Preserving Large Quantities of Data and Maintaining Digital Sovereignty

Alberto Pace
CERN, IT department
Geneva, Switzerland
alberto.pace@cern.ch

Abstract— *Corporate data are often the most valuable assets that companies need to access, analyze and therefore preserve to ensure business continuity. As technologies are evolving rapidly and the volume of data is increasing exponentially, the straightforward response to this challenge is to outsource the problem to external IT companies that can provide attractive costs and effective solutions. However, this does not come without the risk of creating uncontrolled external dependencies and vendor lock-in that turn to be irreversible and can endanger the core business itself and even threaten its survival.*
This paper presents some issues and mitigation strategies that could be adopted in designing proper solutions.

Keywords—RDM Strategy, FAIR, data repository, workflow.

I. INTRODUCTION

In recent years, the evolution of storage technology has improved beyond the best expectations. In the early '90s corporate servers had capacities up to 1 GB. Nowadays, corporate servers can exceed the one PB capacity, i.e. one million times more, representing an increase of six orders of magnitude.

All the other components of computing witnessed similar evolutions. CPU performance has increased by three orders of magnitude while the multicore, distributed computing approach has increased the number of CPU cores available to a data processing application by 2 or 3 additional orders of magnitude.

Networking has also experienced similar improvements. Again, in the early nineties, computers could be connected with networks that were reaching few Mbit/s of data transfer rates. Today, it is possible to have connections that exceed 100 GB/s, which represents a five orders of magnitude improvement.

This trend is not finished yet. Actually, we are just at the beginning and there are clear hints that this trends will continue for several years. For instance, today you can purchase a micro-SD card of 1TB capacity. If you scale the volume of a micro-SD to the volume of a 3.5" hard disk, you could expect more than a 2.8 PB capacity in the volume of a single hard drive.

II. PERFORMANCE AND RELIABILITY OF HIGH DENSITY STORAGE

Another surprising evolution on large flash capacity is that we can expect both performance and reliability to increase. This reason is that the usual techniques used in large servers to increase both performance and reliability can be embedded into a single storage device.

The most known technique to increase performance is striping , where the data are split and then read and/or written in parallel to multiple and independent storage locations. This offers the possibility of an arbitrary performance increase that can be obtained by demultiplexing a data stream to an arbitrary number of parallel streams that access different storage locations. Clearly, this technology can be embedded into the single storage media.

Similarly, there is a wide set of error correction techniques that can be used to increase reliability. This is generally obtained by dedicating a small part of the storage capacity in the media to store error correction information that can be used in case of a media read and/or write error. In this scenario, traditional errors that would in the past lead to data loss, would be automatically corrected by the redundant information and the software embedded in the media, thus significantly increasing the media reliability. In addition, these techniques allow predicting data loss with high accuracy, because the amount of redundancy left available can be used to precisely estimate the probability of data loss.

Therefore, we can expect in the coming years, the appearance of storage devices with very high capacity, very high performance in terms of both latency and throughput as well as an outstanding reliability.

III. THE CHALLENGE OF DATA PRESERVATION

The cost reduction that we have observed in storage media should ease the problem of long term data preservation. When the cost drops, it becomes more affordable to buy additional media to store additional copies of the data that needs to be preserved. Therefore, the naïve conclusion is that this evolution significantly facilitates the data preservation.

Unfortunately, this statement is only partially true, as there is another aspect to take into consideration: If storage is cheap, the amount of data that is produced increases. In this case, the challenge of data preservation becomes more difficult because the amount of data to preserve increases exponentially over time and because data need to be moved from older media to newer ones, the constraints on the surrounding architecture is constantly increasing.

Let's take the example of the networking requirements for moving hundreds of petabytes compared to what was required, few years ago, to move hundreds of terabytes. One could take the simplistic approach to address this challenge by deploying a network that is 3 orders of magnitude faster. Could this approach be effective ? Probably not: ten or twenty years ago, moving hundreds of terabytes would take a couple of months with, on average, one error generated for every terabyte copied. This was producing approximately a hundred of errors to be manually resolved in a couple of months. This was a realistic process to implement at that time.

If you would use the same architecture today, with a network 3 orders of magnitude faster to transfer 3 orders of magnitude more data, you would certainly fail to achieve it in a couple of months. Why? Because as you have the same architecture you can expect the same error rate, and therefore instead of some hundreds of errors, you will have more likely some hundred thousand errors (3 orders of magnitude more) to be manually addressed before you succeed. This highlights why the architecture must be reviewed and, in the particular example, it becomes evident that a specific development is essential to process known errors in a complete automated way, as the manual approach does not scale with the growth.

IV. PRESERVING DIGITAL SOVEREIGNTY

Digital sovereignty is a fashionable word used today in describing computing architectures where external dependencies are minimal. This means that there is no external body that can either de jure or de facto exercise pressure or constraints on the ability to take the necessary decisions to improve an existing architecture.

A more practical viewpoint to define digital sovereignty is related to the identification of which (digital) activities can be outsourced while maintaining the authority to self-govern.

The comparison with outsourcing is important because this approach makes you lose digital sovereignty. On this point, the industry defines few clear criteria that must be addressed before outsourcing, such to be effective.

Namely:

a) The activity is not strategic nor it is core business

b) The activity has clear established standard interfaces or protocols that are used to define the outsourcing contract

c) There are multiple independent vendors implementing these standard interfaces.

If any of these three requirements is not satisfied, you are exposed to problems, in particular vendor lock-in, business or service failure, even blackmailing. The vendor lock-in has a critical impact when data are involved because the ultimate goal to preserve the access to your data is at stake, given that companies and contracts can fail, law changes and contracts can be subject to remote jurisdictions.

There are two levels of external dependency that can severely limit your independence from a particular vendor:

a) the fact that the vendor has your data

b) the fact that you use licensed software from your vendor and that the license can be revoked.

These two aspects will be discussed in the following paragraphs.

V. STORING YOUR DATA IN THE CLOUD

If you store your data in the infrastructure of a cloud provider, you rely on the vendor to implement all the processes necessary to ensure that the data are securely stored, preserved and not accessed by third parties. Of course, you can also store the data yourself on premises. However, when data are on premises, you will have to implement the same processes, but with the advantage that you will be able to audit your infrastructure to know where you are standing.

One question is: how can you verify that all processes you expect to be in place on the vendor's side to properly manage your data are really there? this verification is difficult. Often it relies on a blind trust in the vendor, or it is just "because it is written in the contract".

With data stored on the cloud, you have some indicators and statistics that can help you. For example, the fact that you pay a higher price for a service may give you the confidence that the service is better managed. Or the fact that a vendor claims to have a large number of satisfied customers may also reassure you, as a trouble shared is a trouble halved. Unfortunately, all these arguments are just marketing statements that are unrelated to real facts. Another qualitative indicator is the fact that the vendor claims several years of successful operation without major incidents or data loss. But also in this case, this is another good example where the disclaimer "Past performance is no guarantee of future results" fully applies.

This is where the standard outsourcing approach is important when moving data to the cloud and the cost analysis and the risk analysis are both essential.

First, the cost analysis should go well beyond the recurrent cost per byte stored. The network transfer cost must be carefully evaluated as vendors typical offer free data ingestion, but expensive data retrieval. In addition, the time required to execute a complete data retrieval is important, as you must define from the beginning a possible exit strategy to avoid lock-in. Finally, it is also important to ensure that costs are guaranteed over a certain number of years and that you know your future costs sufficiently early so that you have enough time to review and adapt your strategy.

Then comes the risk analysis, which is by far more delicate, because it requires a subjective judgment to measure the probability of a bad event and its impact. Having said this, here is a list of possible bad events that must not be underestimated:

a) the loss of access to the data due to a technical incident in the vendor's premises

b) the loss of access to the data due to a contractual disagreement with the vendor

c) the loss of access to the data due to a decision from the vendor's jurisdiction

d) the fact that you will need significant investment to either take your data out, or to change vendor (you are locked-in)

e) the sudden (or planned) increase of cost for the vendor service that you are paying for

As there are many things that can go wrong, the single cloud provider approach should only be considered as a short-term solution, where data is not strategic and long-term preservation is not required.

On the other hand, if long term solutions are needed, there are several approaches that can mitigate the risk, the most obvious is to have multiple cloud providers. With multiple cloud providers you are able to constantly verify the interoperability among the cloud providers which will ensure that you are not locked in. You can also store your data multiple times across providers so that any access loss to one copy with one vendor will not affect the other copy.

However, be aware that all these mitigation strategies are effectively transferring back to you the workload that you thought you had outsourced. With multiple vendors, you are now in charge of the data management, allocating and moving the data and archiving it to locations that are now considered ephemeral or unreliable. Therefore, you have insourced the process of data preservation that you initially wanted to outsource.

So far, we have discussed only the data preservation requirement. However, there are many other risks to take in account, when outsourcing cloud storage. A significant example is the risk of disclosure of confidential data due to a technical incident in the vendor's premises. Also here, there are mitigation strategies. In this particular example, the usage of client-side data encryption voids the risk, as the vendor has only access to blobs of encrypted data, and any disclosure of this data would have no negative impact. But … now you have to manage the encryption keys and to provide keys to decrypt the data to any relevant stakeholder who need to access the data. De facto, you have insourced all your security, despite the goal of outsourcing it.

Finally, another mitigation strategy is to keep an additional copy on premises, but again, this comes back to insource what you intended to outsource.

VI. SOFTWARE LICENSES AND DIGITAL SOVEREIGNTY

Similar to the usage of cloud providers, another critical component to preserve digital sovereignty is the software and the licenses that are needed to use it.

The available options go well beyond the simple decision between open source vs proprietary software, and the best approach is, again, to apply the outsourcing strategy and identify costs and risks involved, knowing that the licensing horizon offers countless possibilities.

Whenever:

a) the data being processed does not require high confidentiality,

b) the processing required is based on standard functionalities,

c) there are (tested) interoperable vendors implementing the desired algorithms,

then the need to have source code access is reduced, and a proprietary license may be the most appropriate option. In this scenario, especially when multiple vendors are available, you can expect to be in a strong position to be able to negotiate a cheap license in a sustainable partnership with the vendor that will last several years.

As for the use of storage cloud providers, whenever any of the previous conditions is missing, then a risk analysis becomes necessary. In this case, among all possible risks to be evaluated, one can mention:

a) the risk when using closed source solutions that the software leaks information to the vendor or contains unwanted hidden features that can compromise your security (for example: backdoors or maintenance interfaces)

b) that some or all functionalities of the software can be suddenly remotely disabled by the vendor

c) that license renewal conditions can be unilaterally imposed on you at the end of the current licensing period

Points a) and b) can be mitigated by giving preference to the open source approach that allows the software running in your premises to be scrutinized. On point c), it is important to have licensing conditions that allow some reduced use of the software beyond licensing expiration in case of non-renewal to ensure to have the necessary time to migrate to alternative solutions. In the case of a computing infrastructure that hosts a large amount of data, migrating to alternative infrastructures can take years, and this time must be included in the risk analysis.

These points demonstrate that if the software plays a role in delivering your business, being able to control its strategy can really give you a significant competitive advantage. In this case, it is obvious that software provides a huge flexibility and that is where you invest. However, the more you invest, the more complex it becomes, and this is the area where you need top-level skills in your staff.

As mentioned, using proprietary software under license offers the advantage that costs are easy to estimate and customization can be purchased. In this scenario the competitive advantage remains effective while your license is kept cheap. It is equivalent to outsource an activity that is not important to your business, in order to reduce the cost.

On the other hand, when you move to open source software you may save in licensing cost but you need high skilled personnel that maintain the strategic software, and this can be expensive. However, with this approach, you gain in flexibility and you only pay the customization cost.

Finally, one last option which gives you total sovereignty but also the highest cost is the full stack development, done internally in closed source. This approach is often taken when the software is so strategic that the software becomes the business.

The process to define the software strategy is very similar to the one used to define the extent of cloud storage usage: it is entirely based on a cost and risk analysis approach.

One final important aspect to be aware of is that, when choosing the open source approach, the critical mass to build a successful infrastructure is beyond what a small institute or company can usually afford. In this case, the consortium approach is the best practice to collaborate on well-focused projects that guarantee maintaining ownership of the critical activities at a minimum cost. However, you will not have the exclusiveness competitive advantage of the developments produced by the consortium.

VII. CONCLUSION

If you have various vendor relationships, you should not be surprised that the more critical a component is to your business, the more marketing pressure you will receive to outsource it. The general recommendation is to outsource only standard activities that are well defined and interoperable. This means that you should insource what is specific to you, and your critical activities. Do not outsource your own business! The open source approach remains the best practice to insource your critical activities at a minimum cost, and when you cannot afford the cost, it is preferable to have a consortium approach rather than to accept a vendor lock-in. Both the consortium and the open source approaches can guarantee a fixed cost for software. When you have reached an architecture where software accounts only as a fixed cost, you have reached the perfect scale-out solution, where the marginal cost of your growth will be minimized as you pay only for the cost of the additional hardware and the additional energy consumption. This approach is particularly valid for storage. If you manage to have no variable cost for the software, your cost of adding additional storage will be minimal and you can expect huge savings compared to cloud storage. The only drawback is that the critical mass that you need is large, but this can be addressed with the consortium approach.