Paper 1105-2017

Implementing Capacity Management Policies on a SAS® LASR Platform: Can you afford not to?

Paul Johnson, Sopra Steria

ABSTRACT

Capacity management is concerned with managing, controlling and optimizing the hardware resources on a technology platform. Its primary goal is to ensure that IT resources are right-sized to meet current and future business requirements cost effectively. In other words, keeping those hardware vendors at bay! SAS® LASR servers with its dependence on In-memory resources necessitate a revisit to the traditional IT server capacity management practices.

A major UK based financial services institution operates a multi-tenanted Enterprise SAS platform. The 'tenants' share platform resources and as such, require quotas enforced with system limits and costs for their resource utilization, aligned to the business outcomes and agreed SLA's.

This paper discusses the Implementation of System, Operational and Development polices applicable in a multi-tenanted SAS platform, in order to optimise their investment in the SAS®LASR platform and be in control as to when capacity uplifts are required.

CASE STUDY

Most RDBMS have mature systems management processes complete with DBAs that in conjunction with platform administrators enforce system and capacity management policies as standard. SAS Platforms on the other hand, are not usually so favoured by the traditional IT administrators and as such usually have a rather less formal approach to governance and control. SAS when deployed as a business tool typically lies in the hands of end-users who are notoriously relaxed about housekeeping and adherence to capacity management principles, until performance is impacted or the platform resources become totally depleted. *The key to capacity management stems from being good at implementing effective proactive processes and are not therefore being solely reliant, on processes that are reactive*.

With this in mind, let us consider how capacity management in an enterprise wide shared multi-tenanted IT-supported environment is implemented on a SAS® LASR platform comprising:

- SAS 9.4 Enterprise scale multi-server grid cluster
- 10,000+ Users
- 1,000 Analysts
- Enterprise scale multi-server LASR cluster with approx. 30 Tb (In-Memory)
- Co-located SAS implementation of HDFS with approx. 40 Tb (serving as LASR cache)
- Connectivity to the Enterprise Data Hub comprising: Hadoop, Teradata and other RDBMS
- A fully loaded SAS analytics software stack spearheaded by SAS® Visual Analytics

As you can see this is quite a sizeable and complex environment, serving a large number of stakeholders. The cost to commission and maintain such a platform is quite significant both in terms of hardware and software, not to mention manpower. So how can you manage such an investment and enable the users to make the optimal use of the above technology components?

Now given that most of us can quite easily consume gigabytes of data just on our mobiles alone, just consider how long it will take an army of analysts equipped with SAS® Visual Analytics and connected to the Enterprise Data Hub to consume the LASR memory and associated processing resources? Perhaps a question best not posed at your platform vendors even though most do in fact provide a capacity planning or sizing service for these situations. The SAS® Enterprise Excellence Centre (EEC) services were actually used to size and validate the platform listed above.

CAPACITY PLANNING

Capacity planning is an essential discipline in determining the IT infrastructure required to meet future workload demands. It is often viewed as a delicate and continuous balancing act between productivity and cost. The key to effective capacity planning is about being proactive rather than reactive. Most of us are familiar with the expression “prevention is better than cure”, as far as capacity planners are concerned never has a truer word been spoken.

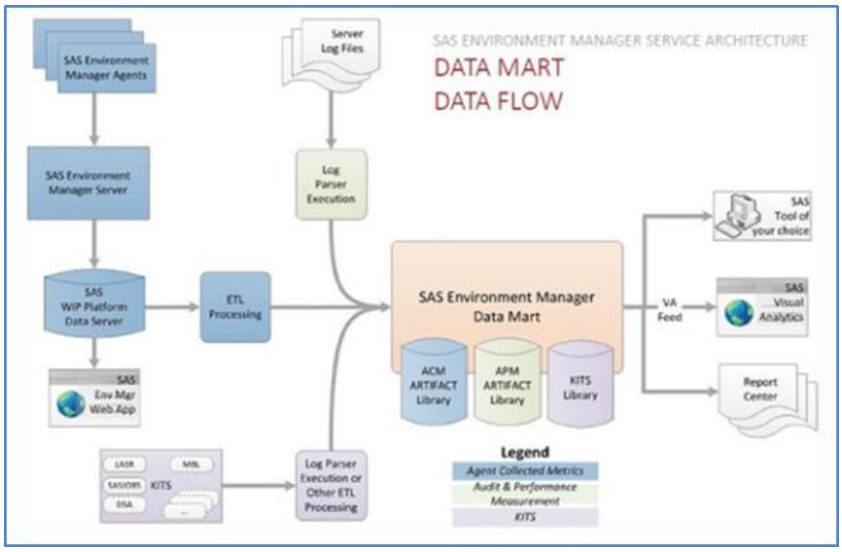
Some users though, may point out that storage and memory media is consistently getting cheaper year-on-year so why be concerned about capacity planning, especially for storage media? Well, if we take memory for example particularly in an enterprise wide, shared multi-tenanted clustered environment, consideration is required as to how far you can scale up vertically. There is also a strong correlation between CPU and memory, when vertical scaling options have been exhausted horizontal scaling then comes into play. Now even if the additional hardware were acquired, infrastructure upgrades have to be planned through governance and control involving design, engineering, architecture and support resources, all of which takes time not to mention money.

The benefits of capacity planning can be summarized as:

- Avoid waste from over-provisioning
- Save time
- Minimize costs
- Avoid risk to critical services
- Reduce capacity and performance bottlenecks
- Increased Productivity
- Avoidance of unnecessary hardware upgrades plus corresponding software license increases

USAGE INFORMATION

Information pertaining to current and historical platform resource utilization can be obtained from the SAS Environment Manager Data Mart (EMDM) as illustrated below. This collects information from the SAS logs to generate Audit Performance Measurement (APM) tables in addition to Agent-Collected Metrics (ACM) from the Servers and disks.



The usage information collected can be viewed via the sample SAS stored process reports provided or auto-loaded into SAS® Visual Analytics reports and monitored periodically.

All the standard measures such as: CPU usage, response time, disk allocation and memory utilization etc. are captured and reported on.

Additional monitoring agents such as Tivoli can also be used to collect capacity management information.

Figure 1. Environment Manager Data Mart Flow

DEMAND INFORMATION

Demand data should wherever possible be collected from the platform tenants although forecasting or modeling techniques can also be used, when accurate demand data is not readily available from the users. A capacity planning form or process should be used to regularly collate key demand measures such as those highlighted below across periods that span into the foreseeable future.

Viewers (consumers)			Developers (consumers)			Batch	
Number of Viewers	Number Viewer Concurrency	Avg. Daily Processing Time (hh:mm)	Number of Developers	Max Developer Concurrency	Avg. Daily Processing Time (hh:mm)	Number of Jobs	Avg. Daily Processing Time (hh:mm)

Display 1. Capacity Planning Form – Part I

The number of consumers, developers and batch job activity is captured (*see display above*) together with volume and capacity projections for both HDFS and In-memory LASR tables (*shown below*). The distribution of the LASR table volumes is also captured across varying size bands. An analysis of the values captured in the featured environment revealed that 95% of the number of tables planned to be built will exist within the first three size bands (from small to large) occupying 50% of the total installed capacity. Therefore only a small minority of tables, just 5% (volume) distributed across the two highest bands will require 50% of the remaining installed capacity. These volume-to-capacity ratios were taken into consideration in the capacity management policies adopted at this site, which was designed with fair sharing of system resources in mind.

Resource Summary (Total)				Resource Distribution - Tables across LASR Memory				
Total Number Tables	Disk HDFS (Gb)	Memory LASR (Gb)	Annual Growth %	Small < 2Gb	Medium 2 - 5 Gb	Large 5-15 Gb	Extra Large 15-50 Gb	Massive 50-286 Gb

Display 2. Capacity Planning Form – Part II

CAPACITY ON DEMAND

So having collected and analyzed your demand forecast information you discover that once the demand for in-memory capacity is realized, you will have reached full capacity and an uplift is therefore required. But before we do so, perhaps now would be a good time to introduce the topic of capacity on demand.

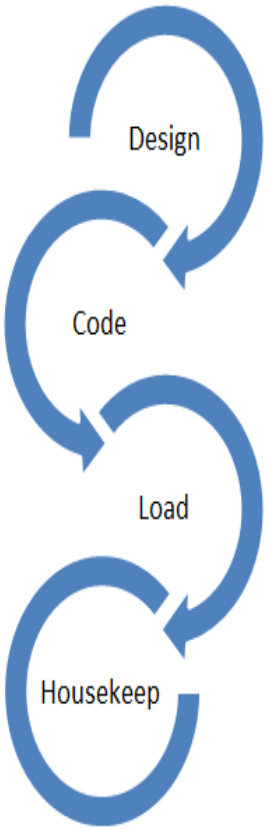
We are all familiar with the controls on a modern central heating system, these are used to ensure heating is scheduled to be on when you are home during the winter months. The hot water on a typical combi boiler comes on after a few seconds of running the tap. This system clearly works well, providing both heat and hot water whilst conserving energy and minimising costs. A similar approach can be taken in a LASR environment with co-located HDFS where loading also typically takes seconds! So for ad-hoc, model training or analytical discovery purposes should you not conserve LASR memory too and where appropriate incorporate a similar on-demand approach to LASR loading?

Figure 2. Central heating controls

The *SAS*® LASR Analytic Server is a persistent, in-memory analytical engine, designed to process large volumes of in-memory tables. Whilst LASR is a persistent service, it is not and should not necessarily be considered permanent particularly with co-located HDFS storage which is purely there to expedite LASR loading. We shall delve further into the capacity-on-demand concept in the LASR loading policies section.

LASR CAPACITY MANAGEMENT POLICY GUIDELINES

The implementation of LASR policies will contribute towards the optimal use of LASR memory resources.



LASR Design Policies – Design policies should be in place to encourage standardization and adherence to the emerging LASR best practices and to ensure general platform and data management policies are not inadvertently being breached.

LASR Coding Policies – Most platform support teams and development forums publish usage guides and best practice papers to the internal SAS user community forums. A number of LASR coding policies are detailed overleaf which are geared towards the goal of achieving the optimum use of the LASR platform and in particular making best use of the available LASR storage and processing resources available.

LASR Loading Policies – In order to best accommodate the inevitable high demand for LASR resources a number of loading policies may be implemented to help deliver a guaranteed level of service to the majority use cases and maintain a fair share of the available resources to all. The loading policies focus on, but are not necessarily limited to, tables classified as large or above (e.g. >= 5 GB) and the conditions upon which they can be loaded and more importantly persisted in LASR memory. The capacity-on-demand concept has been incorporated into the LASR loading policies implemented in the featured environment.

LASR Housekeeping Policies – All stakeholders have the responsibility for performing regular housekeeping on the LASR tier in order to preserve the capacity resources available. Basic tasks such as regularly deleting obsolete tables in both HDFS and LASR areas will potentially make the largest contribution to the platform longevity. The support teams are thus reliant on the LASR data owners and data administrators alike to implement effective local housekeeping processes.

Figure 3. LASR - Capacity Management Policies

LASR DESIGN - POLICY GUIDELINES

The top question on a LASR platform's FAQ section would be "*What is the maximum table size that I can load in LASR memory*?" competing closely with "*How much data can I store in LASR?*" Don't be surprised if as soon as you respond to these queries, user designs seemingly built around these responses surface.

On a more serious note the general approach taken to create a semantic layer for your existing BI tools should also apply when designs are being drafted for your in-memory LASR tier. Dimensional models including star schemas (*ref. IMSTAT*) are supported in LASR memory and are an effective way to avoid duplication in the LASR tier. Avoid generating reporting marts comprising very granular detail data that spans an extended historical period. Historical information serves well to analyze trends as opposed to detailed granular information generally used to facilitate drill down activity which is usually only required for more recent periods. Should both extended history and detail be required in a single reporting summary table, then consider selectively loading the subset of data required at run-time which satisfies the criteria required for your ad-hoc analysis. Maintaining a data dictionary of your HDFS and LASR tables with a good standard naming convention is highly recommended. Finally the need to secure your in-memory LASR tables using Row Level Security should also feature in your designs (Johnson, 2016).

LASR CODING – POLICY GUIDELINES

The table below provides guidance on the coding practices that can be adopted by developers and administrators on a SAS® LASR platform. These should supplement any SAS coding standards that may already be in place. In general, the coding practices listed below apply to all SAS tables or datasets irrespective of size or intended purpose.

Coding Practice	Description
General SAS Coding	• Use a where clause to filter any data not required for LASR processing. • Use a keep statement (*where applicable or SQL equivalent)* to only include the columns required for LASR processing. • Be explicit for the sake of both clarity and efficiency (e.g. don't use Select *). • Specify accurate column lengths from the data and don't rely on the RDBMS defaults. • Use appropriate dataset and column labels. • Assign meaningful table and column names.
Block Size	The block size you use when creating a HDFS table is absolutely critical in conserving both HDFS and LASR storage resources. The Block Size is basically calculated using the number of LASR nodes, the record length and number of rows. Do not opt for a 'one size fits all approach' particularly when you don't have all the above measures at hand. The block size calculation can be dynamically generated at load time from the metadata available via the SAS dictionary views or the equivalent DBA views if your source is an external RDBMS. The SAS supplied %GetHDFSOptimalBlockSizeOptions macro is a good source of reference for HDFS block size calculations. The number of copies you hold in HDFS is also an important capacity factor as is the INNAMEONLY single node limit.
Dataset Attributes	Datasets should of course have meaningful names complete with a descriptive label that ideally contains the user-id and creation date stamp. It is very important that column lengths are sized in accordance with the underlying values and not just defaulted from the source particularly when it resides outside SAS. Data can be loaded from a variety of RDBMS sources which do NOT always automatically translate into optimal column lengths in HDFS and in-memory LASR tables. Case in point Cloudera **Impala** character strings can default to a **32K** SAS column.
Formats	Space can sometimes be conserved by using SAS formats to assign labels to codes in the Visual Analytics interface. This may offer significant savings to the larger tables and should also be considered where codes and formats or lookup tables are already being used within the existing transformation processing before loading to LASR via HDFS.

LASR Compute Columns	Calculated columns and attributes may be set on the LASR table rather than replicated in each associated LASR report thereby offering the same benefits of the equivalent functionality enjoyed in SAS OLAP cubes or information maps.
	
	`PROC IMSTAT;`
	`  table LASRLIB.source_table;`
	`  compute z_computed "z_computed=x+y";`
	`RUN;`
	`QUIT;`

LASR PROC's	**PROC LASR** is generally used for loading and unloading of tables to/from LASR memory. **PROC IMSTAT** manages in-memory tables and SAS LASR analytic server instances. **PROC HPDS2** is highly performant analytical procedure; distributes parallel processing. **PROC METALIB** registers: SAS, RDBMS, HDFS and LASR tables in SAS metadata.

Table 1. LASR coding guidelines

Let us just recap on the standard user roles engaged in a typical visual analytics process to perform data exploration, analytical model building and report creation processes. The first two processes do not generally require LASR data persistence as they naturally involve some manual intervention, whereas reports are typically automated and so generally require persistent LASR data.

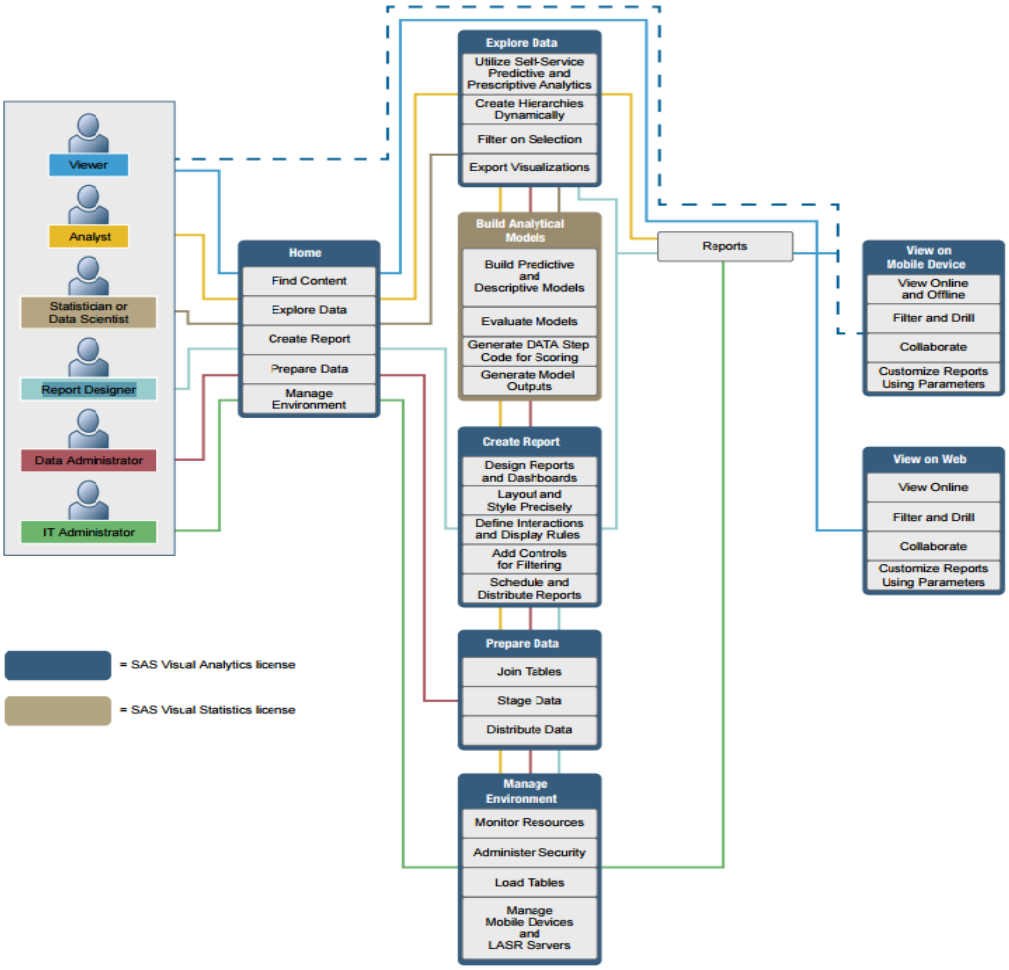


Figure 4. Visual Analytics - User Roles

The Data Explorer is intended for ad-hoc data discovery, model training and investigative insight purposes which often require large in-memory tables. These tables normally serve for a short period and as such should be purged from in-memory after use.

Figure 5. Data Explorer

In-memory tables surfaced via explorations should where possible be loaded in LASR on-demand and purged thereafter.

The Report Designer is intended for: KPI, and dashboard reporting. It is also used for: analytical reporting as well as general BI and MI reporting. Reports are thus geared more towards information consumers as opposed to analysts.

Figure 6. Report Designer

These reports generally require the underlying tables to be persistent in LASR memory.

HDFS and LASR Quotas

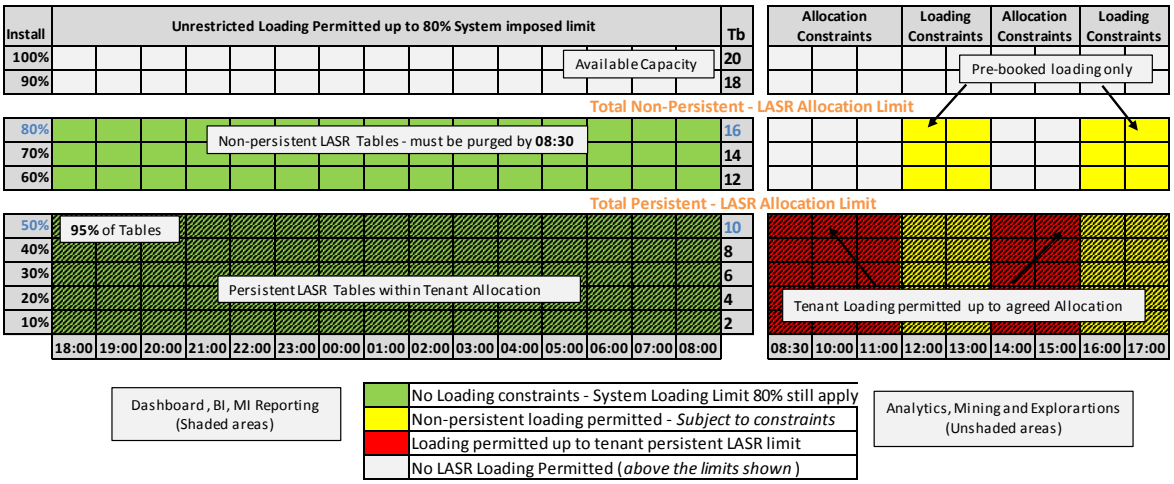
As you assign the appropriate visual analytics components and corresponding capabilities to the army of analysts, you will want to discuss and agree storage quotas for both LASR memory and co-located HDFS. LASR quotas particularly in a shared multi-tenanted environment, may need to be categorized into persistent and absolute (*includes non-persistent*) with quota limits set accordingly. As a starting point, data loaded for exploratory purposes should be regarded as non-persistent, whereas the majority of 'live' reports will be created from persisted LASR data. Now given that a typical co-located HDFS load into LASR will usually only take seconds this should sound quite reasonable. However this does require the cooperation of SAS Analytical users who remember may be quite accustomed to little or no governance in how they use their SAS tools. As such aim to agree and circulate the capacity management policies as early in the tenant on-boarding process as possible.

In an ideal world, this strategy would be easier to implement if the Visual Analytics tools automatically retrieved non-persistent data whenever a report or exploration is accessed. In fact auto-retrieval of archived or off-line data has been around for decades. *Now rest assured that this capability has officially been raised as a feature request with SAS*. So let us now delve further into how quota limits and associated policies are implemented in the featured environment.

Persistent LASR Allocation Limit – Is the maximum capacity a given tenant can load and persist tables in LASR memory. It is recommended that tenants prioritize tables used for: KPI, Dashboard, BI and MI reporting purposes within their quota, ahead of those used for discovery and Analytical Modelling. Whilst tenants are of course free to use their quotas as they wish, the key objective is to persuade them to be early participants in good capacity management processes, in order for all to reap the benefits later on.

Absolute LASR Allocation Limit – This represents the total limit up to which tenants can load both persistent and non-persistent tables in LASR memory. Tables not included with the persistent allocation area are considered non-persistent and as such may be subject to loading policies or constraints.

The LASR loading policy implemented at the featured environment is illustrated in the diagram below which shows that 95% majority of all the expected quantity of LASR tables *(as denoted by the cross shaded areas in the illustration below)* can be accommodated from around half the available LASR capacity. The remaining minority 5% non-persisted tables would therefore be allocated across the remaining half capacity, but in a controlled manner, i.e. during early or late afternoon (*yellow non-cross shaded*) or evening and night periods (*green non-cross shaded*).



Display 3. LASR – Loading Policy at the featured environment

The LASR loading during the restricted daytime periods is done via a capacity reservation process managed by the platform support team, which is discussed further in the sections to follow.

LASR LOADING – POLICY GUIDELINES

The table below lists some loading guidelines that can help manage the capacity on a LASR platform.

Policy	Description
Loading Window Restrictions	**Persistent Quota – LASR Loading** (*No restrictions apply up to the agreed tenant Quota*) **Absolute Quota – LASR Loading:** • Tables above the persistent quota limit are restricted to temporary loading via the capacity reservation system during 12:00 – 14:00 ,16:00 – 18:00 (Mon – Fri) • No loading restrictions apply otherwise up to the absolute tenant quota limit.

Load Time-Limit	Tables loaded during restricted periods and above the tenant persistent limit, should have a self-imposed limit equal to the agreed reservation duration period. Load limits (*specified in secs.*) are achieved as follows:
	`PROC IMSTAT data=LASRLIB.source_table;`
	`lifetime 3600 mode=ABSOLUTE ;`
	`run;`
	`QUIT;`
	*Note tables held within the agreed Tenant persistent quota are **exempted**.*

Purge after Use	Non production tables particularly those classified as: Extra Large or Huge should ideally be purged from memory immediately after use via any of the following: • The IMSTAT procedure with the LIFETIME statement (mode=lastuse) • The PROC LASR procedure with the remove statement • Interactively using the unload option within the VA administration interface
Inactive Purge	All LASR tables persisted or otherwise, held outside the production LASR area and not used within a preset inactive period (*to be agreed*) should be subject to purging from LASR memory at the discretion of the support team. Bear in mind that a co-located HDFS copy of the LASR table exists to accelerate LASR loading which should be a matter of seconds. So once the inactive period has been agreed then this policy will result in better utilization, efficiency and potential cost savings.

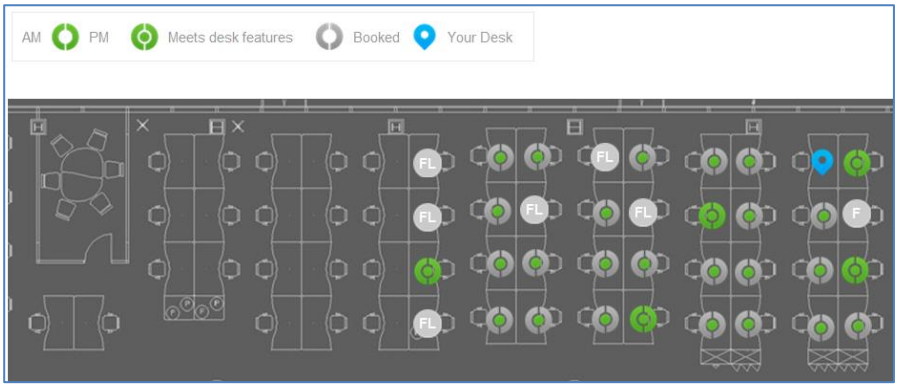
Selective Loads (subset)	Selective loading makes use of a where clause (*as specified below*) to load the required subset of data into LASR as needed for immediate querying via the report or explorer interface. This improves both query performance in addition to conserving valuable LASR resources.
	`PROC LASR port=<port number> add noclass`
	`data=HDFSLIB.source_table (WHERE=(isoname='United Kingdom'))`
	`SIGNER = "https://www.LASR.web.host.net:443/SASLASRAuthorization" ;`
	`performance host="<hostname>";`
	`RUN;`
	This approach can be incorporated into a SAS Stored Process invoked from within the report interface to provide a near seamless user experience. A row level security (RLS) approach whereby rows specific to a given user on an RLS enabled table is loaded at run-time and purged thereafter may also be implemented.

Compression	LASR supports compression which may reduce the memory footprint albeit generally at the expense of increased CPU time. Tables can however be uncompressed prior to querying which will address the performance overhead. Compression needs to be assessed on a case by case basis as not all tables will benefit. Compression can be achieved by using the **squeeze** HDFS dataset parameter or via the PROC IMSTAT **compress** statement.

Table 2. LASR coding guidelines

Capacity Reservation System

Implementing a capacity reservation system is an effective method of adopting the capacity-on-demand philosophy particularly when you are challenged by a capacity full scenario. A real life example of an increasingly familiar capacity reservation system is illustrated below.



Most large companies based in major UK cities are challenged to provide enough desks for all their employees.

Desk reservation systems are typically used to maximize the desk space usage based on a hot-desking policy which yields significant cost savings across the organization.

Display 4. Desk Reservation System

The LASR support team may well see the benefit of implementing a capacity reservation system or process to manage on-demand LASR usage within the load reservation periods, which for the featured platform is during lower utilization periods of: lunch time and late afternoon hours.

The purpose of the reservation limits is to protect the LASR servers from being saturated, although system loading limits are of course set to protect the LASR servers from potential over-loading. The diagram below illustrates a simple capacity reservation system, which can be used to manage the loading of non-persistent LASR tables. The system illustrated permits a maximum of:

- Two tenant departments that can load above their persistent limit in any single restricted hour
- Two hourly reservation slots per day, these can be consecutive (*not shown*)
- Four reservation slots permitted per week

Example Loading Policy Restriction: Max 2 Tenants per hour, each tenant can reserve upto 4 hourly slots per week							
Period	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**% Usage**	
17:00 - 18:00	Tenant 3 Tenant 4	Tenant 7 Tenant 8	Tenant 3 Tenant 4	Tenant 7 Tenant 8	Tenant 9 Tenant 10	80% 50%	Non-persistent Loading permitted to System Limit
16:00 - 17:00	Tenant 1 Tenant 2	Tenant 5 Tenant 6	Tenant 9 Tenant 10	Tenant 5 Tenant 6	Tenant 3 Tenant 4	80% 50%	
13:00 - 14:00	Tenant 3 Tenant 4	Tenant 7 Tenant 8	Tenant 1 Tenant 2	Tenant 7 Tenant 8	Tenant 9 Tenant 10	80% 50%	
12:00 - 13:00	Tenant 1 Tenant 2	Tenant 5 Tenant 6	Tenant 9 Tenant 10	Tenant 5 Tenant 6	Tenant 1 Tenant 2	80% 50%	

Display 5. LASR Capacity Reservation System

Ideally the algorithm should be dynamic and take the actual LASR persistence levels on the platform into consideration, as well as the additional capacity required to be loaded by each tenant, for the respective period. The system may also be used to permit non-persistent loading during low utilization periods that extend beyond the hour or two described above. Alternatively, if your site employs a capacity management or simulation tool then that can also serve as a feed for, or deployed as a capacity reservation system.

SYSTEM & SUPPORT CONTROLS

The LASR environment will usually span two storage media: HDFS (co-located on the LASR servers) and LASR which as we know resides in memory. In a shared multi-tenanted environment the main task will involve setting and enforcing controls to ensure that a stable and performant environment is maintained. Quotas for both HDFS and LASR will undoubtedly be a key point for debate. However, if the capacity management process is planned properly to include where necessary an agreed cost or chargeback model then all the values needed to apportion quotas will of course already be determined.

The key potential controls available to enforce quota limits are as follows:

- Soft controls using the capacity and system monitoring sources to detect when quota limits have been breached resulting in a penalty or chargeback cost to frequent offenders. A simple name and shame approach will however usually suffice as a deterrent.
- Linux control groups (cgroups) are a popular method of enforcing quota limits on host servers. Control groups can be implemented for CPU, Disk, I/O etc. but not supported by LASR memory.
- User Resource limits (ulimit) are used to regulate: the max no. or size of files, CPU and memory.
- TKGrid (resource.settings) script is typically used to enforce in-memory LASR quota limits at the user group level. Once customized the script must be replicated across the LASR cluster servers.
- YARN is the Hadoop resource mgt. tool that can be used to enforce HDFS and LASR quotas.

CAPACITY MANAGEMENT FINDINGS AND RECOMMENDATIONS

As most companies capacity management processes are well established they have intentionally not been covered in their entirety within the scope of this paper. Topics such as Service Level Management and Chargeback would typically feature highly in most capacity management discussions. The key objective of this paper has primarily been to identify processes and policies that may need to be revised from existing in-house processes to accommodate a SAS® LASR platform. The primary SAS® LASR capacity management recommendations highlighted in this paper may be summarized as follows:

- Assign Visual Analytics components and capabilities very carefully especially the load capability.
- Load LASR tables via co-located HDFS, whereby HDFS is used as hot-cache and can therefore be used to facilitate on-demand loading.
- Give careful consideration to the maximum in-memory table size and tenant quota limits you set.
- Review your design, code, load and housekeeping policies and do not be afraid to name and shame policy offenders, or introduce chargeback penalties if all else fails.
- Discuss and agree the new LASR capacity management policies with the users at the outset.
- Consider the impact that the Big Data Platform will have on future LASR capacity demand.
- Incorporate the SAS EEC study service into your capacity management process, once capacity thresholds are genuinely being threatened.
- Implement an effective auto-loading policy, incorporating on-demand loading where feasible at least until an on-demand capability is incorporated into the SAS® Visual Analytics product, which I understand is forthcoming.

CONCLUSION

With the inclusion of Big Data to the Enterprise Data Hub in conjunction with the popularity of in-memory solutions facilitating analytics; now is a good time for you to give your capacity management processes an overhaul. With six-digit upgrade costs quite commonplace, you can save a tidy sum just by increasing the efficiency of your capacity management processes, perhaps even adopting some of the policies discussed in this paper. Whilst we all like the sight of shiny new tin appearing in our data centres, courtesy of our obliging vendors, we do need to pay careful attention to our capacity management processes. Lastly, capacity and performance go hand in hand, if you take care of capacity better performance usually follows and that will always be a much more welcome sight, on any IT platform.

ACKNOWLEDGMENTS

The author would like to call out his manager at the UK financial institution for his sincere support and encouragement with the publication of this paper. Further acknowledgement and thanks is extended to the Data and Analytics Support team for their cooperation in the compilation of this paper.

REFERENCES

Johnson, Paul. 2016. "*An End-to-End, Automation Framework for Implementing Identity-Driven Row-Level Security Using SAS® Visual Analytics*" *Proceedings of the SAS Global Forum 2016 Conference*, *Las Vegas, NV.* SAS Institute Inc.

Available at: http://support.sas.com/resources/papers/proceedings16/11701-2016.pdf.

RECOMMENDED READING

- *SAS® Visual Analytics 7.3: Administration Guide*
- *SAS® Visual Analytics 7.3: User's Guide*
- *SAS® Environment Manager 2.4 User's Guide*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Paul Johnson

sopra steria
CONSULTING

Sopra Steria
4th Floor
30 Old Broad Street
London, EC2N 1HT
United Kingdom
paul.johnson@soprasteria.com