

Hybrid Cloud Bursting: 6 Implementation Fundamentals

White Paper, by Shiv Sikand, Executive Vice President, IC Manage

I. OVERVIEW: HYBRID CLOUD BURSTING & ON-DEMAND HPC

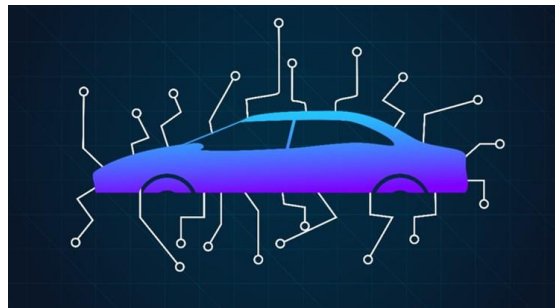
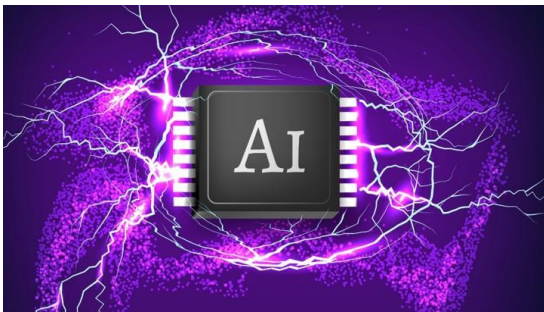
During this time of data explosion and high-performance computing (HPC) requirements, companies have turned to hybrid cloud bursting to deliver elastic compute power when demand exceeds available on-premise capacity.

Cloud bursting can enable development teams from companies of all sizes to instantly run compute jobs with 1,000s to 10,000s of cores and more.

Fundamentals to Assess when Determining Hybrid Cloud Bursting Approach

This white paper discusses key hurdles and implementation fundamentals to address when setting up cloud bursting from an existing on-premise workflow. The detailed content can serve as helpful metrics as your company formulates its hybrid cloud bursting implementation plan.

Although the paper uses semiconductor design as an example case for clarity, the same concepts apply to any number of hybrid cloud bursting applications.



Hurdles

Extremely large EDA data volumes and workflow complexity are causing serious implementation hurdles and challenges, especially when copying on-premise tools, data and workflows into a cloud environment is a pre-requisite to running jobs in the cloud.

The copying process is lengthy, plus synchronizing and supporting two separate flows is extremely expensive. It basically doubles the infrastructure requirements, including back up and disaster recovery, monitoring and job scheduling, along with managing different support teams that understand each environment. Added to this, cloud data storage costs scale up, becoming expensive over time.

An Ideal Hybrid Cloud Environment

Cloud bursting from an effective hybrid cloud environment addresses all the key implementation fundamentals to deliver immediate, transparent access to the cloud's virtually unlimited capacity, while fully leveraging existing, on-premise compute-farm workflows in the cloud.

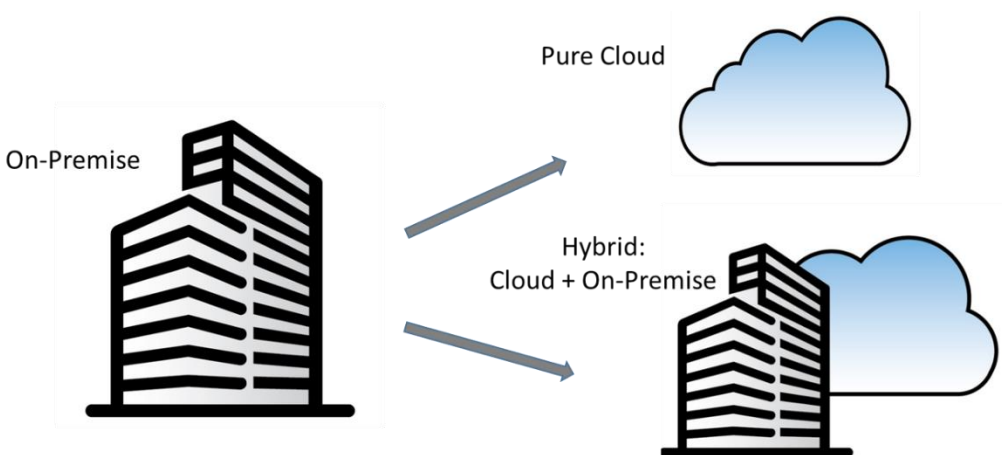
This paper's findings and recommendations are based on IC Manage's work with multiple companies to add to their existing on-premise EDA compute power by using hybrid cloud bursting for unlimited high-performance compute resources.

II. UNDERSTAND CLOUD WORKFLOW OPTIONS

Although it's a bit over-simplified, the primary cloud deployment workflow options are pure cloud and hybrid cloud/on-premise.

A Pure Cloud Workflow is custom built to run only in the cloud. It is architected to take advantage of the parallelism and storage facilities of the cloud. Pure cloud tool sets appeal to companies who want elastic compute and do not already have an entire chip design workflow environment. Additionally, some companies want to move away from managing their own data centers and instead outsource their entire compute infrastructure to pure cloud.

A Hybrid Cloud workflow extends the existing on-premise, compute-farm workflow, with an identical workflow in the cloud. The goal of the hybrid environment is to leverage the substantial investment in on-premise workflows, while enabling users to achieve elastic high-performance compute by scaling up to take advantage of the virtually unlimited compute power of the cloud. Engineers jobs can burst to the cloud during times of peak demand, providing capacity in a transparent fashion. There are varying implementation approaches for a hybrid cloud workflow option, which we will discuss in more detail.



III. THE 6 FUNDAMENTALS FOR IMPLEMENTING A HYBRID CLOUD

1. Copying On-Premise Flows to the Cloud is Unnecessary & Expensive

The way many companies have tried to achieve a hybrid workflow is to copy their on-premise workflows to the cloud. The process goes along the lines of the following:

1. Set up a cloud tool environment
2. Copy a proposed minimum set of on-premise data into the cloud
3. Configure and run in the cloud
4. Cycle back to Step #2 because of missing files that were needed due to hidden references in workflow scripts

However, if you maintain duplicate copies in both the on-premise and cloud environments, you also must constantly synchronize your on-premise dataset with each additional cloud projection. This synchronization can be error prone, similar to the sync problems that multi-site development teams experience; further, it is magnified due to the much greater compute capacity in the cloud.

As a result of the copying process, you also end up building two environments, doubling your infrastructure, including back up and disaster recovery, monitoring and job scheduling. Additionally, you will need two support teams to understand each environment.

Ideally, a true hybrid workflow will have only one workflow to support and maintain, while retaining the full ability for developers to run jobs transparently on-demand, either on-premise or in the cloud.

The data needed to run a job in the cloud is only a small portion of an on-premise flow



Unfortunately, determining the correct subset of on-premise data and copying it into the cloud is extremely hard due to the level of effort required to determine all the interdependencies between the data and the workflow scripts. Further, these interdependencies change over time, compounding the problem.

The workflow breaks down and jobs fail when the right on-premise data set was not copied into the cloud correctly. Incremental updates are painfully slow, and the process is highly manual, requiring a lot of trial and error. We've seen companies take 3 weeks to get just one block running basic functional verification.

To avoid the associated costs and project delay setbacks, the best practice is to avoid duplication whenever possible.

2. Long Data Setup & Upload Times Hurt Usability & Multicloud

Long Setup Time Limits Multicloud Environments

The longer it takes to set up and tear down a single cloud environment, the less likely you are to have the resources to create and support cloud environments from multiple vendors.

On top of this, a lengthy setup process means you are likely to leave your data stored in the cloud, rather than deleting it after running the jobs needed to accomplish an immediate task.

Data Upload Latency Can Erase Hybrid Cloud Bursting Advantages

As engineers work on their design, they typically make design changes, add improvements, get PDK updates, get tool/workflow updates, and do bug fixes on-premise. They then upload their new revisions by rsyncing all these design deltas into the cloud. These steps can take a few hours to a full day, plus lead to data inconsistency errors.

This time-consuming upload latency can fundamentally eliminate the fast compute advantages of cloud bursting.

3. Cloud Data Storage Adds Costs & Risks

Cloud Storage Costs

For cloud bursting to be financially sustainable for an organization, cost factors must be well understood and monitored.

Cloud storage pricing may not seem very high until you realize that a full environment can require hundreds of terabytes of storage, which can be expensive to keep online for months on end.

Deleting Cloud Data Following Cloud Bursting Reduces Risks

When you are done with cloud bursting, it is ideal to erase your data without any trace.

By doing so, you minimize your attack surface and your risk of having valuable IP stolen.

4. Cloud Download Cost can be Expensive

Uploading data to the cloud is free (other than latency 'costs'). However, getting data out of the cloud is very expensive. At a cost of approximately \$10/GB, it can add up quickly.

Thus, you will add a lot of challenges and cost if you must maintain coherency between on-premise and cloud environments. For example, you may want to download your logs and results files, but not your big SPICE transients.

Selective write backs address this factor. For example, a system that saves all writes as deltas and only sends back the data appended to a file, will minimize your cloud data download costs.

5. Cloud Application Performance is a Success Metric

The goal of hybrid cloud bursting is to achieve immediate access to additional resources for high performance computing during peak workloads.

And yet it is common for application performance to drop substantially in the cloud — for a 50% and lower speed than on-premise. This is particularly true for I/O intensive jobs. This minimizes the compute power advantage and raises your compute costs.

Understanding how your hybrid cloud bursting system will manage cloud compute workloads and handle I/O scale out is important. Thus, it is ideal to benchmark application performance also.

6. Cloud & On-Premise Infrastructures are Different

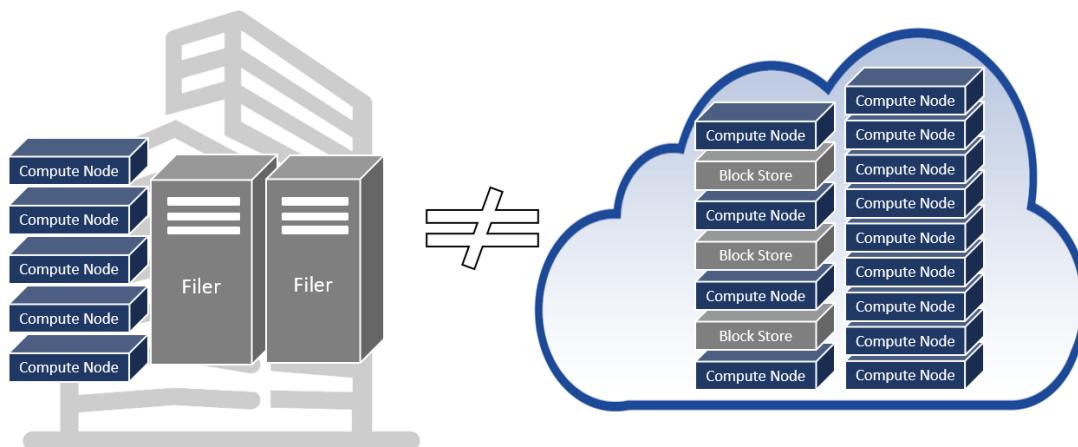
On-premise EDA tools and workflows are built on NFS-based shared storage, where large numbers of shared compute nodes and running jobs expect to see the same coherent data. In

contrast, the norm in the cloud is block storage, which is typically only accessed by one host at a time.

This block storage is very common for high-performance computing applications. But large parallel on-premise file systems, such as Glustre, GPFS, and Ceph, are optimized for multi-node concurrent file write access, not EDA workloads.

On-premise data coherency across many compute nodes is achieved via an on-premise NFS filer, but there is no high-performance equivalent function available as a native instance in the cloud.

A couple of major cloud vendors offer a shared storage solution; even so, while the storage can scale to many millions of nodes, the single thread performance is still much lower than that of an on-premise NetApp or Isilon, which reduces throughput in the cloud, increasing the compute and effective software license cost of each job transferred to the cloud.



Companies can consider multiple options to address this infrastructure disparity between the on-premise and cloud workflows.

- **Roll your own filer.** NFS filers and Linux-based NFS servers are “scale-up” solutions (meaning a bigger box), while cloud architectures are built for “scale-out” (meaning you add more boxes) The disparity between scale-out compute availability and scale-up storage leads to a severe mismatch between compute and file I/O performance, resulting in low job throughput. As a result, while building your own NFS in the cloud using standard Linux servers and cloud block stores may be suitable for mid- to low-end performance requirements, the setup and administration overhead can quickly spiral.
- **Use external filers.** An external filer is not only expensive, but it’s slow for cloud jobs because of additional latency to reach your storage devices.

- **Use a cloud vendor filesystem.** This can address the compatibility challenges; however, shared cloud storage is slower than on-premise storage. This cloud storage option is also more expensive.
- **Use a third-party filesystem.** HPC performance for a cloud-based filesystem requires data storage across many nodes; an “all flash cloud” is very expensive.
- **Use HPC cloud caching.** HPC performance by utilizing caching between multiple readers and writers. Uses small flash as cache, not storage.

IV. CONCLUSION

The use of hybrid cloud bursting is accelerating, as many development teams see leveraging the cloud as the most effective way to meet compute capacity during peak workloads.

As you investigate how to create the optimal hybrid cloud environment for your company, reviewing these hurdles and implementation fundamentals can help you better assess your priorities. Below is a brief summary matrix of the primary implementation options against the key metrics.

Hybrid Cloud Bursting Implementation Metrics

	1	2	3	4	5	6
	Data Duplication & Synchronization Eliminated	Cloud Setup Time (Impacts Multicloud)	Data Upload Time	Cloud Storage Costs	Cloud Data Download Costs	Cloud application performance (vs. on-prem)
HPC Cloud Caching	Yes	Minutes	Minutes	Low	Low	Faster
Roll Your Own Filer	No	Weeks	Days	High	High	Slower
Use External Filer	No	Weeks	Days	High	High	Slower
Cloud Vendor Filesystem	No	Weeks	Days	High	High	Slower
3rd party Cloud Filesystem	No	Weeks	Days	Extremely High	High	Faster

V. IC MANAGE PEERCACHE FOR HYBRID CLOUD BURSTING

IC Manage PeerCache utilizes an HPC Cloud Caching approach and works with both primary cloud workflow options. PeerCache enables an HPC hybrid cloud workflow, plus accelerates cloud application I/O performance in a pure cloud workflow.

Below is how it compares with the implementation criteria discussed above.

1. Does not create duplicate workflow copies

You only support one workflow – your existing on-premise one. PeerCache does this by projecting your entire on-premise environment virtually in the cloud, and then transferring only the exact data set needed for a job on demand.

2. Eliminates cloud data storage

Your cloud workflow is stored in PeerCache’s peer-to-peer caching network, rather than in your cloud back-end storage; changes are written back to your on-premise system. When you’ve finished running a job, you can choose to completely erase all data from the cloud.

3. Takes only minutes to set up a cloud job & upload data

PeerCache’s virtual projection of the job meta data to the cloud eliminates the need for days or weeks of copying data. You get nearly instant cloud bursting – plus multicloud flexibility across cloud vendors.

4. Minimizes cloud data download

PeerCache saves all writes as deltas, and then only sends back the data appended to a file to minimize your cloud data download costs.

5. Accelerates cloud application performance

PeerCache’s scale out I/O technology utilizes fast NVMe peer-to-peer caching, meaning that you get HPC for your applications in the cloud.

6. Is compatible with on-premise & cloud infrastructures

PeerCache works with both on-premise shared storage & cloud block storage infrastructures, without loss of performance or additional costs. It works with existing on-premise workflows.

Learn more about [IC Manage PeerCache for Hybrid Cloud Bursting](#).

ABOUT THE AUTHOR



Shiv Sikand, EVP, IC Manage, Inc.

Shiv Sikand is founder and EVP at IC Manage and has been instrumental in achieving its technology leadership in HPC scale out I/O, hybrid cloud bursting, big data analytics, and design and IP management for the past 15 years. Shiv has collaborated with major systems and semiconductor leaders in deploying IC Manage's infrastructure and technologies across their enterprises.

Shiv started in the mid 1990's during the MIPS processor era at SGI, authoring the MIPS Circuit Checker before specializing on design management tools for a number of advanced startups, including Velio Communications and Matrix Semiconductor, before founding IC Manage in 2003. Shiv received his BSc and MSc degrees in Physics and Electrical Engineering from the University of Manchester. He did his postgraduate research in Computer Science, also at Manchester University.

###