Analysis of software package usage based on methods for identifying similar HPC applications

Denis Shaikhislamov, Vadim Voevodin
Research Computing Center of Lomonosov Moscow State University,
sdenis1995@gmail.com

27 September, 2021
Moscow

Background

- *Moscow State University has Lomonosov-2 supercomputer (#2 in Russia, 4.9 PFlop/s peak, ~1700 nodes).*
- *Supercomputers provide a lot of useful information about jobs: performance data, application packages used, resource manager information etc.*
- *We can analyze this data to find features and patterns in these applications.*
- *Analyzing this data can help detect similar applications, that can allow us to solve other important tasks for the administrators.*

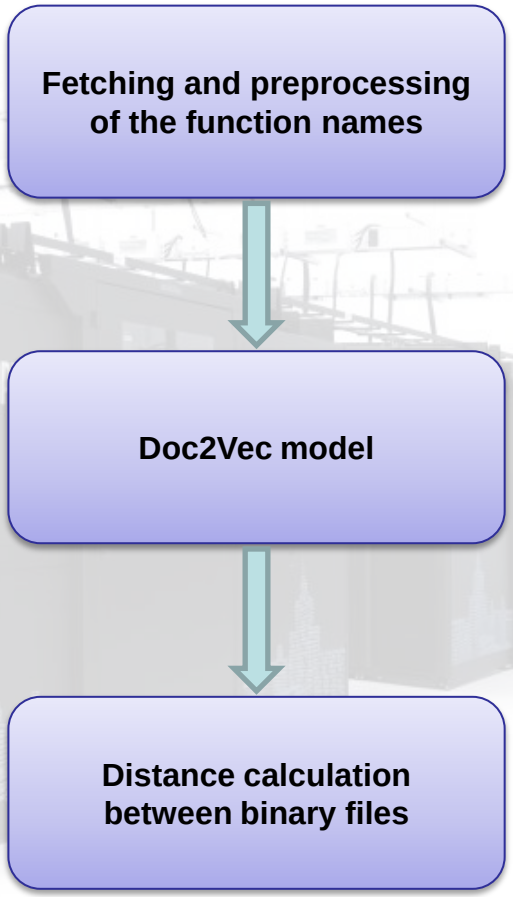
Package detection

- *There are variety of program packages that are used by supercomputer users, like GROMACS, NAMD, NWChem, FlowVision etc.*
- *Package usage detection can provide insight on what is mostly used, what are the needs of users.*

- *There is an existing package detection system based on XALT.*
  - *It replaces linker (ld) and launcher (mpirun) which allows it to see what libraries were linked, what executable files were launched etc.*
- *Sometimes it has troubles detecting packages:*
  - *custom builds (renaming paths/executables);*
  - *just doesn't process some jobs due to failures.*
- *Similar application detection system can be used to solve the problem.*

Static method

- *Input data – binary file information.*
- *Function and variable names (terms) are extracted by "nm" UNIX utility.*
- *Doc2vec model was used to vectorize terms.*
- *Cosine similarity was used for vector comparison.*
- *Accuracy – 0.9+ compared to XALT.*



Using static method for package detection

- *We can assume that similar applications also use same packages.*

Algorithm:

- *When job is launched, we can extract used terms and get its representation using Doc2Vec model.*
- *By comparing it to jobs from knowledge base, we can find similar applications.*
  - *Knowledge base consists of computed representation of a job and its package.*
  - *It contains only unique binary files of specific package.*
- *If we find similar job with a package, we assume that new job also uses that package.*

Current state of static method

- *The static method is running and processing jobs of Lomonosov-2 supercomputer in real time.*
- *Each day, administrators get an email containing all cases with detected packages, both by static method, and XALT.*

Stat2 4:00 AM (15 hours ago)

Job id	Username	unique	doc2vec	xalt	Command	Workdir	Comment
[illegible]	[illegible]	nwchem	nwchem	nwchem			
[illegible]	[illegible]		firefly		[illegible]	[illegible]	
[illegible]	[illegible]		cp2k		[illegible]	[illegible]	
[illegible]	[illegible]	nwchem	nwchem	nwchem			
[illegible]	[illegible]			vasp	[illegible]	[illegible]	[Errno 13] Permission denied: '/opt/ccoe/vasp-5.4.4/bin/vasp_gpu'
[illegible]	[illegible]	nwchem	nwchem	nwchem			
[illegible]	[illegible]	nwchem	nwchem	nwchem			
[illegible]	[illegible]	lammps	lammps	lammps			
[illegible]	[illegible]	lammps	lammps	lammps			

Daily digest of jobs with packages

Evaluation of static analysis

- We processed all the jobs from January to May 2021.
- 27k jobs have been detected using software packages.
  - That's ~22% of total launched jobs in this period (125k).
  - In terms of CPU hours used, these jobs take 45.6% of total CPU hours.

Package	Static + XALT	XALT	Static	Overlap (% of total)	Static excluding XALT (% of total)
LAMMPS	6066	4207	4676	46.4	30.6
NWChem	2421	1469	1164	8.8	39.3
CP2K	4860	818	4778	15.1	83.1
Amber	422	219	334	31	48.1
Gromacs	6072	4334	4828	50.9	28.6
cabaret	318	77	282	12.9	75.8
MOLPRO	973	973	0	0	0
Firefly	1729	742	1636	37.5	57.1

Examples of XALT and static method of package detection

Evaluation of static analysis

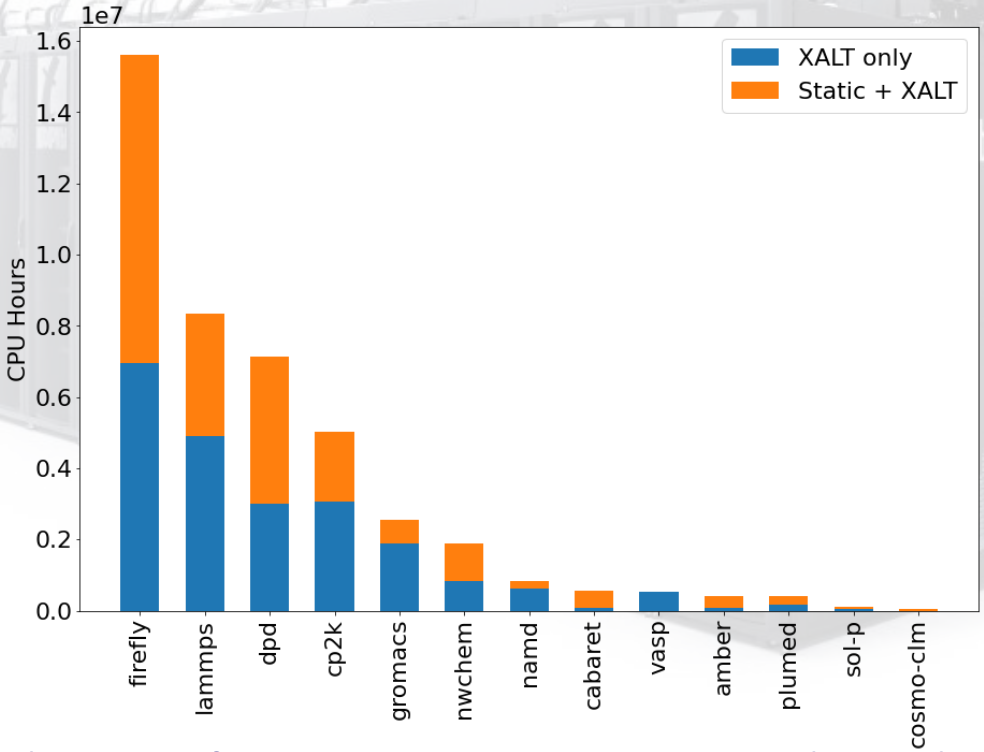
- *Static method helped find ~80% more jobs, than XALT alone.*
- *Sometimes static method can't detect packages, but XALT can, which may be because of:*
  - *insufficient rights to binary files;*
  - *binary file was changed before processing occurred;*
  - *can't extract function names from binary file (MOLPRO).*

Package	*Static + XALT*	*XALT*	*Static*	*Overlap (% of total)*	*Static excluding XALT (% of total)*
LAMMPS	*6066*	*4207*	*4676*	*46.4*	*30.6*
NWChem	*2421*	*1469*	*1164*	*8.8*	*39.3*
CP2K	***4860***	***818***	***4778***	***15.1***	***83.1***
Amber	***422***	***219***	***334***	***31***	***48.1***
Gromacs	*6072*	*4334*	*4828*	*50.9*	*28.6*
cabaret	***318***	***77***	***282***	***12.9***	***75.8***
MOLPRO	*973*	*973*	*0*	*0*	*0*
Firefly	***1729***	***742***	***1636***	***37.5***	***57.1***

Examples of XALT and static method of package detection

Evaluation of static analysis

- *Static method also showed real package's CPU hours utilization.*
- *Some packages use twice as much CPU hours than we knew before.*
- *Sorting packages using new information also changes some of their positions.*



Distribution of CPU Hours spent among jobs with packages

Package version detection problem

- *We can use extracted function names and static method to detect also the version of the package.*

- *We have multiple prebuilt versions of the most popular packages: LAMMPS, NWChem, Gromacs.*
- *There are two ways of approaching the problem:*
  - *We can use unique version's function names to distinguish them among themselves.*
  - *We can use static method, but instead of pairing package to executable's representation we pair package versions.*

- *Approach with unique function names is more transparent and computationally easier.*

Package version detection problem

Algorithm:

- *When we detect that new job use a specific package, we launch package version detection algorithm.*
- *We check whether new job's executable file contains unique function names of different package versions.*
- *If such version is found – we assume that new job uses that specific package version.*

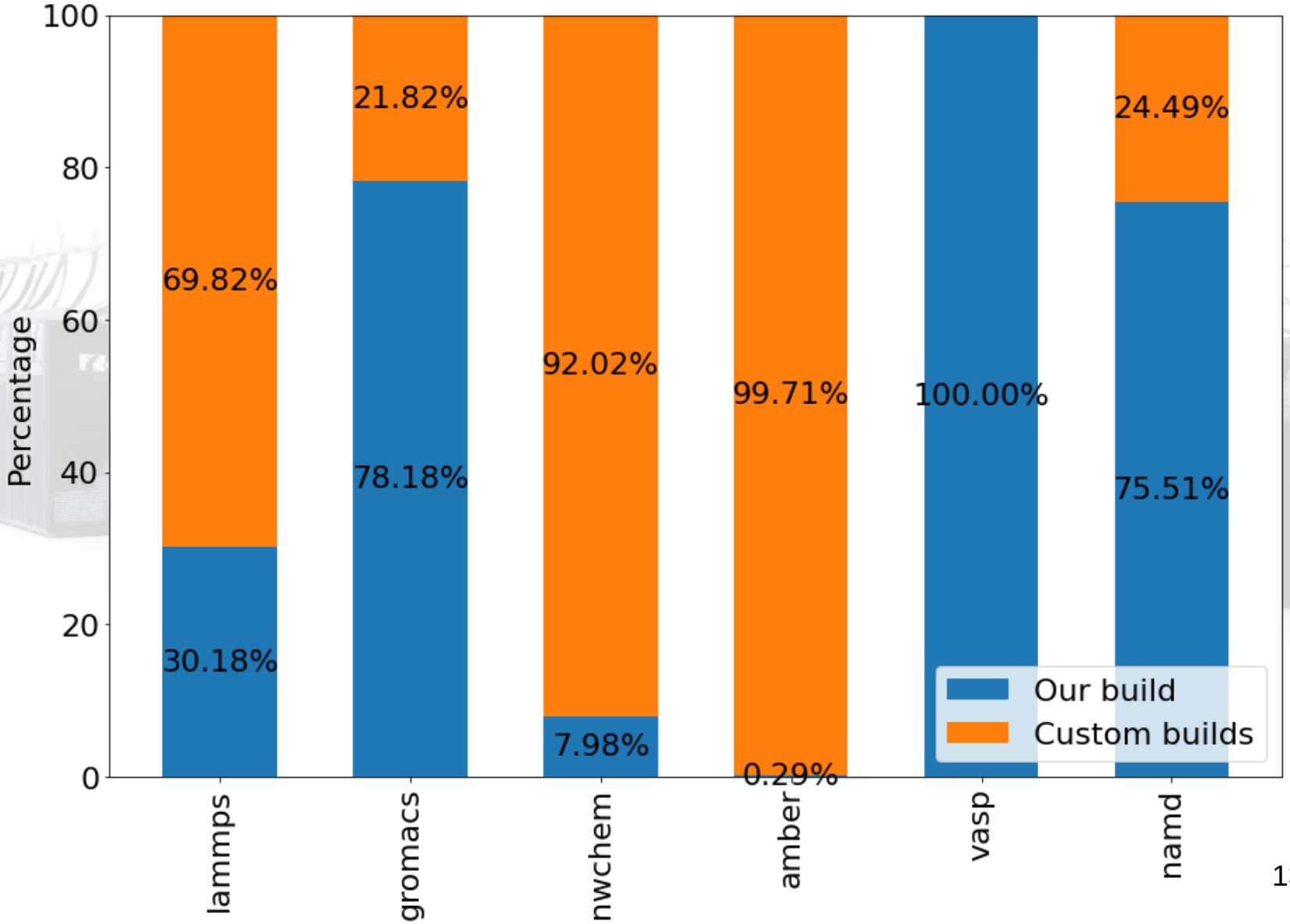
Evaluation of package version detection

- *We can clearly see that the method can detect version packages.*
- *A lot of users that use custom builds use custom versions that are currently are not installed on Lomonosov-2.*
- *Pros and cons of custom builds*
  - *+ custom modification of package to solve specific problem*
  - *+ manual optimization of the package*
  - *- incorrectly built or configured package can greatly impact the performance of calculations*
- *Question: can we distinguish custom versions of the package?*

	Total	*2019.4-gcc-cuda*	*5.1.1*	*2020.3-mpi-cuda*	*2018-icc*	*2018-gcc*
Total	*5013*	*18*	*597*	*337*	*2072*	*901*
Only custom builds	*1096*	*0*	*1*	*33*	*0*	*33*

Found versions in GROMACS jobs

Ours vs custom builds



Static method for package version detection

- *Static similar application detection gives the estimate on how jobs are close to each other – we can use that information for clustering.*

- *If we cluster custom builds of packages, we can detect specific versions of packages.*

- *By analyzing executables in one cluster, we can try to determine which version was used in that cluster.*

New package version detection

- *We clustered LAMMPS jobs using static method.*
- *We detected more than 10 clusters with different versions.*
- *Some versions are easily identifiable by binary file/paths analysis.*

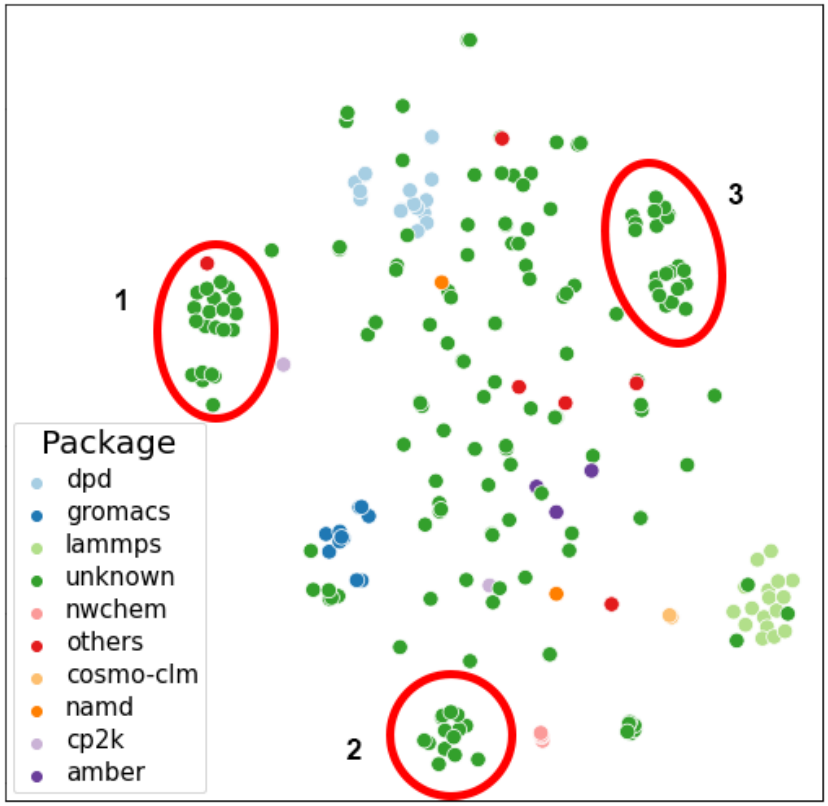
Total	*LAMMPS 2018 CUDA p100*	*CUDA (default)*	*LAMMPS 2018 CUDA*	***LAMMPS 29oct20***	***Unknown v1***	***Unknown v2***
4743	*861*	*257*	*310*	***660***	***445***	***355***

Found versions in LAMMPS jobs

New package detection

- *We can also use static method to detect new and unknown packages.*
- *Jobs are labeled by XALT.*

- *By analyzing clusters in which no packages were detected, we can get new and unknown to the system packages, e.g. XAMG.*
- *Its not always clear, what packages are used in some cases (cluster 3).*



Distribution of unique executables built using the t-SNE. Highlighted clusters are: 1 — Cabaret, 2 — XAMG, 3 — unknown

Conclusions and future work

- *Developed methods of similar application detection show very good results in solving the problem of package detection.*
- *Static method:*
  - *improves existing package detection system based on XALT by almost 80%, which gives better understanding on how supercomputer is used by users;*
  - *helps to detect new package versions that are not preinstalled on Lomonosov-2 – we found new popular versions of LAMMPS and Gromacs.*

In future we want to approach other problems like behavior or execution time prediction which also can be solved by similar application detection methods.

Thank you!

Denis Shaikhislamov
Research Computing Center of Lomonosov Moscow State University,
sdenis1995@gmail.com

27 September, 2021
Moscow