An IT Planning – Capacity Management service is highly dependent from the scope of the service. There is always a difference in the service reports based on the approaches. For example, an initial sizing of a system is very different from a delta sizing. Even between the same approach can be huge differences between customers.

The next slides are not part of the service but necessary to understand the approaches. Each service is scoped together with the customer before start. After the report was finished or a milestone (e.g. baseline finished) is reached the results are presented to the customer.

Please remind this report is just an example. Each report looks different for other customers with other scopes and problem definitions. This service is highly individual.

This example report shows a sizing based on a production system, i.e. delta sizing which is a probable case for a high amount of deliveries for an IT Planning – Capacity Management service.

Remarks are only visible in this example report. They provide additional information for you about the purpose and outcome of the steps taken for the analysis. The results are normally presented directly to the customer to give the necessary explanations.
Initial / Greenfield sizing

**Initial sizing**

- Is used for new SAP systems. The sizing cannot be based on existing workload statistics as there is no SAP system.
- The approach of the initial sizing depends from the expected system size and the accuracy needed. Every step is not necessary every time:
  1. The first approach is to use the Quick Sizer to base a sizing estimation on expected business volumes and/or users.
  2. Due to the limitations of the SAP Quicksizer a second step might be to narrow down the business processes to the most critical in terms of capacity consumption and performance.
  3. The last and most expensive method would be to perform measurements to extrapolate the expected workload.
Delta sizing / resizing

Delta sizing
• This kind of sizing is always based on an existing SAP system.
• This sizing can be used for workload increases by:
  – Check of current system size in terms of actual capacity consumption and expected year to year business growth
  – Workload increase due to technical reasons, e.g. Upgrade, Unicode conversion
  – Business Volume increases, e.g. go live of new regions
  – New functionalities, e.g. go live of new ERP module or new business processes
• The delta sizing is based on the current capacity consumption, current application workload and scalability behavior. Based on this information an extrapolation of the expected capacity consumption can be made.
• There are “special” scenarios, e.g. system consolidations or carve outs, which can be supported as well. However, this approach has to be discussed with the customer and it has to be decided whether the approach is feasible or not.
• The Delta sizing is based on building blocks which are used depending on the circumstances.
Greenfield or initial sizing is an iterative process

- During which initial assumptions regarding volume or response time requirements are successively adjusted and mapped to standard and expert sizing methodologies.
- The main characteristic of greenfield sizing is the chronological component and the fact that the target sizing may change along the project.

Delta sizing / Resizing

- Uses current resource consumption and scalability behavior of the system as a basis.
- On top of that additional contributors are added, depending on whether they apply or not.
  - Also, the sequence may be adapted to the project vision, especially if you look a few years into the future
- There are more or less only two areas of coincidence between the two approaches:
  - Quick Sizer for new SAP standard applications
  - Sizing measurements (or expert sizing) for new custom applications.
Future capacity requirements always are the sum of different elements contributing to an increased load. For the different elements, different sizing approaches are applied.

Depending on the project plan, building blocks may be swapped.

The color scheme indicates the closeness to factual basis.

<table>
<thead>
<tr>
<th>Delta Sizing</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sum of Building Blocks – Overview</strong></td>
<td><strong>Remarks</strong></td>
</tr>
<tr>
<td>Future capacity requirements always are the sum of different elements contributing to an increased load. For the different elements, different sizing approaches are applied. Depending on the project plan, building blocks may be swapped. The color scheme indicates the closeness to factual basis.</td>
<td><strong>New Custom Application</strong> Custom applications are the most difficult ones to assess. Typically, expert sizing is required. <strong>New SAP Standard Application</strong> Use of standard sizing guidelines or the Quick Sizer until an expert sizing makes sense. <strong>Target utilization</strong> SAP standard is 65%, may be adapted according to need <strong>Additional business units go live</strong> Apply factor based on business volume associated with roll out for Roll out of Telesales. <strong>Business Growth</strong> Typically in year-on-year growth. <strong>Upgrade, Unicode conversion</strong> We normally assume +10% for the Unicode conversion, the upgrade factor depends on application and release. <strong>Current resource consumption</strong> Measured KPIs: CPU, memory, disk, business throughput. Check scalability of current load.</td>
</tr>
</tbody>
</table>

© 2014 SAP SE or an SAP affiliate company. All rights reserved.
A service report for a delta sizing consists of following parts.

- Header: introduction, summary, actions list
- Baseline:
  - Description of landscape (capacity deployed)
  - Technical workload analysis (capacity consumed)
  - Application workload analysis (top consumer)
  - Scalability analysis (how does the system scales if additional load is applied)
- Sizing forecast: can have very specific sizing blocks depended from the customer situation and the scope
Starting Point

This service was initiated on request of Example Customer Inc. for system <SID>.

<SID> will take additional workload for the Russian market. The load increase of the first wave is expected at 20-30%. After all rollouts the expected workload increase will be at factor 10.

The focus of this service is to show whether the system will be able to handle the expected workload increase of the next wave. Further the required system resources should be shown for the forecast of the overall workload increase by factor 10.

Remarks

- The starting point describes the scope of the service which was raised during the scoping or a kick-off call of the service.
We created a performance baseline as a first step. This showed that the application hosts have enough resources to handle the upcoming workload of the next wave. The DB host is running on a hardware which is utilized at the optimum. Additional load will lead to CPU bottleneck. However, based on the scalability analysis we think that the improvement potentials found in the coding will reduce the workload in the same amount than the load increase.

We have performed a forecast for the overall workload increase by factor 10. The numbers we got are quite inaccurate. They will reduce significantly by any action taken to reduce unnecessary workload. So the values we got are only to be seen as a value the system would consume if no action takes place.

We recommend a step-by-step approach for upcoming workload increases. This enables you to check and optimize the resource consumption with regards to the workload increase.

Remarks

• The summary describes the major findings and the recommendations.
### Action / Issue List

<table>
<thead>
<tr>
<th>Action</th>
<th>Priority</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Low free CPU capacities on hardware ABC</strong></td>
<td>High</td>
<td>Before / During next workload increase</td>
</tr>
<tr>
<td>Pay attention to potential CPU bottleneck situations on hardware ABC when adding additional load to the DB host. Try to relieve the hardware by moving workload.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step by step approach for further rollouts</strong></td>
<td>High</td>
<td>Further rollout</td>
</tr>
<tr>
<td>Follow a step-by-step approach for further rollouts in order to review the hardware consumption with regards to the business volume increase. If necessary further optimizations should be taken into account.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reduce unnecessary workload</strong></td>
<td>High</td>
<td>Before next workload increase</td>
</tr>
<tr>
<td>Implement the recommendations from the scalability analysis. This will help to reduce the workload. Try to restrict expensive queries issues by users.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Database parameters</strong></td>
<td>Medium</td>
<td>During next system maintenance</td>
</tr>
<tr>
<td>Implement parameter recommendations regarding the database</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Remarks**

- An action list is provided to address the issues found in the service.
Data sources for analysis

Data based on

- Transaction ST06 (SAPHOSTAGENT)
  - Hourly averages for CPU consumption (typically consisting of 60 measurements)
  - Data available for all host in timeframe from May 02\(^{nd}\) 2014 – June 06\(^{th}\) 2014
- Transaction /sdf/mon
  - 1 minute averages for number of work processes and extended memory utilization
  - Data available for all host in timeframe from May 28\(^{th}\) 2014 – June 08\(^{th}\) 2014
- Transaction ST03N
  - Aggregated statistical workload data for May 2014
- Transaction TAANA
  - Table analysis
- Transaction ST13,
  - PERFTOOLS: ABAPmeter to show pure ABAP and DB access performance
- ST04
  - Analysis of DB KPIs and Oracle Cache
- Other transactions

Remarks

- This slide describes on what information the analysis is based on.
Introduction / Summary / Action List

Technical Landscape

Technical Workload Analysis

Application Workload Analysis

Scalability Analysis

Remarks
- In case of delta sizing the system landscape is analyzed in order to find the capacity that is currently deployed.
- We need to know the overall system capacity in order to check if a future workload increase can be handled by the system.
- Further it makes us aware of certain dependencies, e.g. to other systems because of virtualization.
- Makes us aware of heterogeneous hardware / OS landscapes
Assumed capacity

- Xeon E5-4650 (2.7 GHz) based systems: ~2100 SAPS / core
- Power 7 (3.724 GHz) based system: ~3000 SAPS / core

Deployed capacities:

- Application hosts: 67,200 SAPS  
  (4 hosts * 8 cores * 2.100 SAPS)
- Database host: 21,000 SAPS guaranteed capacity  
  (7 cores * 3.000 SAPS)
  48,000 SAPS maximum capacity - VCPUs (16 cores * 3.000 SAPS)
The following slides show the hardware utilization. It makes us aware of:
- Capacity used
- Time patterns
- Free capacity
- Bottleneck situations

A baseline is created for the upcoming sizing calculation.

The way of analysis might differ based on the HW / OS architecture (e.g. in case of virtualization)
Hardware utilization

CPU utilization – application hosts

- Average CPU consumption on application hosts ~30%
- Only few peaks
- Workload seem to be distributed evenly across application hosts
Hardware utilization
CPU consumption – DB host

- Entitlement is sufficient for the current workload on the DB host.
- Only few peaks above the entitlement of 7 CPUs – Entitlement is correctly set
- Daily pattern with highest peaks times at 11 a.m. – 3 p.m.

CPU consumption of host DBhost

<table>
<thead>
<tr>
<th>Date</th>
<th>CPU Consumed</th>
<th>Entitlement</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.05.2014</td>
<td>00:00</td>
<td>7</td>
</tr>
<tr>
<td>27.05.2014</td>
<td>00:00</td>
<td>7</td>
</tr>
<tr>
<td>28.05.2014</td>
<td>00:00</td>
<td>7</td>
</tr>
<tr>
<td>29.05.2014</td>
<td>00:00</td>
<td>7</td>
</tr>
<tr>
<td>30.05.2014</td>
<td>00:00</td>
<td>7</td>
</tr>
<tr>
<td>31.05.2014</td>
<td>00:00</td>
<td>7</td>
</tr>
</tbody>
</table>

CPU pattern for CW 23

© 2014 SAP SE or an SAP affiliate company. All rights reserved.
Customer 16
Hardware utilization

CPU consumption – hardware of DB host (ABC)

- Physical CPU idle shows a low value during peak times. Peaks are caused by the sum of CPU consumption of all LPARs running in this shared pool.
- DB host (= our LPAR) competes for CPU resources with other hosts (LPARs) that belong to other applications.
- Shared pool is utilized at optimum.
- Further load will lead to performance issues because CPU resources are already restricted during peak times.
- If workload increases:
  - Either the LPAR has to be moved to another Shared pool with sufficient CPU resources.
  - Or the current shared pool is released from workload by moving other LPARs.

![Graph showing CPU utilization over time](image-url)

**Key Points**

- **CPU Consumption**
  - Physical CPUs Consumed
  - CPUs consumed by Other hosts
  - Physical CPU idle
Total CPU consumption
Application hosts

- Deployed capacity: 67.200 SAPS
- Average peak: ~18,000 SAPS
- Maximum peak: ~27,000 SAPS
- We see potential for a load increase. The average utilization is at ~27%. A value of 65% is considered to be the optimum to handle also the maximum peaks.
Total CPU consumption
Database host

- Entitled Capacity = Maximum peaks: 21.000 SAPS
- Maximum Capacity: 48.000 SAPS (would be available if no other LPAR would be operated on the shared pool)
- Entitlement meets the maximum peak very closely
- Average peak ~ 17.500 SAPS
Technical Workload Analysis – Baseline Conclusion

- Maximum allocated extended memory is around 55 GB.
- We see a daily pattern.
- Pattern corresponds to the CPU pattern with a peak around noon.
• The average peak consumption of the application hosts is around 18,000 SAPS. The average peak consumption on the DB host 17,500 SAPS. The normal relation Application hosts : DB hosts should be 2:1 or even 3:1. This points to a high load of the database which can have an effect on the scalability of the DB.
• The application hosts show a lot of free capacities.
• The DB host competes with other LPARs on the same resources of the shared pool. There is no space on shared pool level for load increases without reducing workload on the shared pool by moving LPARs.
• The highest peak is around the time of 11 a.m. to 3 p.m. It is a solid peak what does not necessarily point to dialog activity.
Now that we know what capacity the system has and what capacity is consumed we need to find out the sizing relevant activities in the system. Therefore we analyze the workload drivers in a top-down-approach.

- At first we check the distribution of workload across task types and the continuity of the workload.
- Then the workload is investigated by task type
- At the end we know
  - Which transactions / reports are the main load driver in the system and therefore are relevant for sizing
  - Transaction / reports which should not appear in the top consumers as they don’t belong to the value chain of the customer’s business.
Application Workload Overview

- Major workload driver from application point of view is background activity followed by dialog and RFC.

- The increase of the workload is a little bit higher than the number of dialog steps would indicate. The database time is growing constantly which might be related to data growth.

- High portion of DB time compared to CPU time.

Workload by task type

- BACKGROUND: 51%
- DIALOG: 35%
- RFC: 10%
- UPDATE: 3%
- UPDATE2: 1%

Workload history

- Total CPU time (h)
- Total DB time (h)
- Number of dialog steps

- Month

- Number of dialog steps

- Total CPU time (h)
- Total DB time (h)
- Number of dialog steps

© 2014 SAP SE or an SAP affiliate company. All rights reserved.
Background Workload

- Top 20 background jobs contribute 73.8% of the total background workload.
- Some reports are directly or indirectly related to the creation of deliveries what seems to be the major load driver.
- We see relatively high amount of DB times
- To be clarified: High amount of spool activities (ZBE1SD001O01 and RSTS0020)
- Example Customer Inc. stated that users are allowed to execute expensive queries on the system. This should be avoided as well.
We tried to put the CPU consumption in sync with the background workload as it is the highest workload driver. Inside the top 20 background reports we could see a lot of delivery related reports. So we checked the times when the documents are created.

- Deliveries are created during day time (11 a.m. - 3 p.m.)
- Highest peaks are between 1 p.m. and 3 p.m.
- The same behavior can be seen on all working days of May.
- This corresponds to the peaks in CPU consumption. So the creation of deliveries is the major workload driver and has the highest relevance for sizing.
Dialog workload

- Top 20 dialog transactions contribute 74.4% of the total dialog workload.
- Top 2 transactions ZRU1SD01106 and VL10A contribute 30.1%. These transactions are also directly or indirectly related to delivery creation.
- In general we see also high DB times.

<table>
<thead>
<tr>
<th>Transaction code</th>
<th>Transaction name</th>
<th>Transaction Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZRU1SD01106</td>
<td>Russia - Order Consistency Check</td>
<td></td>
</tr>
<tr>
<td>VL10A</td>
<td>Sales Orders Due for Delivery</td>
<td></td>
</tr>
<tr>
<td>VA01</td>
<td>Create Sales Order</td>
<td></td>
</tr>
<tr>
<td>SESSION_MANAGER</td>
<td>Session Manager Menu Tree Display</td>
<td></td>
</tr>
<tr>
<td>VL060</td>
<td>Outbound Delivery Monitor</td>
<td></td>
</tr>
<tr>
<td>ZPL1MM357R1</td>
<td>Stock Overview w/ S Loc. display</td>
<td></td>
</tr>
<tr>
<td>ZVT70</td>
<td>Output for shipments</td>
<td></td>
</tr>
<tr>
<td>ZCORIN5201</td>
<td>Stock overview</td>
<td></td>
</tr>
<tr>
<td>ZRU1SD02790</td>
<td>Extract orders from SAP to IDOC</td>
<td></td>
</tr>
<tr>
<td>BD87</td>
<td>Status Monitor for ALE Messages</td>
<td></td>
</tr>
<tr>
<td>VA02</td>
<td>Change Sales Order</td>
<td></td>
</tr>
<tr>
<td>ZCORMM35206</td>
<td>Material Document List</td>
<td></td>
</tr>
<tr>
<td>FBL3N</td>
<td>G/L Account Line Items</td>
<td></td>
</tr>
<tr>
<td>VT70</td>
<td>Output for Shipments</td>
<td></td>
</tr>
<tr>
<td>VL01N</td>
<td>Create Outbound Div. with Order Ref.</td>
<td></td>
</tr>
<tr>
<td>VKM1</td>
<td>Blocked SD Documents</td>
<td></td>
</tr>
<tr>
<td>ZJP1SD423</td>
<td>Japan Order Flow Consistency Report</td>
<td></td>
</tr>
<tr>
<td>FBL5N</td>
<td>Customer Line Items</td>
<td></td>
</tr>
<tr>
<td>FBL1N</td>
<td>Vendor Line Items</td>
<td></td>
</tr>
<tr>
<td>VA03</td>
<td>Display Sales Order</td>
<td></td>
</tr>
</tbody>
</table>

Average response times

- Total CPU and DB time (h)
- Contribution to dialog workload (%)

Average response times

- Average DB time (ms)
- Average CPU time (ms)
RFC workload

Top 20 RFC Users by Contribution

- Top 20 RFC users show 91.7% of contribution to RFC workload
- Top 4 users show 55.9%
- At this point we are not able to drill deeper because of limitation of our tools (ST03N)
Application workload analysis - Conclusion

- The major workload driver is background and dialog activity in the system.
- We saw a lot of delivery related transactions and reports causing the major workload.
- We saw that the creation of deliveries corresponds to the peaks of CPU consumption. So it is the major workload driver which is relevant for sizing is the creation of deliveries.
The scalability analysis is done to find obvious non-linear scalabilities in the system.

These can happen most likely on the database related to I/O.

The most obvious are expensive select statements in the SQL cache. They do not scale linear because their execution time is depended from more than 1 parameter (e.g. full table scan: data records in table and number of executions).

Some technical parameters are checked as well.

A scalability analysis can be done in several levels which are dependent from the necessity of such an analysis. In this example report we have a very highest level of analysis which has the focus of avoid unnecessary workload from the DB.

The lowest level would be only a rough analysis of the top 20 statements in order to check if they would scale linear. As a subsequent action a TPO for DB might be recommended.
Database KPIs

- Buffer quality = 99.49% (very good with regards to frequency of accesses)
- Based on an ERP system the Physical Reads should be lower than the Synchronous Reads. We see a high number of physical reads pointing to a lot of data read sequentially from the database.

- We recommend to set following parameters.

<table>
<thead>
<tr>
<th>Parameter Type</th>
<th>Parameter</th>
<th>Parameter value</th>
<th>Recommended value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>PCKCACHESZ</td>
<td>131072</td>
<td>=AUTOMATIC</td>
</tr>
<tr>
<td>REG</td>
<td>DB2_HASH_JOIN</td>
<td>YES</td>
<td>NOT SET</td>
</tr>
<tr>
<td>REG</td>
<td>DB2_MEDIUM_PAGE_SUPPORT</td>
<td>OFF</td>
<td>NOT SET</td>
</tr>
<tr>
<td>BP</td>
<td>BUFFSD1D</td>
<td>65536</td>
<td>=AUTOMATIC</td>
</tr>
</tbody>
</table>
The SQL cache to the left shows the top 18 SQL statements sorted by total elapsed time observed on the 16th of June.

The statements ranged 1st, 6th, 10th and 16th are issued by F_GET_DELIVERY_DATA in include ZXXXX

In sum they are responsible for 24.7% of the DB time recorded in the cursor cache.

A possible fix is described in the following slides. It is highly recommended to initiate the code changes as soon as possible.
Unnecessary load by F_GET DELIVERY_DATA

Problem description

- F_GET_DELIVERY_DATA in ZXXXX contains unnecessary selects to MCH1 and T005T that are responsible for ~ 25% of the overall time recorded in the SQL-cache.
- A rough check of internal table I_MCH1 filled with the select from MCH1 shows no usage beyond the use in the select from T005T.
- With the huge number of records returned the select form MCH1 is already very expensive, but the real problem is the use of the result list in the select from T005T.
- With the many rows returned the select for all entries is split into many individual statements with many duplicate entries for I_MCH1-HERKL.
- As the select bypasses the buffer and forces the DB to do a sort and write to temporary table to guarantee a "select distinct" the cost to the DB is very high.
- All of this is done to pre-fetch records from T005T, a table that is buffered on the applications servers and does not need to be read from the DB at all.
Unnecessary load by F_GET_DELIVERY_DATA

Recommendations

- Delete all of the coding shown at the left (up to the ENDFORM)
- Delete the additional select from T005T in f_get_delivery (lines 2217-2229 and lines 2254-2267) (What ever is read there is overwritten by the subsequent selects anyhow.
- Replace all read table READ TABLE I_T005T (only in form f_get_country_origin ?) by selects from table T005T. As the table is buffered generically with 2 key-fields the select will read from an application server buffer.
- Replace all read table READ TABLE I_T005 (is there any?) by selects from table T005T As the table is fully buffered the select will read from an application server buffer.
Now we have a clear picture about the usage of the available system resources and scalability. This enables us to state about the future resource utilization.

The way to calculate the resources are quite different. This way and the result depends very much from the customer input. The calculated values are then check against the experience. The results are discussed and assessed.

In this example report, the customer was very “rough” in his estimation. The result was therefore not very precise and possibly to high. However, the result is not given to the customer without any comment on this.

In this example we only state about the ability of the current hardware to handle the very next workload increase, but not the overall workload increase as the point in time is not appropriate to do this as the result will be most likely far too high.
Delta Sizing for Example Customer Inc. – <SID>
Sum of Building Blocks

For the SAP-based sizing projection for <SID> load increase we used the below building blocks which can be of course adapted.

- Target utilization: 65% Target utilization
  - AS ABAP: ~ 305,000 SAPS
  - DB: ~ 296,000 SAPS

- Additional business volumes: Factor 10 (given by Example Customer Inc.)
  - AS ABAP: 198,000 SAPS
  - DB: 192,500 SAPS

- Business Growth: We assume 10% growth for one year. Factor: 1.1
  - AS ABAP: 19,800 SAPS
  - DB: 19,250 SAPS

- Current resource consumption: Measured KPIs: Average CPU peak in SAPS (pages 13,14)
  - AS ABAP: 18,000 SAPS
  - DB: 17,500 SAPS
Sizing results – discussion and approaches

Based on the findings about the scalability we estimate that the load increase by the first wave can be eliminated by reducing the unnecessary load in the programs mentioned in the scalability analysis.

Taking a sizing factor of 10 into account we would get an estimation of ca. 300.000 SAPS for each, DB and application. This seem far to high. Calculating a factor of 10 will also multiply inaccuracies. At this moment in time, a forecast is not feasible due to a lack of preciseness and missing optimizations. The current optimization potentials can have a big impact on the current consumption. For an example: If the DB load can be reduced by 30% based on the recommendations we gave the basis for our calculation would change. The result would be 30% lower what means 90.000 SAPS on the DB server what is significant.

We therefore recommend to go step by step, i.e. wave by wave and check how the system resource consumption responds to the business volume increase. Further optimizations are a good way to reduce the future hardware demand. We assume that the demand of hardware will be much lower than the current forecast.

We can support this process with further checks of the hardware consumption related to the application workload with the scope of reducing unnecessary system workload.