The dataset is produced by the software XALT, (https://github.com/Fahey-McLay/xalt) which tracks and collects information about MPI_based software libraries and executables used on high performance computing systems, also known as supercomputers. The XALT data is used to generate statistics about the system in which the XALT software is installed, specifically libraries and applications usage. This curated and published XALT dataset, in JSON format, contains information on the number of nodes, libraries, and executables used by each user running a given computational job on Stampede (https://www.tacc.utexas.edu/stampede/), a supercomputer deployed and maintained at TACC (https://www.tacc.utexas.edu/). This data does not capture other types of jobs, such as non-MPI jobs and serial jobs. Also, the data has information about community libraries, so the names of personal codes have been replaced by a randomly generated hashes. For privacy purposes the users names have been anonymized and replaced by a User id. In the data dictionary below, each of the elements in the dataset are defined.

full name	**description**	**datatype**	**null**	**example**
allocation	The administrative account that that users run jobs against and a user may have multiple allocations. The string has been anonymized for publication. This string is consistent across the entire dataset.	string	Yes	A00084719
date	The job’s start date and time. Format, YYYY-MM-DD HH:MM:SS.	string	no	2015-04-01 00:16:21
exec_path	Mapped codes that represent the user’s executable path. If the executable is a known community code then the name of the real executable is mapped to a list of	string	no	WRF*

	executable names mappings. Denoted by an asterisk by the path code. A dictionary of the community codes is available as a separate document: Identified_Community_Codes. If the executable is not a common one then the executable name (with the path removed) is converted to a sha1 string. For example, any user running the "SamBill" code will have the same sha1 string.			
field_of_science****	The scientific domain of the project. This information is integrated to the XALT data from TACC's accounting information. When users create an allocation request to use a national open science computational resource, they have to describe the project for which they will use the resource. This includes the field of science to which the project belongs to. Users select the field from a list of scientific domains prepared by the National Science Foundation. Note that the list has general classes of sciences. Thus, mapping from a class of science to the science/s involved in the problem to be resolved may not be precise.	string	Yes	biology***
host	The computing resource in which	string	no	Stampede

	the job is ran.			
job_id*	Identifies a user’s job. The job_id may be associated with multiple runs. This string is consistent across the entire dataset.	string	no	4922626
linkA	Collection of libraries used to complete instructions in executable files. Format, structured series of arrays with two values: module_name and path.	string	no	[{“library_module_name”: null, “library_path”: “/lib64/libc-2.12.so”}]
library_module_name	Software libraries that define functions which allow the executable to run. part of linkA	string	yes	“library_module_name”: null
library_path	The directory location and name of a library used during the job. part of linkA	string	no	“library_path”: “/lib64/libc-2.12.so”
module_name	Executables maintained by TACC that users employ	string	yes	valgrind/3.8.1
num_cores	The number of processors used during a job.	integer	no	48
num_nodes	The number of nodes used during a job.	integer	no	3
num_threads	The number of threads that make up a job.	integer	no	1

run_time**	The number of seconds it took to complete the job.	float	no	11.79
start_time	The time the job started formatted in Unix time.	float	no	1427864961.87
user*	Anonymized unique user id for the account’s owner. This string is consistent across the entire dataset.	string	no	U00361810
build_user	The user who built the executable. The user and build_user may not be the same. There are three possible values: system, the user_id, and unknown. system - the executable was built by a TACC staff member. The user and build_user may not be the same. user_id - the executable was built by a user. The user_id will match the anonymized user id that represents the account’s owner. unknown - the builder of the executable is unknown	string	no	system
build_time	When the program was built formatted in Unix time. The time	float	no	1427864951.87

	could be seconds or years before start_time.			

*There are some users that wipe the environment prior to working as a part of their scientific practice. For those jobs, the user, job_id, and other environmental variables may not be available. When the job_id is unknown, the run_time will be 0.0.

**If a job does not finish in the allocated time, run_time will equal the designated time from accounting. The run_time will only be 0.0 if the job_id is unknown.

***If the user wipes the environment, the field of science will be null

****The field of science is self-selected by the users and should apply to the project domain (which may not be the same as the user's identified domain). The field of science is also restricted to one entry, so multidisciplinary projects are not captured in the dataset.

Definitions

library – a collection of self-contained component of a program (module), with a well-defined purpose and boundary.