By: Karl Arao

This is the general methodology we use for consolidation and sizing

Diagram

Description automatically generated

I have a 15 page write up here that details on how to execute a consolidation and sizing exercise <https://github.com/karlarao/sizing_worksheet/blob/master/Consolidation%20and%20Resource%20Management.pdf>

You can follow the example using the sizing worksheet Randy J. and I created <https://github.com/karlarao/sizing_worksheet/blob/master/sizing_worksheet.xlsm>

I still use that sheet for consolidating < 50 databases. But for modeling 100s or 500+ of databases I would use our internal web app (ESP – Enkitec Sizing and Provisioning) which is the apex version of the worksheet (created by Carlos S., Christoph R., Mauro P., Frits H., me, Randy J., Jorge B., and others) mainly to automate the balancing of the instances across nodes (let’s say keeping each node at <30% utilization) and creation of sizing scenarios across different hardware make/model or cloud environments (OCI, AWS, GCP, Azure, VMware, etc.).

Before the ESP web app and before Oracle productized the methodology into “EMGC Consolidation Planner” <https://docs.oracle.com/cd/E24628_01/doc.121/e28814/consolid_plan.htm#EMCLO966>

This was around Exadata V2 and X2 era (my doodle below says 9 years ago), we were getting consolidation requirements moving as much as 400+ Peoplesoft databases on a half rack. We had to show to the customer that DB requirements need to fit the hardware capacity and modeling the consolidation using real workload numbers is the only way (no guessing or guestimates). The collectors and sizing tool were refined over and over (70+ sizing engagements before the web app was created). And based on the consolidation experiences we also refined the methodology.

The latest addition was projecting of the headroom expiration date (forecast) which can be statistically modeled using the time series data (this came up from a DBaaS project for capacity planning their on-prem private cloud). <https://github.com/karlarao/forecast_examples/tree/master/monte_carlo>

Diagram, schematic

Description automatically generated

The collection tool ESP uses is this <https://github.com/carlos-sierra/esp_collect> , it outputs a csv file w/ focus on resource requirements. ESP is also collected by EDB360.

What I use is this <https://github.com/karlarao/run_awr-quickextract> which also includes ESP and other workload characterization info (SQL, ASH dump, CDB/PDB calculated fields I use in tableau, etc.) that would allow me to break down the SQL workload drivers by dimensions (parsing schema, service\_names, etc.). Over the years I have built template Tableau dashboards that would just refresh based on new data collection. And I use this dataset heavily for performance troubleshooting.

I’ve got a few key points on sizing:

1. Use of percentile on consolidated time series data
2. Size based on A) regular season and B) peak seasons of the year (Mother’s Day and Christmas)
3. More data = more accurate sizing and allows demand forecasting
4. **Use of percentile on the consolidated time series data of the databases**

If you just combine all the max or average numbers of all the databases without considering the stacked time series, you may end up w/ an over provisioned on-prem environment.

Here’s an example:

* Here are the four databases on 2 node Exadata cluster X6-2
* The chart is a cluster wide view of CPU usage based on ASH data (filtered by CPU + CPU Wait)
  + The chart says “Minute” but this is how it will also appear if sliced by “Second”. Check my ASH granularity math investigation here <https://karlarao.github.io/karlaraowiki/index.html#%5B%5BASH%20granularity%20math%5D%5D>
  + The important thing here I’m using ASH data vs wide AWR numbers. ASH data is what we use in ESP for CPU sizing.

Given the workload data, the CPU requirement I would use as input on my sizing based on 95th percentile is 50 CPUs or 28% CPU utilization (50/176).

If I use the December 15th peak of all peaks, then I’ll end up with 90 CPUs or 51% CPU Utilization (90/176). I would make an exception and use this data IF after workload qualification I found out that the SQLs running during this period are not adhoc SQL\*Developer sessions.

Graphical user interface

Description automatically generated

The beauty of doing the sizing manually using an exploratory UI (Tableau) is I’m flexible when it comes to filtering out the time slices not related to the actual database workload or organic growth.

We can’t do this data filtering right now in ESP but the percentile number that will be used on the data set can be selected.

Table

Description automatically generated

1. **Size based on A) regular season and B) peak seasons of the year (Mother’s Day and Christmas)**

IF the data is available, I would size based on non-peak and peak seasons of workload. That way you know how much the hardware resource consumption swings when the business is at its peak and how much hardware you need to allocate for it.

From experience there are a few dates where the workload peaks:

* Mother’s Day
* Christmas
* Black Friday
* Month-end, Quarter-end, Year-end processing
* Industry or company specific events

The data I’ve shown on the CPU percentile example above is a cross point of regular processing period and Christmas season peak. Knowing that during Christmas peak the workload increases by 30% (the SQL workload drivers correlate to the CPU utilization) is vital information for sizing.

Chart

Description automatically generated

The ideal case is you have both the historical database numbers and the business KPIs. This way you can correlate the business peak w/ system usage.

If you don’t have the historical AWR data to see the effects of these business peaks, then the historical business KPI is fine. The key thing here is you want to see the growth rate and where the peaks happen.

Below is an example of business KPI of a worldwide money transfer company. Their ORMB infrastructure peaks during Mother’s Day and Christmas when customers need to send money to their loved ones. But in general year by year from 2016 to 2019 (not shown below) the growth rate is at steady < 3% per year.

I got this data from one of their functional teams because I had to forecast for the upcoming Christmas peak (SLA requirement for the hardware upgrade) which is still within the range from last year.

Chart

Description automatically generated with medium confidence

1. **More data = more accurate sizing and allows demand forecasting**

The more days you have on your sizing data the merrier. In the world of forecasting, you can only project out to the future with the amount of data points that you have. In other words, you can’t forecast for 2 years if you don’t have at least 2 years worth of historical data.

Let’s say my data points are normalized to days. If I only have 6months worth of history, I can only confidently project to the next 6 months and even if the trend seems like a straight line I would use the most conservative forecast quantile (outermost 99%) if I had to project out farther to the future until the cross point of the capacity line.

Here's an example of CPU forecast (<https://github.com/karlarao/forecast_examples/tree/master/monte_carlo>)

Diagram

Description automatically generated with medium confidence

ASM storage forecast

Chart, line chart

Description automatically generated