Using Ellexus Breeze for EDA Workload Migration to AWS

AWS Whitepaper

Using Ellexus Breeze for EDA Workload Migration to AWS: AWS Whitepaper

Table of Contents

Using Ellexus Breeze for EDA Workload Migration to AWS

Publication date: **June 2020** (*Document revisions* (p. 16))

Abstract

This whitepaper outlines the best practices for migrating Electronic Design Automation (EDA) workloads to AWS using the I/O profiling and dependency analysis tool suite Breeze from Ellexus. Profiling the EDA tool on premises and in the cloud ensures that the AWS configuration is right sized for the application and that costs are optimized.

Introduction

Ellexus provides I/O profiling solutions to monitor, profile, discover, and debug scientific and engineering workloads and high performance computing (HPC) systems.

Ellexus Breeze is an offline analysis tool that profiles workflows to collect file system dependencies as well as CPU, memory, and I/O requirements. This allows you to analyze and migrate legacy EDA tools, and right-size cloud resources for the entire workflow.

The first challenge when migrating a workload to a new environment is making it work. The second challenge is making it work well. Breeze addresses both of these challenges by giving you the information you need to make informed choices and troubleshoot issues quickly.

Although a deep understanding of Electronic Design Automation (EDA) workflows is not necessary, you should have a basic understanding of EDA tools and their underlying infrastructure.

Addressing migration challenges with Breeze

The following metrics are collected by Breeze, which are vital when migrating workloads to the cloud:

- File and network dependencies
- Process tree and scripted infrastructure
- Program arguments and environments

Customers often use a manual process, and leverage the data from Breeze to run an initial proof of concept (POC) on AWS with an environment that is similar to their existing on-premises infrastructure. After the initial POC is completed, the Breeze Command Line Interface (CLI) can be used to automate the process and migrate the remaining workflows.

Application dependencies and containerization

Often, EDA tools can form legacy workflows that make it difficult to determine which files, libraries, and applications are needed. Breeze addresses this issue by automatically detecting which files, libraries, and applications are needed. This can be as fine grain as listing every file, or a more high-level approach can be taken, determining only the mount points that need to be replicated on AWS.

Once the dependencies have been collected with Breeze, you can containerize the workflow. What to include and exclude from the container depends on your long-term data management strategy, and how the container will be used. If the same rules can be applied to multiple workflows, the containerization process can be automated using the workflow dependency output from the Breeze CLI. More details on dependency output are provided later in this whitepaper.

Debugging licensing issues and other changes in environment

Moving an EDA workflow to new infrastructure usually involves changes to the workflow environment. For example, connecting to a license server on AWS might result in quicker access to licenses, and jobs completing faster. Problems that arise could be related to the new environment, the on-premises IT infrastructure, the EDA tool, or the scripted flow that orchestrates it. Breeze helps to debug these problems by capturing the application and how it runs. For example, a file permissions issue would show up as a failed file open. A missing library would be similarly highlighted, and if the tool is looking in the wrong place for a file or license, Breeze will tell you.

Example: Migrating Xilinx Vivado

Let's dive into an example with Breeze. The workflow we are going to migrate runs synthesis on a Xilinx MicroBlaze project.

Topics

On-premises architecture

The on-premises architecture for this example uses an NFS shared file system to host tools and projects. Additionally, there is a second shared scratch space which is used in application processing. Breeze will be used to discover all dependencies that will be needed to migrate the tool and project data to AWS, as well as provide system level information to optimize the AWS architecture (for example, CPU, run time, etc.). Here is an architecture diagram of the on-premises environment:

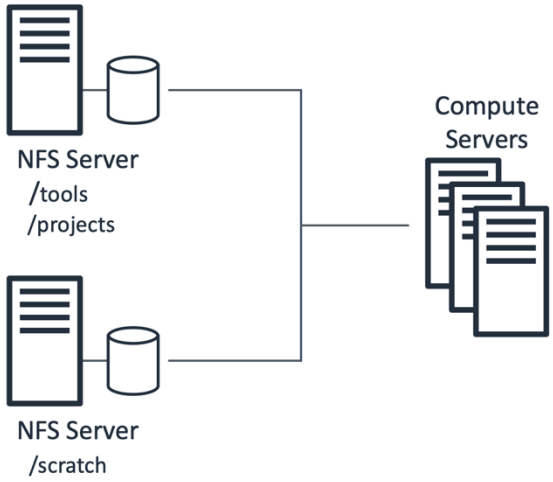

Figure 1: Architecture diagram for on-premise environment

Tracing the EDA workflow with Breeze

Breeze works straight out of the box with no changes to the workflow or the machine. Because it runs in userspace and installs with just an `untar` command, it only takes seconds to get started tracing the EDA workflow.

Breeze can be downloaded from the Ellexus downloads webpage. The *Breeze Installation Guide* and *Breeze User Manual* can be found here: Breeze release notes and manuals. Breeze is installed in the shared file system, ensuring that it's accessible from all hosts. You must complete license configuration before running Breeze. Contact Ellexus for licensing options, or license Breeze directly through the AWS Marketplace.

The Breeze licensing model permits tracing and profiling of applications anywhere in the world on any machine. The obfuscated trace data can be moved to a system with a licensed copy of Breeze for analysis. This model makes it practical to license Breeze through AWS Marketplace and use it to analyze applications that you are running on premises. Simply trace your on-premises workflow and then upload the trace data to your Breeze Amazon Machine Image (AMI) on AWS for analysis.

The following command will trace the EDA workflow and store the trace data in `/tmp` for post processing.

```
$ cd /nfs/workflows/xilinx/microblaze/xcvu9p-flga2104-2L-e

$ <Breeze install>/trace-program.sh -f /tmp/trace.out ./run_synth.sh
```

Interpreting Breeze trace output from the EDA workflow

The trace output from the Vivado synthesis run is viewed with the Breeze interface and is shown in Figure 1. In the Breeze **I/O Summary** view, the workflow duration is displayed for each of the I/O types (Good, Medium, and Bad). In this case, a lot of time is spent doing reads of 32 KB or smaller. This I/O profile is common for EDA tools that often need to access small files. This is supported by the amount of time being spent opening files, from which only a small amount of data was read or written. The table at the bottom lists the processes in order of utilization, and shows that most I/O was performed by the EDA tool itself.

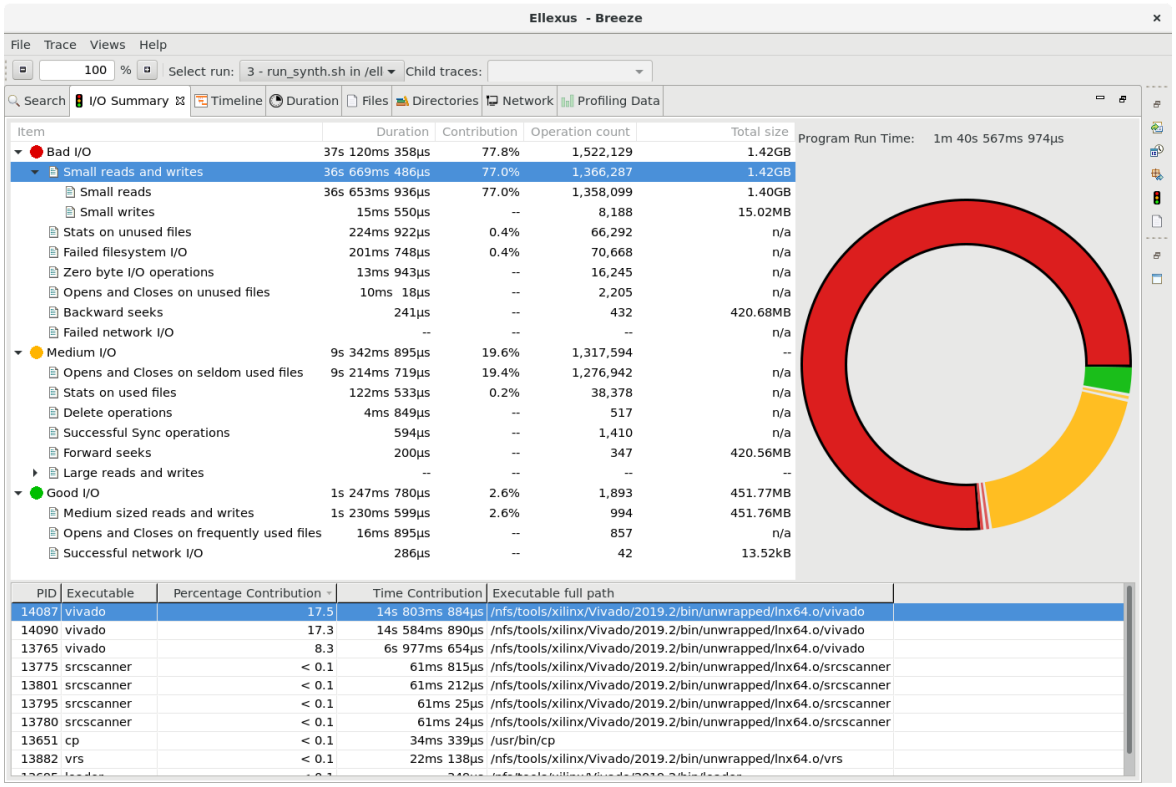

*Figure 2: Breeze **I/O Summary** view shows time spent in I/O & processes that contributed most*

Switching to the Breeze **Timeline** view, gives us the full process tree showing the environment preparation performed by the scripted infrastructure and the call into the EDA tool itself. Clicking on each process gives us the arguments and environment of each process, as shown in Figure 3.

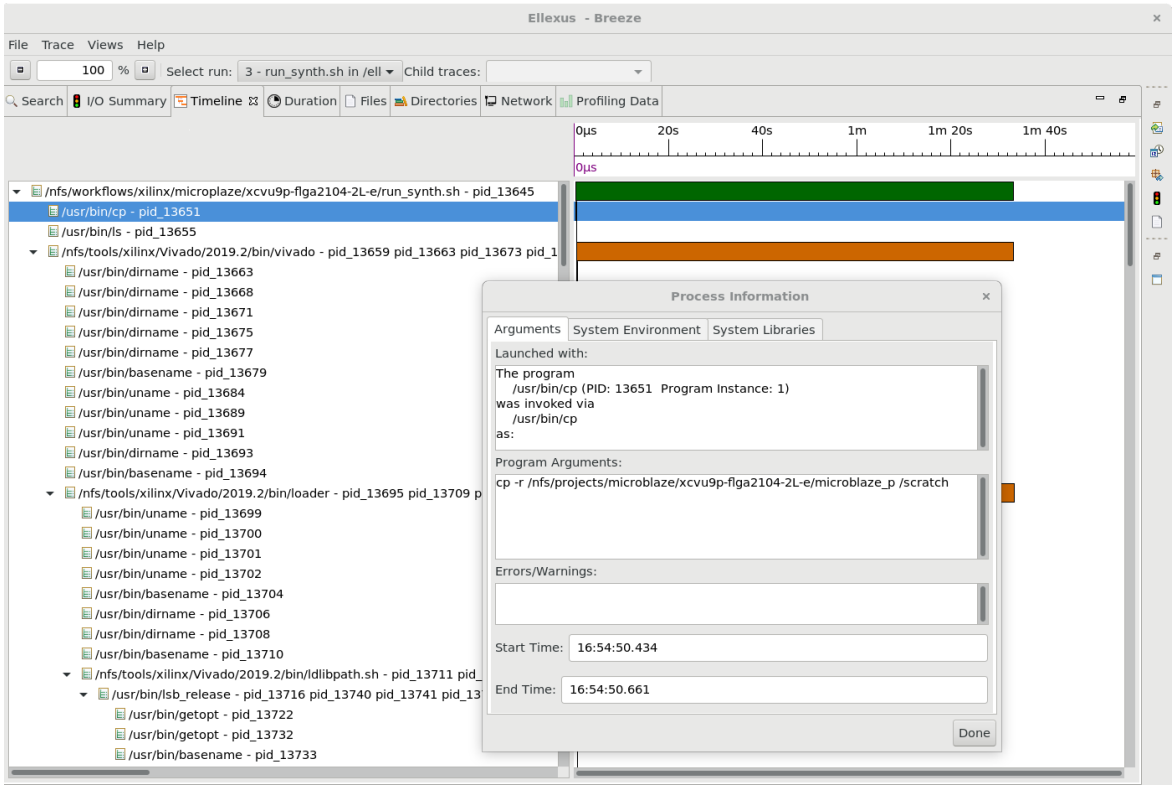

*Figure 3: Breeze **Timeline** view showing the process tree and arguments of the selected process*

Extracting workflow dependencies

The application dependencies can be found by switching to the Breeze **Files** view, as shown in Figure 4. This table displays every file that has been accessed by the EDA tool. It also displays the mount point and, if additional options are set at trace time, it also shows the Linux package needed to install the dependency, if applicable. Scrolling right will show every operation that has been performed on each file, the performance of that I/O, and whether the operation was successful.

The files view for this project tells us that:

- the EDA tool is installed in `/nfs/tools/Xilinx`
- the Xilinx MicroBlaze IP is contained in a subdirectory of `/nfs/projects/`
- the workflow scripts are in `/nfs/workflows`
- the workflow uses `/scratch` for temporary files

This table shows you where an application is spending its time, but if all you need are its dependencies, you can export this list using the Breeze CLI. With this data, everything needed to replicate the on-premises infrastructure on AWS is now available.

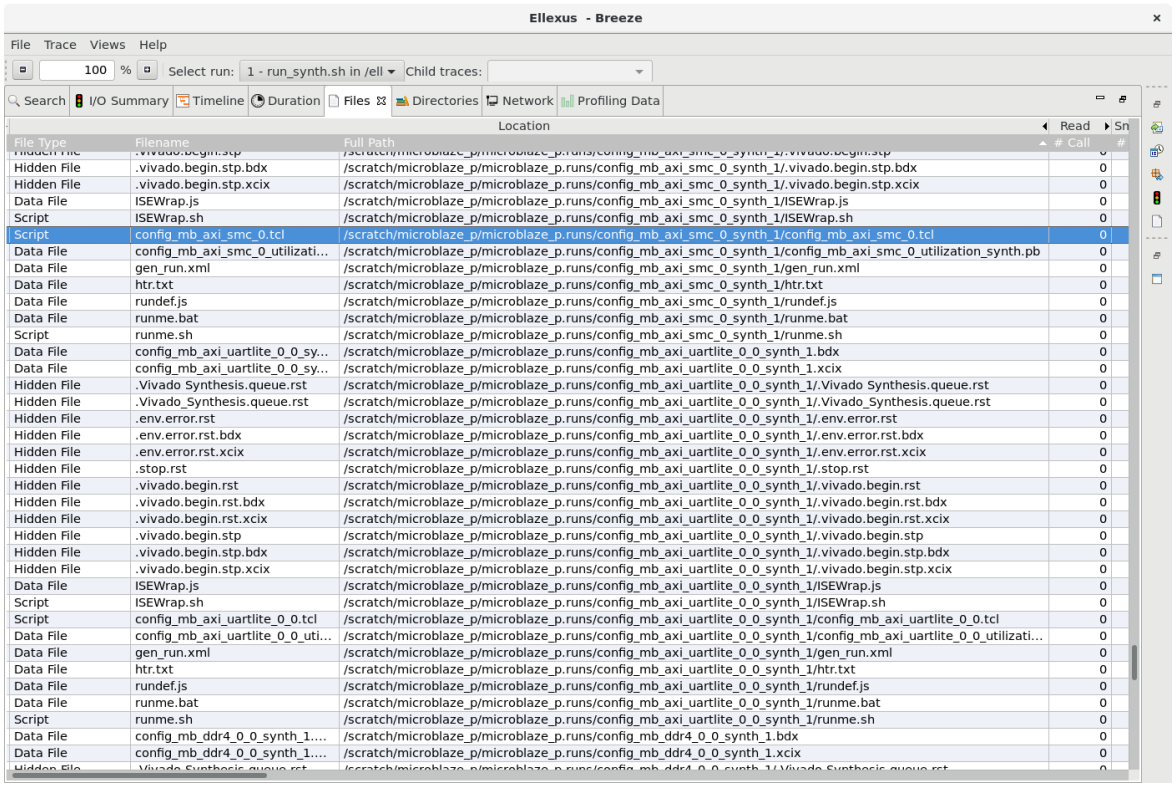

*Figure 4: Breeze **Files** view showing a list of every file used and its I/O patterns*

Right sizing and performance

EDA tools are known for being CPU and I/O intensive. Figure 5 shows the I/O patterns, CPU, and memory usage for the EDA tool. At different times of the synthesis run, the EDA tool is I/O, CPU, and memory intensive. Applications that perform many small read operations, as this one does, are often both CPU and I/O bound as they are limited both by the storage latency and the speed of the CPU to issue the I/O requests. This is supported by the CPU pattern, which hovers between 200% and 400%, while the number of I/O operations is about 80K IOPS.

This data implies that the AWS Cloud infrastructure should have several fast CPU cores and a storage solution that can perform at 80K IOPS.

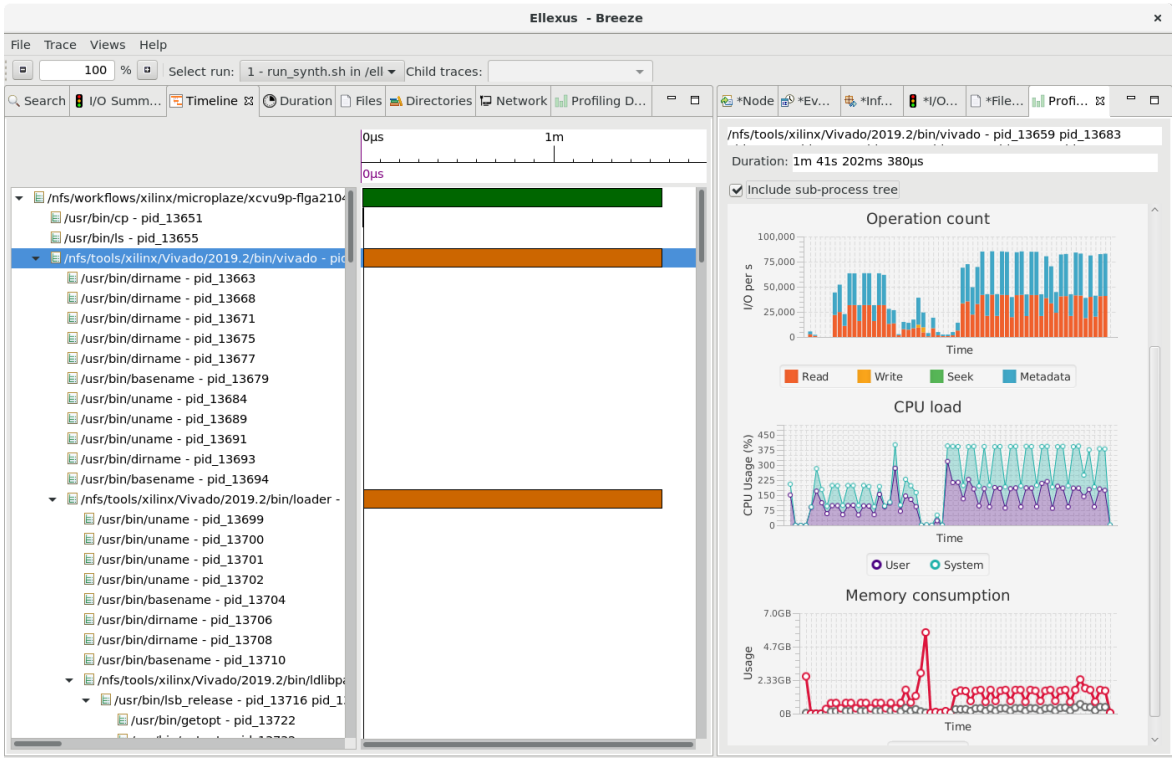

*Figure 5: Breeze **Timeline** view shows the I/O, CPU and memory of the EDA tool over time*

The Breeze **Profile** view shows I/O, CPU, and memory usage over time. This view includes a detailed breakdown of storage performance and the impact that has on the workflow. Figure 6, taken from the **Profile** view, shows how much time is spent in read operations per second. This workflow can spend over 500ms per second waiting on read calls. It's likely that the workload is multi-threaded so that many operations can be issued in parallel, and could benefit from having fast local storage. Some experimentation is recommended to determine which storage option gives you the best return on investment.

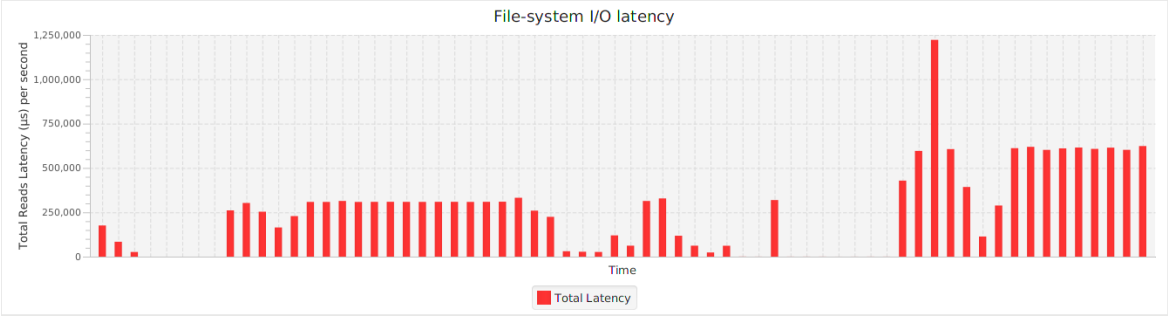

*Figure 6: Read latency over time, taken from the Breeze **Profile** view*

Cloud architecture

While the lift and shift approach can get you started on AWS quickly, making small changes to workflows allows you to advantage of native AWS services, resulting in cost reduction and getting results quicker. In this case, the workflow is straightforward and Breeze identifies which scripts need to be updated when migrating to AWS.

We have chosen an Amazon EC2 z1d instance. z1d instances deliver high single thread performance due to a custom Intel Xeon Scalable processor with a sustained all-core frequency of up to 4.0 GHz—the fastest of any cloud instance at this time. The z1d provides both high compute performance and a large amount of memory, which is ideal for EDA workflows. Additionally, the z1d instance has an Amazon EC2 instance store, which are local NVMe SSD drives local to the instance. The z1d instance fulfills our CPU and storage requirements, and we can adjust the configuration later to optimize for cost and performance.

The project and workflow have been packaged and stored in Amazon S3. This is a cost-effective way of storing large amounts of project data. When migrating this EDA tool to AWS, a new workflow was configured that pulls the project data from Amazon S3 and then runs the workflow using a local instance store.

Using FSx-Lustre for high performance shared storage

For this example, there are no shared file system dependencies when running on AWS. If the workflow had dependencies on shared file systems, using Amazon FSx for Lustre can provide the necessary performance to run EDA workflows on AWS. Additionally, the project data and results can be stored in Amazon S3, and before the instances are launched, the FSx-Lustre file system can be populated with the data from Amazon S3. This eliminates the need for a local instance store. It would then be possible to terminate the compute instances and post process the results in bulk from the shared storage.

Reference Architecture

Here is a reference architecture showing the flow of running Breeze tracing Xilinx Vivado on AWS:

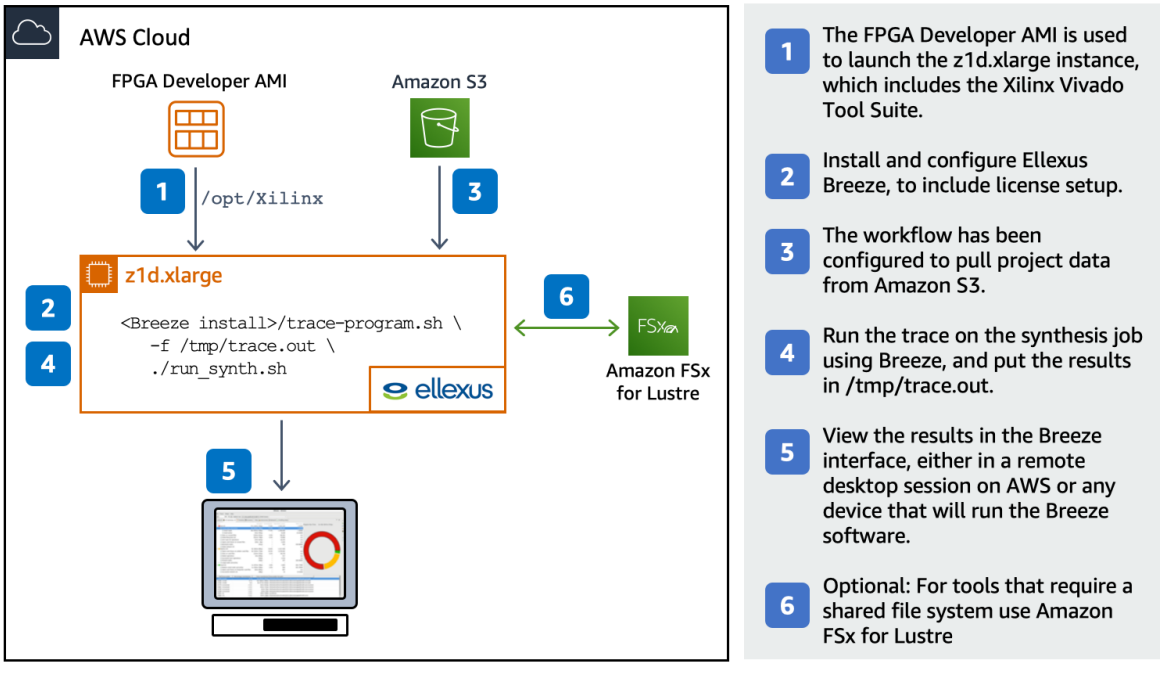

Figure 7: Reference architecture showing running Breeze tracing Xilinx on AWS

Profiling the workflow on AWS

Running Breeze on the new workflow on AWS allows us to see the architecture changes. The AMI has the Xilinx tool installation and the top-level workflow script needed to start the workflow. This script calls the AWS CLI (pre-installed with the image) to pull the project data from S3 object storage. The workflow is then run using a local instance store.

Figure 8 shows the new timeline with the calls to the AWS CLI. Breeze shows you I/O statistics and the dependencies for each process in the workflow. The view on the left shows the entire workflow, and the view on the right shows a detailed breakdown of what the AWS CLI is doing. You can see the connections to the object storage.

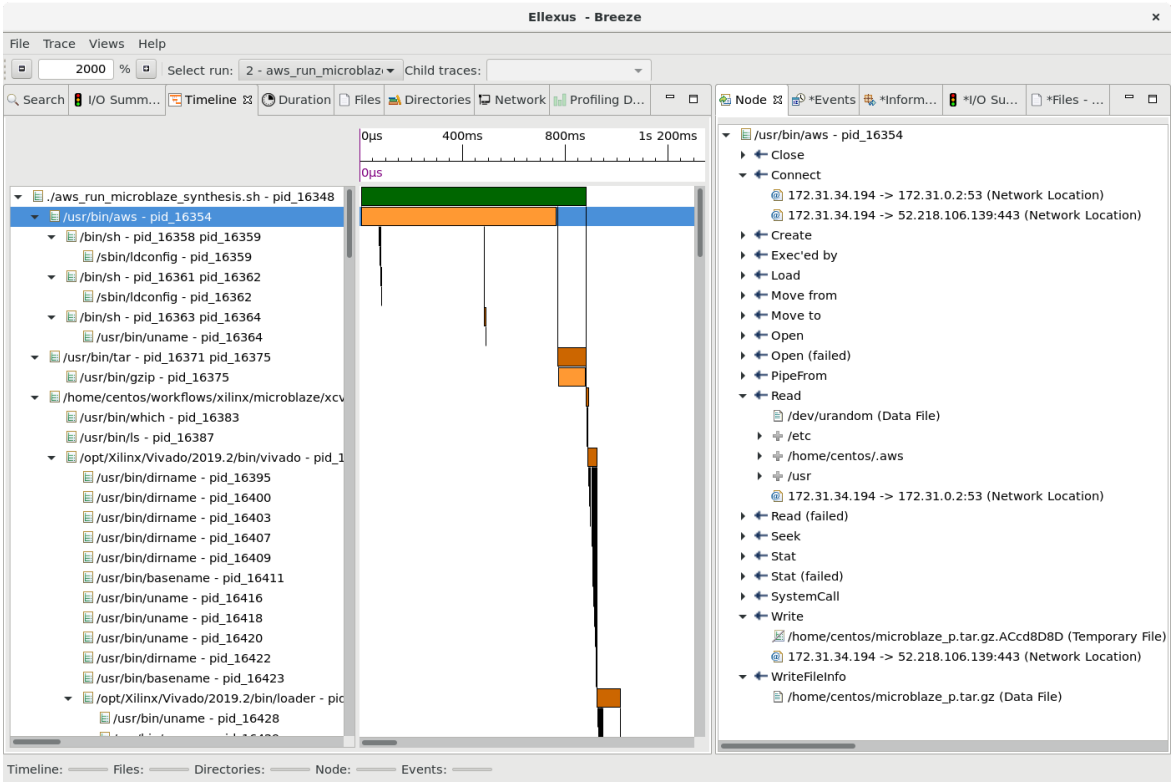

*Figure 8: Breeze **Timeline** view showing details of operation performed by the AWS CLI to Amazon S3*

Checking application correctness

Application correctness is about making sure that the workflow is consistent across all platforms. For example, are all the dependencies and library versions the same, is the EDA tool being called with the same arguments? In EDA, accountability and reproducibility are vital, so it's critical to ensure that workflows are not just working, but working correctly. By capturing arguments, the environment of every process in a scripted workflow, and all the dependencies, Breeze can verify that the application is running in exactly the same way on AWS, and reading the same files as it does in an on-premises environment.

In Figure 9, we can see that the version of Xilinx Vivado is the same as the version we used on premises. We can also export the file dependencies and verify that any differences are not functional differences.

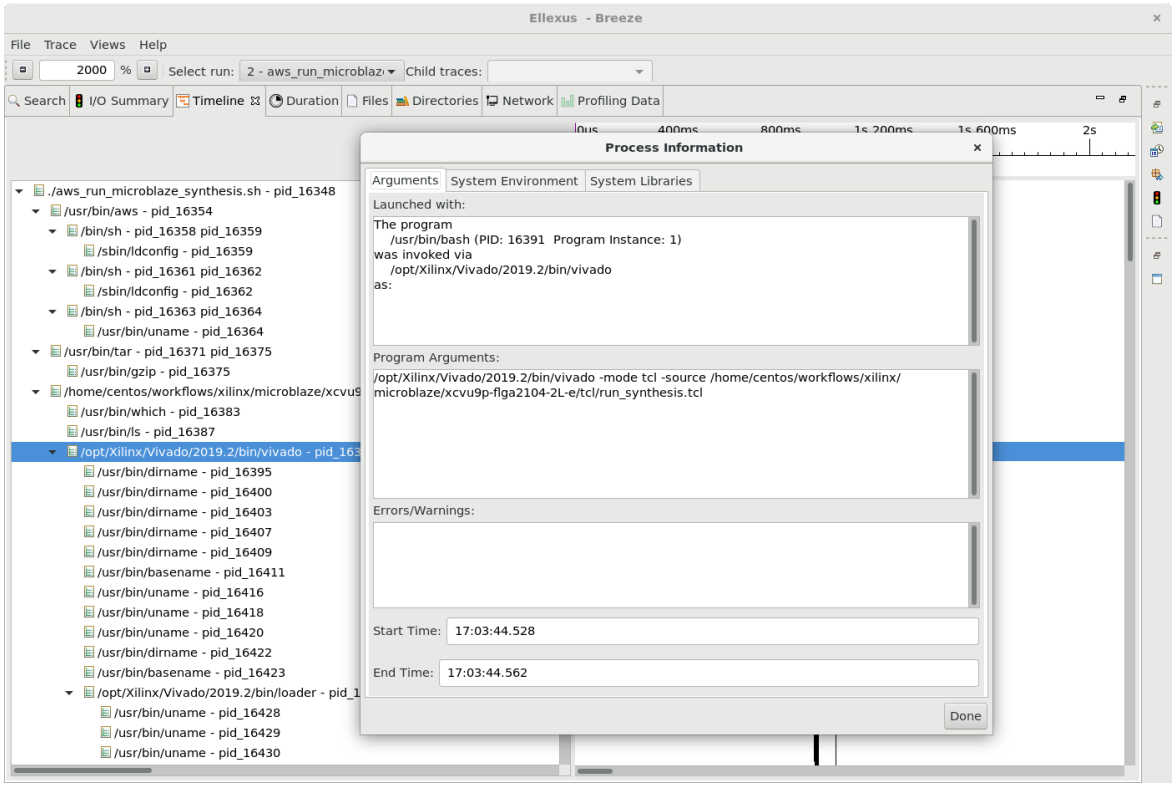

Figure 9: Version check and file dependency verification

Automating the migration process with the Breeze CLI

Once the method of migrating workloads to AWS has been established, the Breeze CLI can be used to automate your go-to-cloud strategy. The Breeze CLI, also known as the Breeze Automation Platform, enables the automated export of all the data available in the GUI as plain text, CSV, or XML output. The following command exports a simplified list of file and mount point dependencies, but you can also export the I/O profiling data and other reports from Breeze.

```
$ <Breeze install>/breezeAP.sh <path to>/trace.out -y \
     dependencies -f csv -o <path to output>
```

To get a complete list of export options, refer to the *Breeze User Manual*, or simply run the Breeze Automation Platform script, `breezeAP.sh`, with no arguments.

Conclusion

The semiconductor and electronics industry is migrating many EDA tools, as well as entire chip design workflows, to AWS. The example used in this paper is a typical EDA tool example, showing how Ellexus Breeze packages an EDA tool for migration and reduces the time to move workflows to AWS.

Contributors

Contributors to this document include:

- Dr. Rosemary Francis, CEO, Ellexus Ltd
- Mark Duffield, Tech Lead for Semiconductor and Electronics, AWS
- Faraz Angabini, Solutions Architect, AWS

Further reading and information

For additional information, see:

- Introduction to semiconductor design workflows on AWS
- Ellexus Products on AWS Marketplace

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

update-history-change	update-history-description	update-history-date
Initial publication (p. 16)	Whitepaper published.	June 11, 2020

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.