<th>Net Gross (%)</th>
<th>Net (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime analysis</td>
<td>1</td>
<td>17,387,134</td>
<td>0</td>
<td>100.0</td>
</tr>
<tr>
<td>Call transaction RSA1</td>
<td>1</td>
<td>17,366,974</td>
<td>9,497</td>
<td>100.0</td>
</tr>
<tr>
<td>Program RSABV_START_NEW</td>
<td>1</td>
<td>17,374,686</td>
<td>2,219</td>
<td>99.9</td>
</tr>
<tr>
<td>Screen entry</td>
<td>1</td>
<td>17,286,538</td>
<td>281</td>
<td>99.4</td>
</tr>
<tr>
<td>FAX screen SAPSSYS0</td>
<td>1</td>
<td>17,284,387</td>
<td>55</td>
<td>99.4</td>
</tr>
<tr>
<td>Event LDB-Processing</td>
<td>1</td>
<td>17,264,751</td>
<td>3,306</td>
<td>99.4</td>
</tr>
<tr>
<td>Function RSAG_WKORDENCLSTART</td>
<td>1</td>
<td>17,168,250</td>
<td>105</td>
<td>99.0</td>
</tr>
<tr>
<td>Call screen 2001</td>
<td>1</td>
<td>17,161,626</td>
<td>5,942</td>
<td>98.7</td>
</tr>
<tr>
<td>Screen entry</td>
<td>1</td>
<td>17,112,219</td>
<td>621</td>
<td>98.4</td>
</tr>
<tr>
<td>DB2 screen SAPRSADM</td>
<td>1</td>
<td>16,972,050</td>
<td>65</td>
<td>97.6</td>
</tr>
<tr>
<td>Module(FBD) STATUS_2001</td>
<td>1</td>
<td>16,972,477</td>
<td>10,020</td>
<td>97.5</td>
</tr>
<tr>
<td>PERFORM CREATE_OBJECT_GL_PERIOD</td>
<td>1</td>
<td>13,153,540</td>
<td>301</td>
<td>75.7</td>
</tr>
<tr>
<td>Call method CL_RSAGW_ACTIVATION</td>
<td>1</td>
<td>13,152,615</td>
<td>616</td>
<td>75.6</td>
</tr>
<tr>
<td>Call method CL_RSAGW_GENERAL_AWW &gt; CONSTRUCT</td>
<td>1</td>
<td>12,424,766</td>
<td>170</td>
<td>71.5</td>
</tr>
<tr>
<td>Call method CL_RSAGW_GENERAL_AWW &gt; CONSTRUCT</td>
<td>1</td>
<td>12,409,091</td>
<td>130</td>
<td>71.4</td>
</tr>
<tr>
<td>Call method CL_RSAGW_GENERAL_AWW &gt; CONSTRUCT</td>
<td>1</td>
<td>12,222,460</td>
<td>1,406</td>
<td>70.3</td>
</tr>
<tr>
<td>Function RSAG_WKODCL_CALLBACK</td>
<td>1</td>
<td>11,974,692</td>
<td>1,160</td>
<td>59.0</td>
</tr>
<tr>
<td>PERFORM RSAGW_MAKE_TREE_FIRST_TIME</td>
<td>1</td>
<td>11,938,180</td>
<td>123</td>
<td>58.7</td>
</tr>
<tr>
<td>PERFORM RSAGW_MAKE_INTERNAL_GTREE</td>
<td>1</td>
<td>11,932,774</td>
<td>87</td>
<td>58.6</td>
</tr>
<tr>
<td>PERFORM RSAGW_EXPAND_LAST_ACTIVITY</td>
<td>1</td>
<td>11,937,201</td>
<td>163</td>
<td>56.9</td>
</tr>
<tr>
<td>PERFORM RSAGW_EXPAND_TREE</td>
<td>1</td>
<td>11,931,103</td>
<td>223</td>
<td>55.9</td>
</tr>
<tr>
<td>PERFORM GET_ALL_NODES_FROM_DB</td>
<td>1</td>
<td>11,567,846</td>
<td>101</td>
<td>55.5</td>
</tr>
<tr>
<td>PERFORM RSAGW_GET_ICUT_EGL</td>
<td>2</td>
<td>11,861,680</td>
<td>34,197</td>
<td>62.5</td>
</tr>
<tr>
<td>PERFORM F15_GET_AREA_EGL</td>
<td>2</td>
<td>11,360,117</td>
<td>731</td>
<td>36.8</td>
</tr>
<tr>
<td>PERFORM RSAGW_GET_RULE_EGL</td>
<td>4</td>
<td>8,710,085</td>
<td>4,887</td>
<td>32.9</td>
</tr>
</tbody>
</table>

List has been sorted
Runtime of function modules:

- **Gross Time**
  - Includes times for sub-function calls

- **Net Time**
  - Excludes times for sub-function calls
Avoid unnecessary loops

**Good:***

Delete from itab where field1 = condition.

**Bad:***

Loop at itab.
If itab-field1 = condition.
Delete itab.
Endif.
Endloop.
Avoid unnecessary selects

- Use DB features instead of programming individual logic

**Good**

```abap
SELECT SUM(amount) INTO Sum FROM tab WHERE year = '2002'.
```

**Bad**

```abap
Sum = 0.
SELECT amount INTO Add FROM tab WHERE year = '2002'.
Sum = Sum + Add.
ENDSELECT.
```
ABAP Tips – 3 –

- Avoid unnecessary selects
  - Try to reuse data

Good: SELECT f1,f2,f3 FROM table WHERE cond
Good: SELECT f1, f2 FROM table WHERE cond
Bad: SELECT f1, f3 FROM table WHERE cond
Avoid unnecessary selects

- Avoid nested SELECTs
- Tip: If you discover several full table scans with “FOR ALL ENTRIES” in Basis 6.20, change rsdb/max_in_blocking_factor from –1 to 500.

```
SELECT * FROM knb1
FOR ALL ENTRIES
IN I_kun WHERE
kunnr = I_kun-kunnr.
* do something
ENDSELECT.
```

```
LOOP AT I_kun
SELECT * FROM knb1
WHERE kunnr = I_kun-kunnr.
* do something
ENDSELECT.
ENDLOOP.
```
ABAP Tips – 5 –

Buffering DB tables

- Use Start Routine in Transformation Rules to build up internal tables and use them in the rule definition
- Use Array operations instead of SINGLE accesses
- Don’t forget to refresh the tables when necessary

**SELECT * FROM knb1 INTO itab FOR ALL ENTRIES IN I_kun WHERE kunnr = I_kun-kunnr.**

**LOOP at itab.**
  * do something
**ENDLOOP.**

**LOOP AT I_kun**
**SELECT * FROM knb1 WHERE kunnr = I_kun-kunnr.**
  * do something
**ENDSELECT.**
**ENDLOOP.**
ABAP Tips – 6 –

- **Accessing internal tables**
  - Sort internal tables and use binary search on it OR
  - Hashed Tables

- **Accessing DB Tables**
  - Fully buffered DB tables are transferred completely to the application buffer when accessed the first time (see SE11-settings of table)
  - Specify whole key in WHERE clause, if possible – or specify key which is included in one of the secondary indices
  - Try to avoid “NOT” in WHERE clause, as there is no index support
Analyze high database runtime

- Screen and features differ for each DBMS

Data Buffer Size & Quality, buffer quality should be > 95%
Analyze high database runtime

- Screen and features differ for each DBMS

- Displays all DB processes

- Locks held by another process

- Check SQL statements from history

- Check Table Buffers
Database Tables and Index Monitor (DB02)

Analyze high database runtime
- Screen and features differ for each DBMS

Refresh statistics for up-to-date information on indices and tables

Compare table/index size

Check missing indices
Buffer / Memory Monitor (ST02) – 1 –

Analyze high database runtime
- Buffers not used?
- Buffer replacement expensive?

Hit ratio > 95%

Check free space and free entries

Swaps?

Table Buffer

Exp/Imp SHM \rightarrow OLAP Cache

THE BEST-RUN BUSINESSES RUN SAP
Buffers in BW

- Generic Key Table Buffer
  - Most BW-specific control tables
  - RS*-Tables
- Single Record Buffer
  - Master Data Tables
- Export/Import Shared Memory
  - OLAP Query Cache
Operating System Monitor (ST06) – 1 –

Analyze high system runtime

**Detailed Analysis**

**CPU Utilization**

**Memory Swapping** (page out)

**Disk Utilization** (I/O Bottleneck)

**LAN throughput** (network problems)
Operating System Monitor (ST06) – 2 –

Analyze high system runtime

- Detail Analysis

snapshot analysis

24h analysis
Web Query Analysis

Typical questions for (single) Web queries:

- How do I measure the total response time of a Web Query?
- How do I analyze where most of the query execution time is spent in BW?
- Is the time required for rendering the Web page on the front-end significant?

Typical questions for Portal pages:

- How are multiple URLs processed on the time line, i.e. which URL is called at what time, and how long does it take to come back?
- Which URLs dominate the overall response time of a Portal page?
Web Query Analysis – Overview

2 different measuring points:

"frontend"

Monitoring of WAS / BW / DB internal activities

"backend"

Frontend monitoring
Network traffic monitoring

HTTP Monitor

Web Browser

IEMonitor

HTTP

HTTP

request

response

HTML

ICM

OLAPINIT

DB

OLAP

Rendering

Monitoring of WAS / BW / DB internal activities

"frontend"
Web Query Analysis – Frontend Monitoring

Key Performance Indicators for Frontend Monitoring:

- Query response time
  - Time to load HTML page
  - Frontend rendering time
- Transferred data volume
- Number of round-trips between frontend and server

IEMon.exe

- Simple Web browser replacement with protocol logging facilities
- Used to analyze load and frontend rendering time of an HTML page
- Easy deployment

httpmon.exe

- Set up as a proxy server
- Requires changes of IE settings
- Requires environment variables to be set for configuration
IEMon.exe – Usage

Simple Web browser replacement

No installation process, just copy and execute

Uses IE control of the Internet Explorer installed on workstation

---

Path to log file

View log file

View log as it is produced

Navigate button

Stop navigation

Address bar

Status line
IEMon.exe – Logfile

Load time

Frontend rendering time

Millisecond time stamps

Status information
IEMon log files can become very complex and lengthy. Hard to find matching events for load and render time.
IEMon.exe – Log File Analysis

IEMon log file contains information for lots of different events

BUT: only 3 types of events required to determine load and rendering time:

- Before Navigate event
- Navigate Complete event
- Document Complete event

Sequence of events in log file depends on structure of HTML code

Complex log files: analysis tool for information extraction

analyze_iemon.exe

- takes path to IEMon log file as command line parameter
- produces comma separated list of values for start time, load time, rendering time, total response time, and URL
- Output can be redirected into .csv file and conveniently be opened with MS Excel
IEMon.exe – Log File Analysis in Excel

<table>
<thead>
<tr>
<th>Start time</th>
<th>Rel. Start</th>
<th>Load time</th>
<th>Render Time</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>46453963</td>
<td>0</td>
<td>3715</td>
<td>16535</td>
<td>20250</td>
</tr>
<tr>
<td>46457420</td>
<td>4657</td>
<td>4547</td>
<td>9133</td>
<td>18547</td>
</tr>
<tr>
<td>46456841</td>
<td>5738</td>
<td>3726</td>
<td>9163</td>
<td>18867</td>
</tr>
<tr>
<td>46458782</td>
<td>5909</td>
<td>3805</td>
<td>8833</td>
<td>18347</td>
</tr>
<tr>
<td>46458803</td>
<td>5949</td>
<td>3366</td>
<td>10726</td>
<td>20240</td>
</tr>
<tr>
<td>46459822</td>
<td>5969</td>
<td>3446</td>
<td>8462</td>
<td>17996</td>
</tr>
<tr>
<td>46460227</td>
<td>7371</td>
<td>5263</td>
<td>2003</td>
<td>10637</td>
</tr>
<tr>
<td>46460436</td>
<td>7342</td>
<td>5343</td>
<td>1052</td>
<td>18337</td>
</tr>
<tr>
<td>46461135</td>
<td>8302</td>
<td>9644</td>
<td>60</td>
<td>17306</td>
</tr>
<tr>
<td>46461295</td>
<td>8832</td>
<td>9966</td>
<td>320</td>
<td>18657</td>
</tr>
<tr>
<td>46461275</td>
<td>8422</td>
<td>11407</td>
<td>411</td>
<td>20240</td>
</tr>
<tr>
<td>46463397</td>
<td>10544</td>
<td>1290</td>
<td>10</td>
<td>19734</td>
</tr>
<tr>
<td>46468888</td>
<td>17135</td>
<td>180</td>
<td>0</td>
<td>17315</td>
</tr>
<tr>
<td>46470619</td>
<td>17796</td>
<td>200</td>
<td>0</td>
<td>19996</td>
</tr>
<tr>
<td>46471120</td>
<td>10347</td>
<td>10</td>
<td>10</td>
<td>19999</td>
</tr>
<tr>
<td>46472542</td>
<td>13038</td>
<td>150</td>
<td>0</td>
<td>19938</td>
</tr>
</tbody>
</table>

Output produced by analyze_iemon

Graphical representation with a few clicks
Set up as a proxy server

Configuration by environment variables:

- **HTTP_PROXY**: `proxy_server:proxy_port`
  (e.g. `pwdf0754:1080`)
- **NO_PROXY**: list of web server domains which do not need proxy (e.g. `*.sap.corp;*.sap-ag.de`)

`httpmon` listens on port 8000 by default → change Internet Explorer settings
httpmon.exe – Usage

Be sure that you delete the ICM Server Cache (TA SMICM Goto → HTTP Server Cache → Invalidate → Global) to get the compressed data volume!

- Transferred data volume (compressed!)
- Path to log file
- httpmon listen port
- httpmon target proxy and port
- BW server pwdf0754
- Port 1080
- Transferred data volume (compressed!)
- Set data volume counters to 0

Path to log file:
httpmon
Listen port:
httpmon
Target proxy and port:
httpmon pwdf0754:1080
BW server:
pwdf0754
Httpmon.exe – Use Case

First execution: download of MIME objects

Second execution: MIME objects were read from browser cache
Frontend Monitoring – Summary

3 important tools for frontend monitoring of Web pages:

- **IEmon.exe:**
  - Load time
  - Frontend rendering time

- **analyze_iemon.exe**
  - Produce .csv file from IEMon log file to view results in Excel

- **httpmon.exe:**
  - Transferred data volumes
  - Number of round-trips between frontend and server

These tools are available to customers (without support) and can be downloaded from SAP Service Marketplace, alias /bw, section Performance

**Recommendation:** Do not trace more than one navigation step at a time. For multiple navigation steps use individual log files!

- Open IEMon log file, use Save as ... to store in a different file
- Delete contents of log file in editor, save back to original file
Backend Monitoring

Open question: How to analyze query execution in BW backend?

Query analysis: trace of ABAP function calls (transaction SE30)

First solution: call SE30 in transaction RSRT
- Display result as HTML
- BUT: MIME handling and other things are different from a „real“ http service connection

Second solution: switch on ABAP trace for each incoming http request
- Configuration of http services (transaction SICF)
- Can trace only those service requests coming from a specific IP address  ➔ no impact on other concurrently executing requests
Restrict tracing on requests handled by this service

Restrict tracing on requests coming from SAPGUI IP address only

Idle time limit
After tracing has been activated in SICF:

- Start Web browser (Internet Explorer, IEMon, ...)
- IMPORTANT: the browser must be started AFTER switching on the trace. Otherwise no trace files will be written!
- Execute query
- Use transaction SE30 to analyze ABAP trace
Sort descending by column „Net“ to identify most time consuming function modules.
Backend Monitoring – Summary

Use transaction SICF to switch on ABAP traces for incoming http service requests

Use transaction SE30 to analyze ABAP traces

Note:
- ABAP traces can have substantial impact on query response times!
- BUT: times reported in ABAP trace do not contain SE30 execution times

… don’t forget to switch off ABAP tracing when you’re finished …
Information on Performance

- SAP Service Marketplace alias ‘BW’ → **Performance**
- SAP Training BW360: Performance & Administration
- Collective performance notes BW 3.x
  - Note 567745: DB-specific Settings
  - Note 567746: Query & Web Performance
  - Note 567747: Extraction & Loading