

# Sizing

## ASAP FOR BW ACCELERATOR SAP BUSINESS INFORMATION WAREHOUSE



---

*How to size of a BW system*

Version 3.0  
December 2002

## Table of Contents

ASAP for BW Accelerator .....	1
<b>1 INTRODUCTION.....</b>	<b>1</b>
<b>2 SIZING .....</b>	<b>1</b>
<b>2.1 R/3 System .....</b>	<b>1</b>
<b>2.2 BW-Systems.....</b>	<b>1</b>
2.2.1 Overview.....	1
2.2.2 First sizing approach (T-shirt sizing).....	3
2.2.3 Detailed sizing approach.....	4
2.2.3.1 Disk Space Requirements in General.....	4
2.2.3.1.1 Overview .....	4
2.2.3.1.2 Estimating an Infocube .....	4
2.2.3.1.3 Dimension Tables .....	4
2.2.3.1.4 Fact Tables.....	5
2.2.3.1.5 PSA tables.....	5
2.2.3.1.6 ODS tables.....	5
2.2.3.1.7 Limitations.....	5
2.2.3.1.8 Re-Estimation .....	5
2.2.3.2 Simplified estimation of disk space ( used by the quicksizer) .....	7
2.2.3.3 CPU requirements in general .....	8
2.2.3.3.1 Simplified CPU sizing for data staging .....	9
2.2.3.3.2 Simplified CPU sizing for queries.....	9
2.2.3.4 Memory requirements in general .....	11
2.2.3.4.1 Simplified memory sizing for data staging.....	12
2.2.3.4.2 Simplified memory sizing for queries.....	13
2.2.3.5 Sizing result for 2 –Tier systems.....	14
2.2.3.6 Sizing result for 3-Tier systems .....	14
2.2.3.7 Frontend PC requirements .....	15
2.2.3.8 Network requirements.....	15

# 1 Introduction

This document describes how to estimate hardware resources that are necessary to run a particular workload on a BW System, providing guidelines for the definition of hardware requirements like main memory, disk space and CPU. Generally, it is much more difficult to estimate the sizing requirements of a BW system than an R/3 system. The reason is that user behavior is more difficult to project than with the R/3 system, as BW provides great flexibility in order to meet the varied reporting needs of an organization. With this in mind, the goal of this document is to present general guidelines that can help in determining hardware requirements for a BW system. **It is always encouraged to work directly with hardware vendors to receive further assistance in sizing a BW system.**

## Software Version Supported

This document was written for BW version 3.0, but it should apply to all versions of BW.

The recommendations provided in this document, regarding estimation of hardware requirements of BW systems, are assumptions based on empirical data gathered from existing installations and based on the output of certified BW benchmarks.

# 2 Sizing

Necessary resources for achieving a particular throughput and response times depend on several factors such as e.g. database, operating system, number of users, amount of online/background operations, load profile.

The values always refer to a minimum configuration and should be increased in cases of doubt.

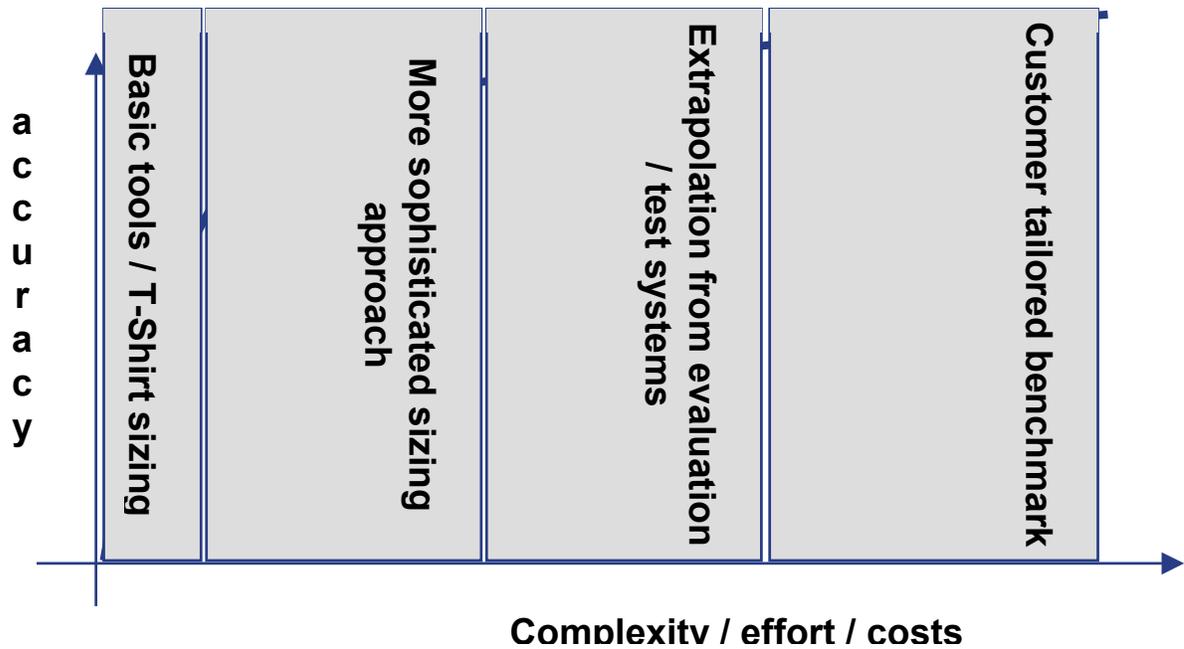
## 2.1 R/3 System

When connecting a BW system to an R/3 system it has only a minor net impact on the hardware sizing of the R/3 system. The reason for this is that the presence of a BW system alleviates much of the load that reporting typically places on the R/3 system. On the other hand data extraction can require significant R/3 system resources.

## 2.2 BW-Systems

### 2.2.1 Overview

The reliability of a BW sizing is based on the given information. Based on the information you can choose one out of the 2 different sizing methods. The first approach (T-shirt sizing) is used for starting projects gathering the first experiences with BW. That means whether the customer nor the consultant is able to provide detailed information about the implementation. If there are more detailed information is available you should use the 2 approach including Disk, CPU and Memory sizing.



In the graphic you can easily see the dependencies between accuracy and the costs of the sizing procedure. The more accurate the sizing should be, the more complex is the sizing. Even using the detailed sizing approach introduced in this document is based on some assumptions (described in the following chapters) which can in some cases result in a misleading sizing.

A reliable sizing can only be done running a customer tailored benchmark.

**2.2.2 First sizing approach (T-shirt sizing)**

Categories of sizes have been developed for the hardware sizing for a BW System. They correspond to T-shirt sizes.

<b>BW Category (T-Shirt size)</b>	<b>Configuration</b>	<b>Hardware</b>	<b>No. of users</b>
<b>XS</b>	DB + App-server on one server	>= 1 GB of RAM, >=2CPUs, DB 30 – 50GB	1 – 10
<b>S</b>	DB + App-server on one Server	>= 2 GB of RAM, 2 – 4 CPUs DB 50 – 100GB	10 – 20
<b>M</b>	DB -, App-server on one or several servers dependant on the hardware configuration	>= 4 GB of RAM, 4 CPUs for DB- and App-server DB 100 – 200 GB	20 – 50
<b>L</b>	DB- + App-server on several servers dependent on the hardware configuration	>=4 GB of RAM, DB instance >=4 CPUs, App-server >=2CPUs, >= 2 GB of RAM DB 200 – 1000 GB	50 – 150
<b>XL</b>	One DB instance, multiple App- servers	>=4 GB of RAM, DB instance >=8 CPUs, App-server >=4CPUs, >= 4 GB of RAM >1000 GB	> 150

### 2.2.3 Detailed sizing approach

#### 2.2.3.1 Disk Space Requirements in General

##### 2.2.3.1.1 Overview

Disk space requirements depend heavily on the design of Infocubes and data distribution. The following hints on estimating disk space do not account for possibly sparse tables or compression the database might use. There is a quicksizer available on <http://service.sap.com/quicksizer> that helps you perform what is described below.

The Business Information Warehouse software including the development workbench requires approximately 9 GB of disk space.

The required disk space for data depends on:

- Number of Infocubes
- Number of key figures and dimensions per Infocube
- Volume of data loaded
- Number of aggregates
- Amount of master data
- Number and size of flat files used for data load, residing on application servers
- The number and size of PSA and ODS tables
- Indexes

##### 2.2.3.1.2 Estimating an Infocube

When estimating the size of an Infocube, tables like fact and dimension tables are considered. However, the size of the fact table is the most important, since in most cases it will be 80-90% of the total storage requirement for the Infocube.

When estimating the fact table size consider the effect of compression depending on how many records with identical dimension keys will be loaded.

The amount of data stored in the PSA and ODS has a significant impact on the disk space required. If data is stored in the PSA beyond a simply temporary basis, it could be possible that more than 50% of total disk space will be allocated for this purpose.

##### 2.2.3.1.3 Dimension Tables

- Identify all dimension tables for this InfoCube.
- The size and number of records need to be estimated for a dimension table record. The size of one record can be calculated by summing the number of characteristics in the dimension table at 6 bytes each. Also, add 6 bytes for the key of the dimension table.
- Estimate the number of records in the dimension table.
- Adjust the expected number of records in the dimension table by expected growth.
- Multiply the adjusted record count by the expected size of the dimension table record to obtain the estimated size of the dimension table.

### 2.2.3.1.4 Fact Tables

- Count the number of key figures the table will contain, assuming a key figure requires 10 bytes.
- Every dimension table requires a foreign key in the fact table, so add 6 bytes for each key. Don't forget the three standard dimensions.
- Estimate the number of records in the fact table.
- Adjust the expected number of records in the fact table by expected growth.
- Multiply the adjusted record count by the expected size of the fact table record to obtain the estimated size of the fact table.

### 2.2.3.1.5 PSA tables

The record length of a PSA table can be estimated by adding the field lengths of all the fields of the respective InfoSource and 48 bytes for request-id and other fields being added by the system.

Multiplying the record length by the number of records being held in the PSA table will give you an estimate on disk space needed per PSA table.

Data in PSA might be deleted regularly depending on your scenario (e.g. use ODS for permanent storage). Estimate the number of records accordingly.

### 2.2.3.1.6 ODS tables

The record length of a ODS table can be estimated by adding the field lengths of all the fields of the respective InfoSource.

After activation of data the activation queue is empty, so the space needed for new data can in most cases be neglected.

For the change log table add 48 bytes for request-id and other fields being added by the system.

Multiplying the record length by the number of records being held in the ODS table will give you an estimate on disk space needed per ODS table.

The number of records of the change log table depends on the average number of updates and the delta type used for the ODS object.

The minimum size of the change log table has the same size like the active data table. (assuming that all records are distinct.)

### 2.2.3.1.7 Limitations

There are no limits to the number of InfoObjects, number of InfoCubes, or the number of reports per InfoCube.

The system administrator can create 13 dimension tables in addition to the dimensions for time, Packet ID, and measure of unit for each InfoCube.

Number of characteristics in all dimension tables:

Each of the dimension tables can have 248 InfoObjects, computed by  $255 - 6$  (no access to these fields)  $- 1$  (Dimension ID) to obtain 248. For performance reasons we recommend a maximum of 16 InfoObjects per dimension.

A fact table can have up to 233 key figures, computed by  $255 - 6$  (no access to these fields)  $- 16$  (dimensions) to obtain 233.

### 2.2.3.1.8 Re-Estimation

After making an estimate, you should verify it by loading a small amount of data, such as a few thousand records and then analyzing the space used to confirm the original estimate. You should also analyze the input sources to determine the number of records from each source system.

If RAID (Redundant Array of Inexpensive Disks) is utilized, adjust the overall estimate accordingly. For example, when using RAID 1 (disk mirroring), you should double your estimate.

Estimates on disk space should be checked frequently. Data Warehouse requirements can change over time and can be dramatically changed by apparently minor modifications. Therefore, the impact on data storage requirements should be considered for every change. Also, reviews need to be performed periodically, such as every three to six months, in order to ensure that current growth estimates are correct.

### 2.2.3.2 Simplified estimation of disk space ( used by the quicksizer)

The following assumptions are made in order to simplify the disk sizing:

- compression factor resulting out of identical records is 0% (all records are distinct)
- index space is added with 100%
- space for aggregates is additional 100%
- compared to the size of the facttable we ignore the size of the dimension tables
- PSA data is deleted after upload or after a certain time.
- No. of masterrecords is less than 10% of the facttable (could be wrong for Retailcustomers)
- We ignore compression algorithm of the different database partners. (e.g. 10 digits are stored in 6 bytes or less and 17 digits are stored in 10 bytes or less in certain databases)

A simplified estimation of disk space for the BW can be obtained by using the following formula:

For each cube:

Size in bytes =

$(n + 3) \times 6 \text{ bytes} + (10 \text{ bytes} \times m) \times$

[ rows of initial load + rows of periodic load \* no. of periods]

**n** = number of dimensions

**m** = number of key figures

+

For each ODS (incl. change\_log):

Length of the infosource \* [ rows of initial load + rows of periodic load \* no. of periods]

**Remark** : the changelog has the same size like the active ODS table if we load distinct data. The maximum could 2 times of the active ODS table.

To the obtained results you have to add

- + 20% per PSA
- + 100% of total size of the infoproviders for aggregates / changelog (ODS)
- + 100% for indexes
- + 10% Master data
- + 2\* size of the biggest facttable at least 10 GB

**Remark :** the changelog has the same size like the active ODS table if we load distinct data. The maximum could 2 times of the active ODS table.

### 2.2.3.3 CPU requirements in general

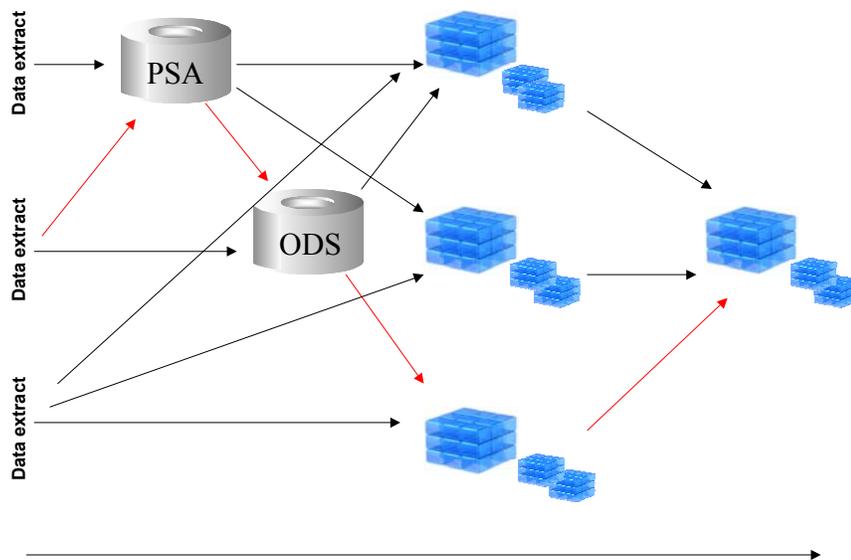
The CPU requirements of BW depends heavily on the design of the datamodel and the queries/user behavior.

The CPU resources for data staging depends on:

- number of infoproviders
- volume of data loaded
- number of aggregates
- amount of rollups/changeruns
- number of distinct requests

In order to find the cpu requirements you have to determine the critical path for the data staging. This path is different from customer to customer and could be different from upload to upload.

E.G. the redline in the picture below describes the critical Path incl. load, rollup, change\_run, index rebuild, gathering statistics. This path has to be measured in order to get the number of records per hour , which could be applied to the system.



The CPU resources for queries depends on:

- number of infoproviders
- number of connected users
- number of aggregates
- types of queries

In order to find the cpu requirements, you have to classify your queries in different query classes like “easy”, “medium” and “heavy” and analyze the different response times in order to get relational factors. Additionally you have to classify your user behavior in different groups like “normal”, “advanced” and “power” users and to examine which user type is using which kind of query.

### 2.2.3.3.1 Simplified CPU sizing for data staging

The following assumptions are made in order to simplify the cpu sizing for data staging:

- A system is able to apply 750000 records per hour per job
- No special Transformation rule and update rule are used
- The critical path contains a ODS –object , a infocube and 10 aggregates
- The data applied to the system could be split into several requests (one job per request)
- 350 SAPS are used for one job

The CPU requirements are calculated as follow:

**SAPS =**

**#jobs \* 350 SAPS / 0,65 (at 65% CPU consumption)**

While the number of jobs are calculated as following:

**#jobs = {sum (delta upload)/ maintenance window } / 750.000**

### 2.2.3.3.2 Simplified CPU sizing for queries

The number of users working in the system forms the basis for estimating cpu resources of the database and application servers. Different definitions of “user” are available:

- normal users
- advanced users
- power users

- Low user: accesses the BW System from time to time, is typically a user seeking information occasionally using predefined reports and generates in average over the week **1 Navigationstep** per hour
- Medium user: accesses the BW System regularly and continuously, is typically a user navigating within reports supported by aggregates and generates in average over the week **11 Navigationstep** per hour
- High user: works intensively with the BW System, the high user who runs ad hoc queries mostly not supported by aggregates and generates in average over the week **33 Navigationstep** per hour

Remark:

- The classification of the users (normal, medium, high) is corresponding to the user classification within the early watch report.
- A default user distribution based on a customer survey is 71% : 27% : 2%

Additionally we have to define different types of reports being used within BW:

- easy queries
- medium queries
- high queries

- „Easy“ queries: predefined reports using optimal aggregates. Waited with a factor **1**
- „Medium“ queries: slicing and dicing, navigating in reports, using various aggregates Waited with a factor **1,5** (that means the query uses 1,5 times more of resources compared to an easy query)
- „Heavy“ queries: ad-hoc reports with unpredictable navigation paths, access of detail data, full table scans. Waited with a factor **5** (that means the query uses 5 times more of resources compared to an easy query)

Remark:

These factors are based on the BW benchmark scenario. In different scenarios using other reports the factors could be different.

Using these definitions we are able to build the following matrix which should be filled by the responsible person doing the sizing:

	Easy query <b>1</b>	Medium query <b>1,5</b>	Heavy query <b>5</b>
# normal users <b>X</b>	80%	20%	0%
# advanced users <b>Y</b>	50%	50%	0%
# power users <b>Z</b>	0%	0%	100%

Remark:

The distribution within this matrix should be changed according to the customer load. If this is not known you can use this distribution as a default assumption.

The total number of navigationsteps handled by a system is computed in the following way (according to the example above):

Nav/h =

$$\begin{aligned}
 \text{Total \# of Nav/h} = & X * (0,8 * 1 + 0,2 * 1,5 + 0 * 5) * 1 + \\
 & Y * (0,5 * 1 + 0,5 * 1,5 + 0 * 5) * 11 + \\
 & Z * (0 * 1 + 0 * 1,5 + 1 * 5) * 33
 \end{aligned}$$

Remark:

One of the outputs of a BW benchmark is: **Total # of Nav/h**

In order to map these requirements to SAPS we have to convert Nav./h into SAPS

**SAPS =**

$$(\text{Nav/h} * 1,33 * 2,28 * 9 / 60) / 0,65 \text{ (at 65\% CPU consumption)}$$

While 2,28 is a relational factor between a BW benchmark and a SD benchmark (the definition of SAPS is based on the SAP SD benchmark) and 1,33 is an adjustment factor measured from productive customers.

#### 2.2.3.4 Memory requirements in general

The memory requirements of BW depends heavily on the design of the datavolume and the number of connected users

The memory resources for data staging depends on:

- number of parallel jobs

- number of records loaded within one data package
- Complexity of update and transformation rule
- degree of parallel processes within the RDBMS
- BW buffers configured for the system (table buffers, program buffers, export/import,...)
- RDBMS buffers (data buffer, process buffer)

In order to get the information about the memory requirements, you have to check the memory consumption for each distinct load step in your system and summarize the memory with consideration of parallel jobs . The critical path described above is not always the step, which consumes the most memory.

The memory resources for for queries depends on

- number of connected user
- keep alive time for WAS
- data volume of the result set

During the daily load, the system has to be examined, how many users are connected to the system. The memory used by these users can be measured with the transaction sm04 and could be summarized. Another parameter which influence the memory consumption is the keep alive time of the Web Application Server. The WAS is holding a connection to the system for this time. Even if the user has closed his browser.

### 2.2.3.4.1 Simplified memory sizing for data staging

The following assumptions are made in order to simplify the memory sizing for data staging:

- Number of records per data package is 50000
- No special Transformation rule and update rule are used
- The critical path contains a ODS object, a infocube and 10 aggregates
- 300 MB are used for one job
- minimum of BW buffer is 500 MB
- minimum RDBMS buffer is 700 MB
- Max. RDBMS buffer is 10 GB
- 10 MB buffer is used for each mio. Record of the largest infocube
- 120 MB is used for each process connected to the RDBMS
- for parallel processing 2 times \* number of jobs is used within the RDBMS

The BW memory requirements are calculated as follow:

**Memory BW =**

**#jobs \* 300 MB + 500 MB**

While the number of jobs are calculated as following:

$\#jobs = \{ \text{sum (delta upload) / maintenance window} \} / 750.000$

This number could be used as a suggestion how many batch processes should be configured.

The RDBMS memory requirements are calculated as follow:

**Memory =**

**700 MB + max(rows of infoprovider)/1 mio. \* 10 MB + #Jobs \* 120MB + 2\*#Jobs\*120MB**

While the number of jobs are calculated as following:

$\#jobs = \{ \text{sum (delta upload) / maintenance window} \} / 750.000$

### 2.2.3.4.2 Simplified memory sizing for queries

The following assumptions are made in order to simplify the memory sizing for queries:

- Users are connected via WAS
- Keep alive time is 60 seconds
- 30 MB are used for one job
- minimum of BW buffer is 500 MB
- minimum RDBMS buffer is 700 MB
- Max. RDBMS buffer is 10 GB
- 120 MB is used for each process connected to the RDBMS
- no parallel processing for queries
- One BW process can handle 1 high or 2 medium or 5 low users concurrently

The BW memory requirements are calculated as follow:

$$\text{Memory} = 500 \text{ MB} + \text{number of connected users} * 30 \text{ MB}$$

While the number of connected users are calculated as following:

$$\begin{aligned} \text{Number of connected users} &= (\# \text{high user} + \# \text{medium user} + \# \text{low user}) / 3600 \text{ s} * \text{keep\_alive\_time} \\ &= (\# \text{high user} + \# \text{medium user} + \# \text{low user}) / 3600 * 60 \end{aligned}$$

The RDBMS memory requirements are calculated as follow:

$$\begin{aligned} \text{Memory} &= \\ &700 \text{ MB} + \max(\text{rows of infoprovider}) / 1 \text{ mio.} * 10 \text{ MB} + \\ &\text{number of concurrent users} * 120 \text{ MB} \end{aligned}$$

While the number of concurrent users are calculated as following:

$$\text{Number of concurrent users} = (\# \text{high user} + \# \text{medium user} / 2 + \# \text{low users} / 5) / 3600 \text{ s} * 60$$

This number could be used as a suggestion how many dialog processes should be configured.

### 2.2.3.5 Sizing result for 2-Tier systems

If the system should be installed on central server the following calculation should be made in order to get the right amount of CPU and memory consumption:

$$\text{SAPS} = \text{maximum}(\text{SAPS data staging or SAPS query})$$

$$\text{Memory} = \text{maximum}(\text{memory data staging (RDBMS + BW)} \text{ or } \text{memory query (RDBMS + BW)})$$

### 2.2.3.6 Sizing result for 3-Tier systems

If the system is divided in a dedicated RDBMS server and a BW server the the right amount of CPU and memory consumption is:

BW:

$$\text{SAPS} = \text{maximum}(50\% \text{ SAPS data staging or } 80\% \text{ SAPS query})$$

$$\text{Memory} = \text{maximum}(\text{memory data staging (BW)} \text{ or } \text{memory query (BW)})$$

RDBMS:

$$\text{SAPS} = \text{maximum}(50\% \text{ SAPS data staging or } 20\% \text{ SAPS query})$$

$$\text{Memory} = \text{maximum}(\text{memory data staging (RDBMS)} \text{ or } \text{memory query (RDBMS)})$$

The total amount of SAPS of a 3-Tier system could be higher compared to a 2-Tier system. The reason is based on the fact that CPU resources of a central system can be shared dynamically.

### **2.2.3.7 Frontend PC requirements**

Please look for note 321973

### **2.2.3.8 Network requirements**