Oracle Applications Capacity Planning for Middle Tier Servers

By Devanand Nayak, DSP Global Limited

Purpose

Capacity planning tends to lie at two extremes. On the one hand we have the approach that takes months and months and uses complex mathematical models, simulation and the like. On the other hand we have the approach which takes minutes and involves thinking how things were in the past and multiplying it by some fudge factor. The first approach is usually beyond the budget of both time and money of most projects that you and I work on. The second approach usually leaves us with our fingers crossed.

Surely there must be some middle ground that a person faced with preparing a capacity plan can make. This paper sets out to achieve just this goal.

Capacity Planning

When we talk about capacity it must always be with reference to some workload. For example, how many CPUs (the capacity) do we need to support 500 users (the workload). In essence capacity is a function of the workload. Capacity = F(Workload). Thus, in practical terms capacity planning requires a formula, no matter how simple or complex, that predicts the capacity required for a defined workload. For example, the formula below is an example of a simple equation that could be used to estimate memory requirements:

```
Required Memory =
      Number of Oracle Users *
      MB per person +
      some overhead
```

The skill in developing this formula lies in not just how accurate it is, but also in other factors such as how easy is it to use. Thus, formulas such as the one illustrated are widespread in the systems sizing sections of product manuals not just because they are accurate, but as important, because they are easy to calculate, and that is no bad thing.

This paper will discuss a method of deriving the capacity planning equation from an existing system that is both easy to use and accurate. An obvious prerequisite is that you must already have an existing system. This could either be your live system, another customers live system or some simulation/test system.

Once we have established the capacity planning formula it can be used to predict the required capacity for defined workloads. This is useful for not only deciding on future hardware upgrades as the system grows, but also for predicting what the minimum upgrade needs to be to handle an existing workload that is performing badly.

Method

The following is the outline of the capacity planning method and also provides the outline for the paper:

Measure Current Utilization of Subsystem (E.g. Memory) Under Study

The choice of metrics to be collected will be considered and what tools are available to measure them. In particular the hp-ux measureware suite will be mentioned, but other tools such as vmstat and sar will also be dealt with.

Decide if Current Capacity is Enough

The presentation will discuss how to analyse the metrics collected in step one and determine if the subsystem configuration is adequate. That is, do we currently have a problem?

Characterise the Workload

In order to specify how much memory is needed or how many CPUs are required an understanding of the workload which must be supported is required. A very simple method for determining the current workload for Oracle Applications is presented. Furthermore, a method of measuring the workload employing standard Oracle Applications auditing functionality is considered.

Build the Capacity Planning Model

The purpose of a capacity planning model is to establish the relationship between the workload and the capacity required. The type of model that is presented uses Linear Regression techniques, but, relies on Microsoft Excel to work out all of the mathematics. No deeper mathematical knowledge than constructing graphs with Excel is required.

Use Model to Predict Capacity

The model constructed can now be used to predict the required capacity for future workloads, and/or, predict what the minimum upgrade needs to be to handle the existing workload.

Where the Model Fits

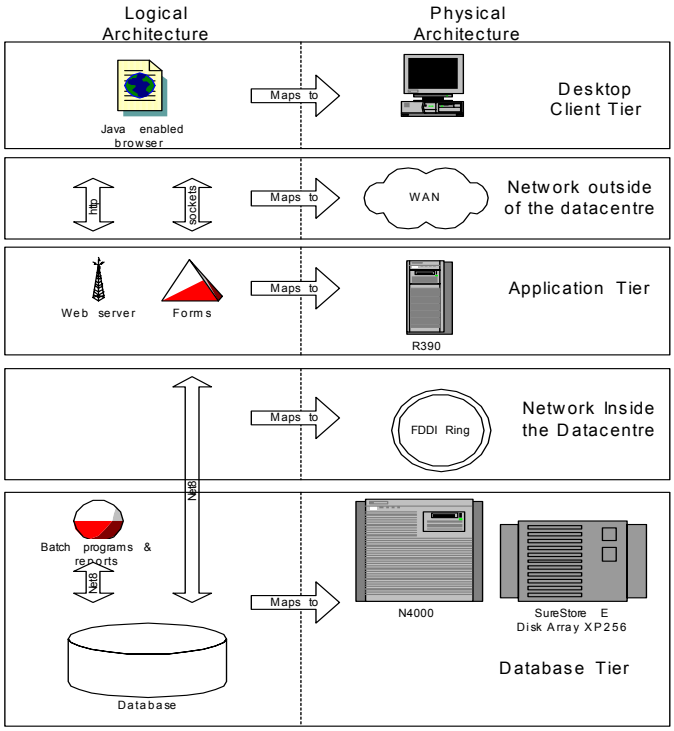
Why Middle Tier Servers?

The paper is called "Oracle Applications Capacity Planning for Middle Tier Servers" because it is my experience that the method works best on middle tier servers. The reason I give for this is a middle tier server is usually dedicated to one task, i.e. serving forms, this fits the linear models prescribed in this paper better than a server performing multiple tasks, e.g. running a database server and concurrent managers. This does not mean you should not use this method on other types of servers, it just means I know it works well on middle tier servers. As you will see by working through this paper it is very straightforward to apply this method to any type of server, and you will discover quickly if the model is indeed linear or not. Also, there is a lot to be learned regarding the performance and workload of your server by using this method even if a linear model turns out not to fit too well.

Case Study

I have used this method to predict cpu and memory requirements for various Oracle Applications workloads. In general linear formulas work much better predicting memory than cpu. Nevertheless, the method has been used to predict cpu with good success.

Throughout the paper a case study will be used to illustrate the method. The case study is of a HP R-Class server running hp-ux 11. The server is used as the Oracle Applications Forms server of a 3 tiered Oracle Applications system architecture, which is shown below:



Release 11.0.3 of Oracle Applications was installed. The following table lists the Oracle technology stack contained on the server:

Purpose	Applications Logic	Oracle Home Contents	Appl Top Contents
Forms server	Forms	Oracle Application Server 3.02 Oracle Designer Server 1.6.1 (Forms 4.10.10.x) Oracle8 Enterprise Server 8.0.5	Oracle Application release 11.0.3 forms

The R-class server was short of memory and was causing performance degradation due to memory paging to cope with the memory pressure.

The current memory configuration was 1.25GB and the server was supporting up to 100 users on peak days. The capacity planning method was used to determine what the necessary memory upgrade should be to support not just the current workload, but growth to 150 users.

Measure Current Performance

Choosing a tool to gather metrics.

The first step in the method is to establish a baseline of the current performance. There are numerous tools available for gathering the metrics required, e.g

sar

vmstat

hp-ux Measureware extract

Other 3rd party tools

As usual the tool of choice will be limited by what is currently installed on the platform and by what tools you currently have expertise in. Most of my capacity planning work has been done on hp-ux platforms, and where available the extract utility that forms part of the Measure suite of tools, has proven easy to use, will gather almost every metric you may desire and is very flexible. Also, in general measureware is installed on most hp-ux boxes. I use sar on Solaris platforms as it is free and gathers most metrics.

Features to look for when choosing a tool to gather metrics are:

It gathers the metrics you need. E.g., The sar tool does not measure memory utilization on the hp-ux platform.

It is designed to gather historical data. Tools such as top and glance are not designed to gather historical data, unlike sar and measureware.

It can easily be set up to gather stats over a long period of time. For instance, sar can be configured in a standard way to gather statistics continuously from boot time. Scripts are provided with most operating systems to take care of purging old data, etc. Whereas, the vmstat tool does not have any of these handy scripts available.

Current Performance Analysis

Preparing the data.

In order to gain an understanding of the metrics that are collected it is useful to plot them in Microsoft Excel. The data can be loaded into Excel by preparing the collected metrics into comma separated value (CSV) files. Nevertheless, being a DBA I prefer to load the metrics first into an Oracle database. The metrics can then be pulled from Oracle into Excel via ODBC. I find this method more organized and flexible over maintaining CSV files.

Whichever tool is being used to gather the OS metrics, the data should be collected into flat files that can be uploaded into Oracle using SQL*Loader. This is straightforward using the extract utility, on the other hand, when sar is used the output will need to be formatted using awk. This is another consideration when choosing a collection tool.

Which metrics

One of the best qualities of the utilization metric is that even as a single number it conveys meaning. Nevertheless, a single number can never tell the whole story. Usually utilization along with the queue length should give a sound basis for determining whether performance can be improved by additional capacity. Utilization and queue length can be readily measured for CPU and Disk subsystems, but not for memory.

In our case study the memory performance of the server was measured using data collected by MeasureWare Agent software from HP. This tool does report a queue length, however, in our case study the paging statistics are used. The following metrics were collected at ten minute time intervals over a period of a day:

Total Memory Utilization

User Memory Utilization

Page Out Rate (pages/second)

Visualization

Spreadsheets are not just good at creating time sheets, but are very powerful data analysis tools. In our case study Microsoft Excel was used to visualize and analyze the data. As previously discussed the following process was used to view the data in Excel:

Use MeasureWare extract tool to capture data into ASCII flat files.

Use Oracle SQL*Loader to load the flat files into a database schema.

Load data into Microsoft Excel via ODBC.

Plot the data in Excel.

The memory utilization over the period of a day is shown for the server in the following chart (figure 1):

The X-axis plots the time of day. The Y-axis on the left measures the memory utilization and Y-axis on the right shows the page out rate. An examination of the graph clearly shows that when memory is 100% utilized the memory subsystem starts paging to cope with demands. The system does not have enough memory to prevent paging.

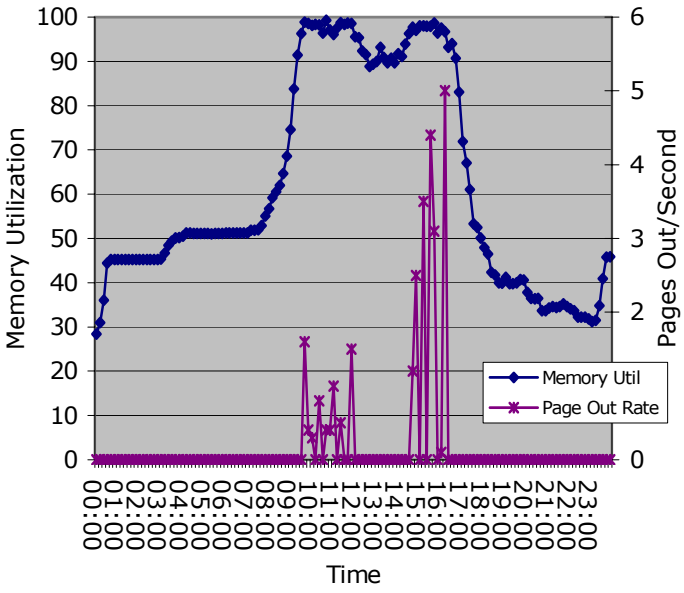


Figure 1 – Memory Utilization and Page Out Rate against Time

The result of the wasted disk IO and CPU resources that are used paging is performance degradation. Thus, memory needs to be expanded to cope with the existing workload. The additional memory required will be calculated later in this paper. First we will examine what the existing workload is.

Workload Characterisation

Defining the Workload

In order to plan or predict capacity requirements, its relationship to the workload must be established. When defining the workload it is always tempting to go into lots of detail usually with the goal of being as accurate as possible. This is always a mistake. Define your workload as simply as possible. Do this for 2 reasons, firstly, before getting bogged down into complicated detail prototype a simple model, you can add more detail if you need later. Second, capacity plans usually end up before decision-makers (a.k.a. Managers) and if you want the plan to be read, let alone understood the simpler the better.

In our case study the server acts as a forms server. Initially, it may seem that a "form" should be used to quantify the workload, however, I chose an "Oracle Application User" instead, as this unit is more easily understood by anyone and can be predicted more readily by management.

Measuring the Workload

The standard Oracle Applications auditing feature can be used to measure the number of Oracle users logged in.

To configure Applications auditing the site level profile option "Sign-On:Audit Level" must be set. This option is usually set to allow real time monitoring of Oracle Applications Users via the Security\User\Monitor form within Oracle Applications. Nevertheless, the data that is collected is held in the FND tables and there are a number of standard concurrent reports that can be run that give historical information.

The profile option can be set to the following values: User, Responsibility and Form. Despite only using "User" level data, I normally set the profile option to Form. This allows finer grained real

time monitoring and also gives me the option to characterize the workload in further detail at a later time.

Joining the workload and performance data

The workload data must be tied to the performance data. This is done using date and time. The performance data was collected at ten-minute intervals aligned to the hour, e.g. 0000, 0010, 0020, etc over 24 hours. Thus, in order to make use of our workload data we must collect our user count data for these exact same time intervals.

As the workload data is already stored in tables in the APPLSYS schema, code can readily be developed to aggregate the data appropriately. A summary table should be created to store the data in the format required to join it to the performance data. The workload and performance tables can then be directly joined using date and time as the join key.

In order, to move further processing away from the live database and server these tables should exist in a separate database on a development server.

This workload data is now ready to be loaded into the spreadsheet for analysis.

Workload Analysis

We now have both the performance and workload data tied together by time interval and conveniently stored in an Oracle database.

Figure 2 is a duplicate of figure1, but this time with the number of Oracle Users overlaid on it:

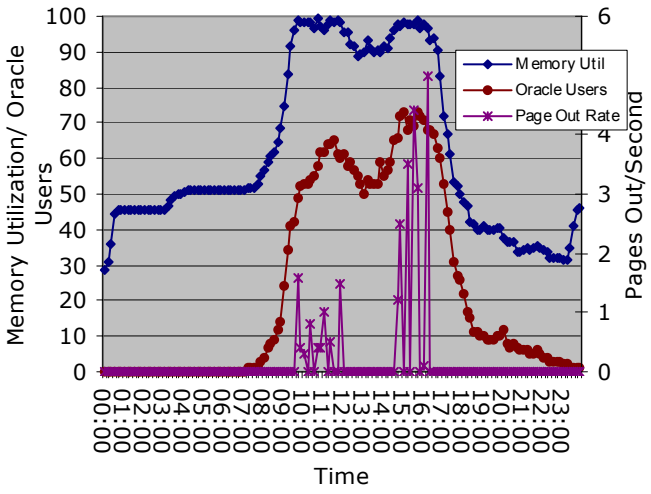


Figure 2 – Memory Utilisation, Page Out Rate and Users against Time

As before, the X-axis plots the time of day. The Y-axis on the left measures the memory utilization and now also the number of Oracle Users logged in. The Y-axis on the right shows the page out rate.

This graph shows that despite the memory utilization hitting 100%, more users keep on logging into the system. The additional memory required for these users is thus made available by the memory subsystem by paging memory to disk.

The shape of the Oracle Users plot shows a mountain range with 2 peaks. Contrast this shape to the profile of the memory utilization plot, which has its peaks clipped off where the utilization is limited at 100%. It is interesting to note, that the memory utilization peaks have been replaced by page out activity. In order to determine the required memory utilization we need to restore the peaks that should be on the memory utilization mountain range.

Memory Model

Establishing the Relationship between Oracle Users and Memory

Figure 2 is repeated in figure 3, but this time without the page out detail.

As expected the graph shows that as the number of Oracle Users increases or decreases so does the memory utilization. There is a definite relationship between the number of Oracle Users and the Memory Utilization, but what is that relationship? Furthermore, if we knew how they are related we could use that relationship to predict what the memory would be for a given number of users.

As with everything we have done so far lets visualize the relationship to gain a deeper understanding. The chart in figure 4 plots memory utilization against Oracle Users.

One month's MeasureWare data was used as input for the model. The additional data helped the accuracy of the model, but as used in all the other charts, a day period would have been sufficient.

The X-axis plots the number of users and the Y-axis measures the memory utilization. The brown/top line is the Total Memory Utilization and the pink/lower line is the User Memory Utilization. User memory is memory that is used to store user processes, it does not include kernel and buffer cache memory.

The chart shows that as the number of Oracle Users increases the user memory utilization increases proportionally plotting a straight line, thus, demonstrating a clear linear relationship. The user memory flattens out only when the total memory utilization hits 100%, however, if the total memory was not exhausted then it is reasonable to expect user memory utilization to continue upward in a linear fashion. This expectation is marked on the chart by the black straight line (figure 4).

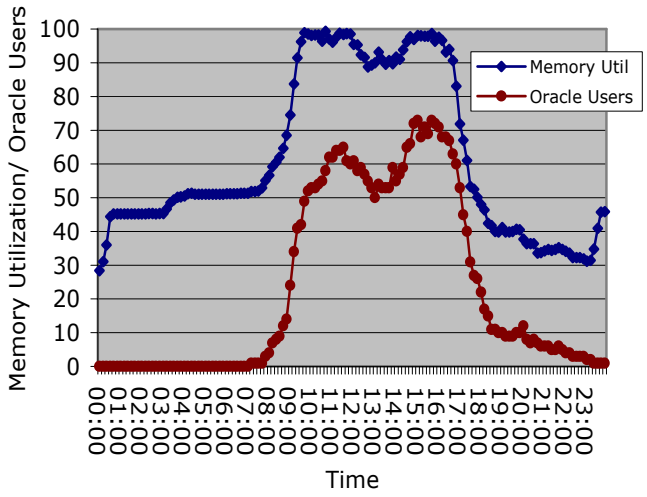


Figure 3 – Memory Utilisation and Users against Time

The black straight line was added to the chart in figure 4 by:

selecting the User Memory Utilization plot line

clicking the right mouse button

choosing add trend line

The linear formula for this trend line can also be calculated by Excel and is shown figure 4. This formula is the relationship between Oracle users and memory utilization and is repeated below:

```
User Memory Utilization =
    0.97 * Oracle Users + 18.4
```

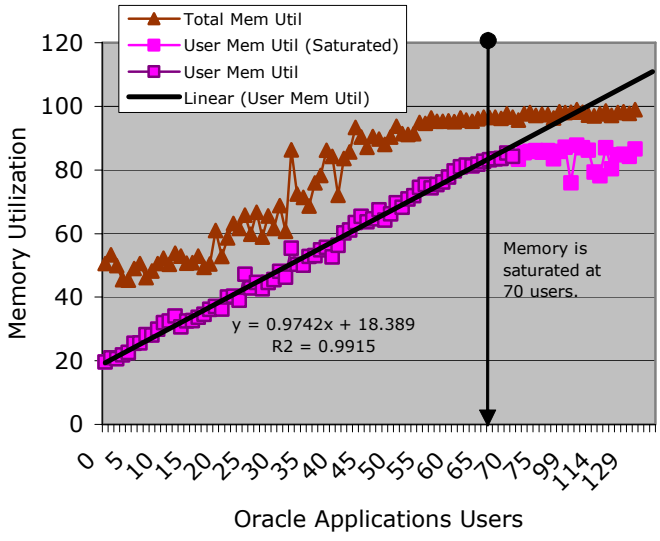


Figure 4 – Memory Utilisation against Oracle Applications Users

Note, that when calculating the trend line only data up to 70 oracle users was used as indicated on the chart by the black arrow. The reason for this is because memory utilization measurements beyond the arrow are skewed due to memory saturation. That is, no matter what the value of Oracle users it is impossible for the value of Total Memory Utilization to exceed 100%.

The formula derived above is the formula needed to predict user memory utilization. In order to calculate total utilization we just need to add the memory required for the kernel and buffer cache. This can be deduced easily from the difference between total memory utilization and user memory utilization and is calculated to be 26.1%.

Thus, the formula to calculate the total memory utilization is:

```
Total Memory Utilization =
    0.97 * Oracle Users + 18.4 +26.1
```

A formula that calculates gigabytes rather than percentages is preferred, and is shown below:

```
Total Memory GB =
    (0.97 * Oracle Users + 44.5)/100 * Server Memory
```

As server memory is 1.25GB, the formula to predict server memory in GB is as follows:

```
Total Memory GB =
    (0.97 * Oracle Users + 44.5)/100 * 1.25
```

Predicting memory requirements

Memory required for current workload

This formula can now be used to predict memory requirements for the current and future workloads.

In the performance analysis of the server in our case study it was seen that more memory was required to prevent paging. How much more memory is required?

In figure 1, the maximum number of users that were logged in on that day was 72. Thus, by substituting 72 into the formula we can predict how much memory is really required:

```
Total Memory GB =
    (0.97 * Oracle Users + 44.5)/100 * 1.25

1.45 GB =
    (0.97 * 72 + 44.5)/100 * 1.25
```

Hence, the memory requirements for a workload of 72 users is 1.45GB.

The formula can be used to re-plot figure 2. Notice the peaks that were missing on the previous figures have been restored to the profile of the memory utilization plot in figure 5.

Memory Requirements for the Future Workload

As part of the case study it was also required to work out how much memory would be required to support 150 users. The formula can be used to answer as follows:

Total Memory GB = (0.97 * Oracle Users + 44.5)/100 * 1.25

2.38 GB = (0.97 * 150 + 44.5)/100 * 1.25

Scalability Ceiling

It is also interesting to calculate the maximum number of users that the server will support. The HP R-class server will scale to a maximum memory configuration of 3GB. According to our model a memory configuration of 3GB will support 200 users.

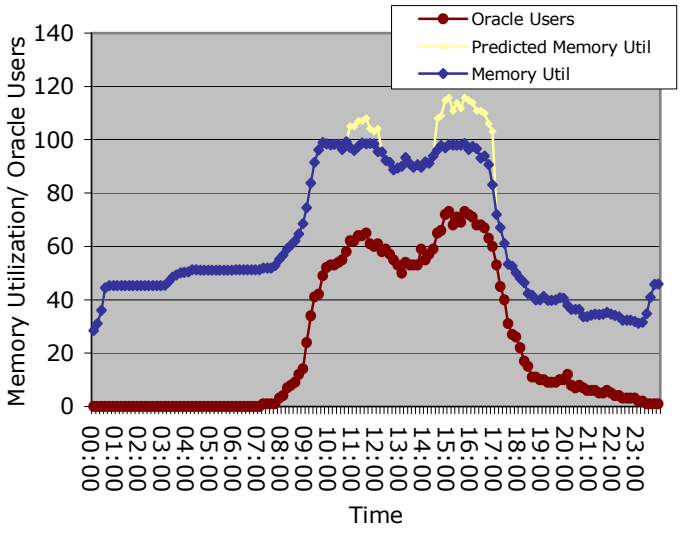


Figure 5 – Predicted Memory Utilisation Against Time

Conclusion

Capacity planning forms part of every DBAs role. The ethos of this paper was to focus on practical techniques that can be used on the projects that every DBA works on.

Before making it to the end of this paper, perhaps the idea of using linear regression to predict capacity might not have seem practical. It is worth noting that the only time the term linear regression was mentioned was in the introduction and the conclusion. The main content of the paper, apart from illustrating how to draw Excel graphs was simply applying common sense.

References

The Art of Computer Systems Performance Analysis, Raj Jain.

Simple Regression Analysis Template, www.orapub.com, Craig Shallahamer

About the Author

Dev Nayak works as a System Architect and Oracle DBA for DSP Global Ltd and can be contacted on dnayak@dsp-global.com. DSP Global Ltd provide the Oracle DBA and E-Business suite support services **DBA***direct* and **APPS***direct* and also provide project consultancy services.

A: DSP Global Limited, Vigilant House, 120 Wilton Road, London, SW1V 1JZ
T: +44 (0) 20 7808 7052
F: +44 (0) 87 0122 5763
W: www.dsp-global.com