

"I saw an 87% improvement in run time!"

Ahmed Ismail, software engineer at Mentor Graphics

At Mentor Graphics, Breeze was used to speed up a software build by removing calls to bash and pwd. Thousands of calls from a Makefile were using more than 80% of the run time.

Why should you care about I/O?

Your compute is speeding up, but your systems are slowing you down. As systems grow, it's vital that you take control of the way you access your data. Inefficiencies in the I/O can be introduced by third-party applications, user errors or in the way applications are deployed. Mistakes such as storing data in the wrong place, small I/O or excess meta-data can overload compute, network and storage. By using Breeze and following our best practices, you will fix short-term problems, understand storage requirements and plan for the future.

Use Breeze to become fast, agile and cloud ready

Breeze gives you the information you need to solve deployment and performance problems quickly and tune your applications for your IT setup whether on-prem or in the cloud. It forms a common language for IT managers, users and software vendors to work together, taking the guesswork out of issue resolution. Breeze tells you:

- Which files and network locations your program has accessed
- How much data has been read or written
- How many meta-data operations such as move() or create() your program has performed
- ... and much more

Profiling file I/O with Breeze

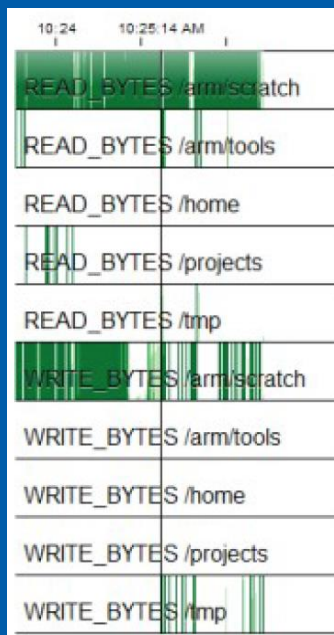
Even when programs are designed well, it is still easy for local settings to affect performance. Files stored in the wrong place can affect your application and everything else on the cluster. Breeze profiles application file I/O so you can ensure you are storing files in the right place.

Trace application dependencies

Breeze automatically detects which files, programs and libraries have been used by an application to create a bill of materials for each run. This can be used to solve deployment issues or to prove the correctness of a deployment.

Solving installation and migration issues

Often find yourself running strace and writing your own scripts to analyse the results? Stop and switch to Breeze now! Commonly, a program will work on one machine, but not on another, or for one user and not another. With Breeze you can quickly compare the two, saving hours of engineering time.



Customer case study: A rogue job at Arm

It only needs a small number of badly behaving jobs to overload shared file systems and cause system bottlenecks, which this application is doing by putting data in the wrong place. Temporary data is being saved to shared storage instead of local storage, which slowed everyone's work on the cluster.

In this case, Breeze highlighted an easy fix that increased performance without rewriting the main application.



Gain a unique view of your data

Breeze's unique traffic light view provides a quick, clear snapshot of how your job is performing. Seeing how much is good I/O and how much is bad lets you zoom in on the problems quickly.

One clear GUI

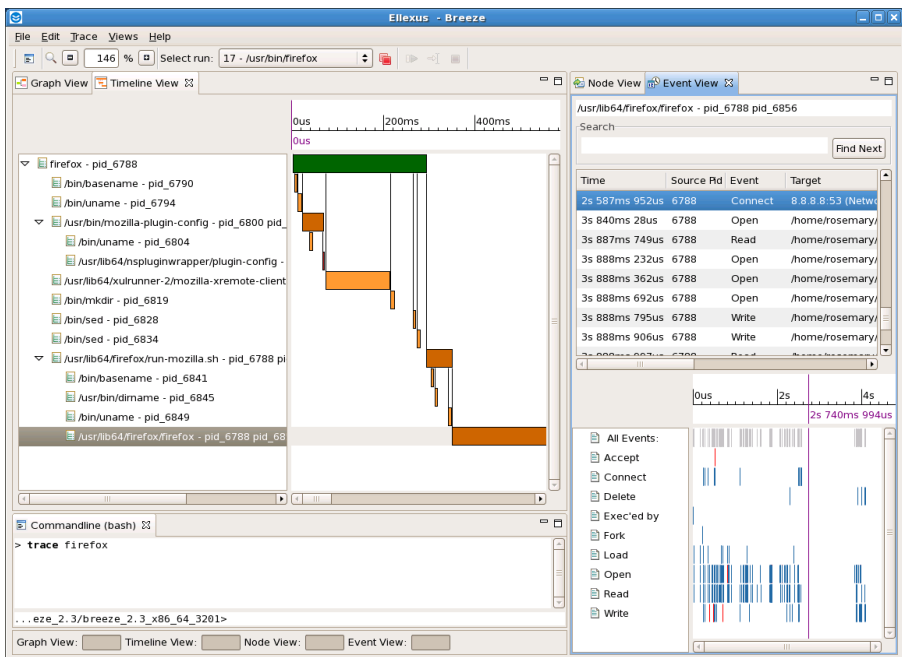
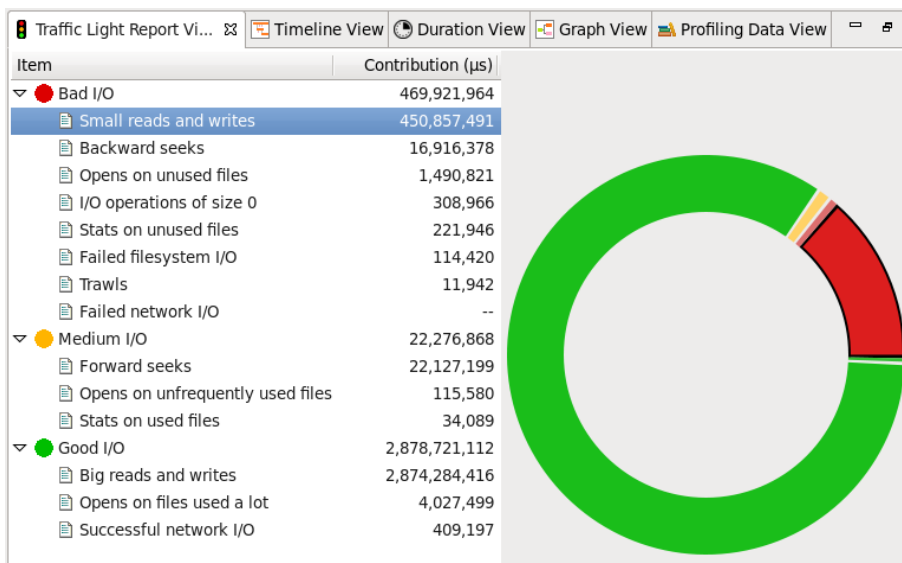
Breeze's graphical user interface offers multiple views that allow you to explore your data and drill down to find the information that you need.

A command-line API

Automate checks to improve the quality of service that you offer to your team. By integrating Breeze into your working practices, you can solve problems before they become a problem.

Improve communication with non-expert cluster users

Breeze's Healthcheck report creates a simple resource that you can give to non-expert cluster users, detailing straightforward I/O issues.



Prepare for your next migration with Breeze

Whether you're heading to the cloud or to a new storage architecture, Breeze will provide invaluable information for your migration strategy. Use the list of dependencies provided by Breeze to solve installation problems and migrate your applications to the cloud or a container platform easily. Analysis on resource use and I/O patterns lets you size storage requirements easily.

Customer case study: HPC and customer support at ANSYS

"Breeze allows us to trace our software at customer sites, making it easy to look up which libraries have been found and which ones are missing. I was impressed with what it showed us and the time it will save us in troubleshooting these sorts of issues"

**Simon Cross, Manager
Global Installation Support, ANSYS**



Setting up the licence

Breeze and the Ellexus Healthcheck both use the RLM licence server.

Tracing an application

Both Breeze and Healthcheck use the same trace file to generate the report:

```
./breezeAP_<version>_x86_64/trace-program.sh -f /tmp/tracelog app_to_trace_and_args
```

You can configure the data that Breeze collects using the environment variables in:

```
./breezeAP_<version>_x86_64/profiling-options.sh These need to be set before the trace is run.
```

A trace-only version of Breeze is available as a small download to make it easy to trace applications remotely and bring the results back to your own environment for analysis. This makes Breeze a great choice for solving customer service issues.

Breeze command-line API

Breeze comes with a command-line API called the Breeze Automation Platform. This lets you export the data as xml or CSV for automatic processing. You can write your own custom queries or integrate Breeze into your test framework. The following example exports a list of dependencies from a set of traces as a plain text list:

```
./breezeAP_<version>_x86_64/breezeAP.sh '/path/to/app_trace[0-9]*' -y dependencies -f txt -k -o /path/to/outputdir
```

Generating the Healthcheck report

The Healthcheck report can be generated from a Breeze trace or you can trace your application and generate the trace in one go.

```
./ellexus-healthcheck_<version>_x86_64/ellexus-healthcheck.sh -f ~/tmp/HCreport /tmp/tracelog  
./ellexus-healthcheck_<version>_x86_64/ellexus-healthcheck.sh -f ~/tmp/HCreport app_to_trace_and_args
```

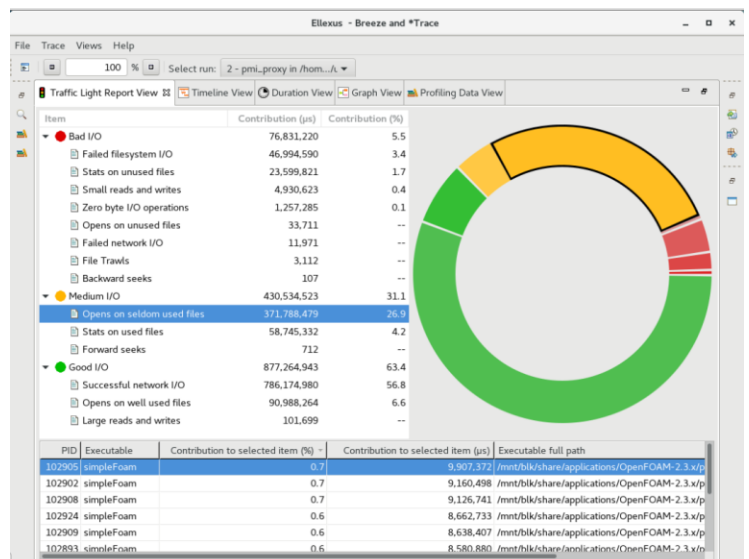
The Healthcheck report will be generated inside the output directory specified. It will include an html report as well as lists of program, file and network dependencies.

Getting started with the Breeze GUI

Launch Breeze with the following command and import your trace from the file menu:

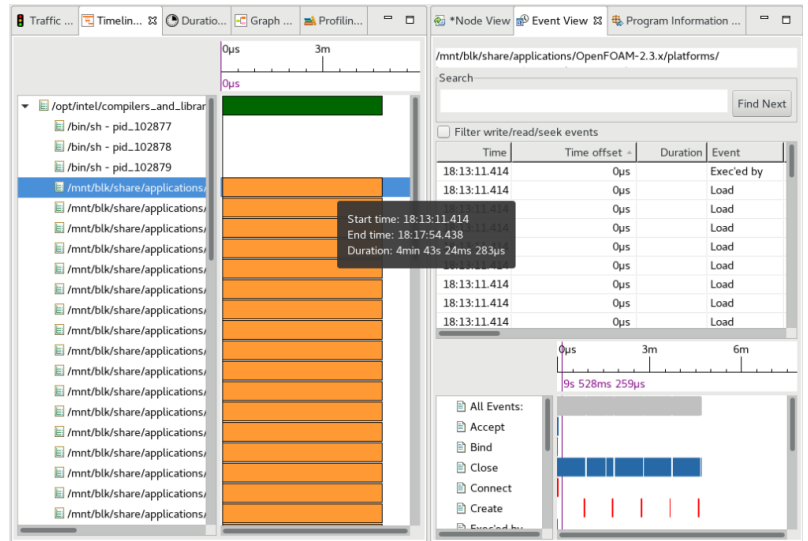
```
./breezeAP_<version>_x86_64/breeze.sh
```

The traffic light view gives you an overview of how much time is spent in good or bad I/O.



To find a particular program within your application you can search for it or select it in the Timeline view.

The node view and event views then show more information for the selected process. You can see lists of network connections and file dependencies or a table of events in chronological order.



Full Path	# Call	Small (< 4kB) Read	Large (≥ 100MB) Read	Write	Small (< 4kB) V
/etc/passwd	1,250	1,250	0	0	
/mnt/blk/share/applicat...cts/setConstraintTypes	400	200	0	0	
/sys/devices/system/node/node1/distance	153	51	0	0	
/sys/devices/system/node/node0/distance	153	51	0	0	
/mnt/blk/share/data/Op...t/polyMesh/pointZones	100	100	0	0	
/mnt/blk/share/data/Op...nt/polyMesh/faceZones	100	100	0	0	
/mnt/blk/share/data/Op...nt/polyMesh/cellZones	100	100	0	0	
/sys/devices/system/node/node1/cpulist	52	51	0	0	
/sys/devices/system/node/node0/cpulist	52	51	0	0	
/sys/devices/system/node/node0/cpumap	50	50	0	0	
/dev/shm/Intel_MPI_...f9aeTC	50	0	0	1	
/dev/shm/Intel_MPI_...ZDJKTS	50	50	0	1	
/sys/devices/system/node/node1/cpumap	50	50	0	0	
/mnt/blk/share/data/Op...Bike/processor104/0/U	10	1	0	0	
/mnt/blk/share/data/Op...Bike/processor110/0/U	10	1	0	0	
/mnt/blk/share/data/Op...Bike/processor130/0/U	10	1	0	0	
/mnt/blk/share/data/Op...Bike/processor141/0/U	10	1	0	0	

If you are planning an application migration or checking the correctness of an application, you should look in the Files view. Here you can discover every file that has been used, as well as an overview of the I/O patterns for that file.

The Profiling view gives information over time for file and network data. You can see IOPs as well as a histogram of I/O operation latency.

As well as a summary of all I/O Breeze will automatically gather sperate statistics for each mount point and network connection.

