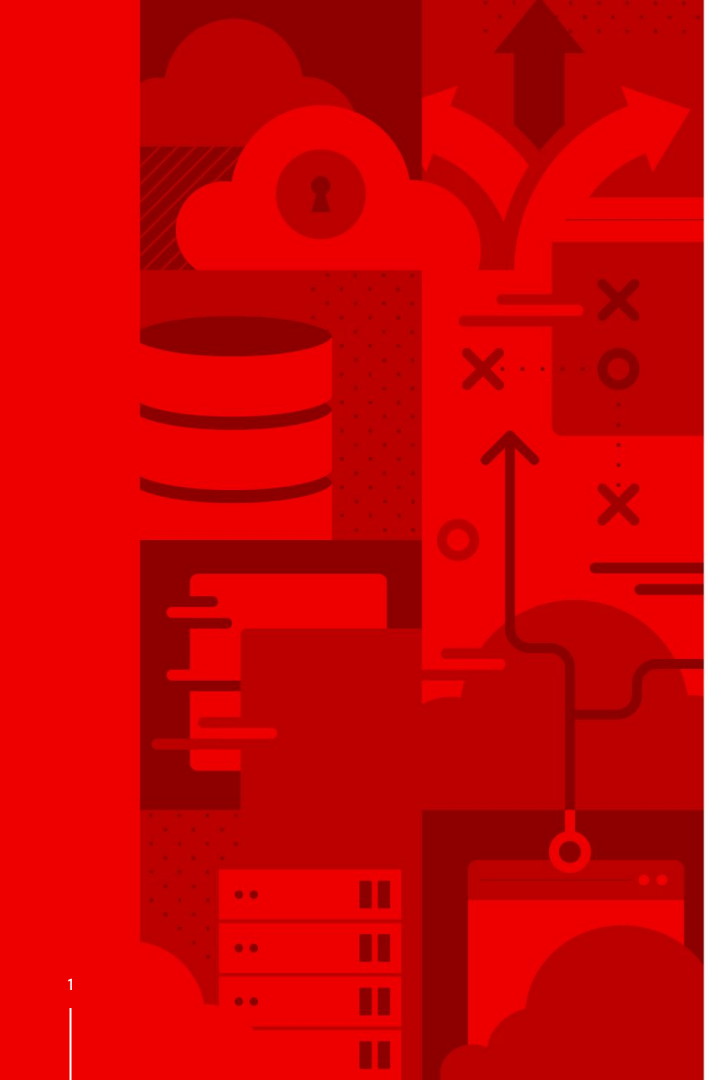


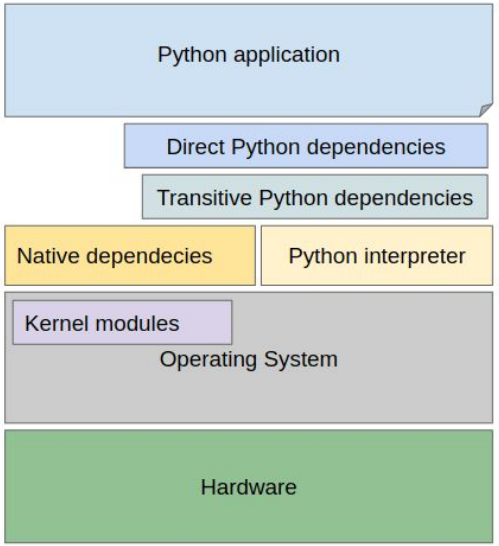
NeurIPS - Thoth

Conference on Neural Information Processing Systems

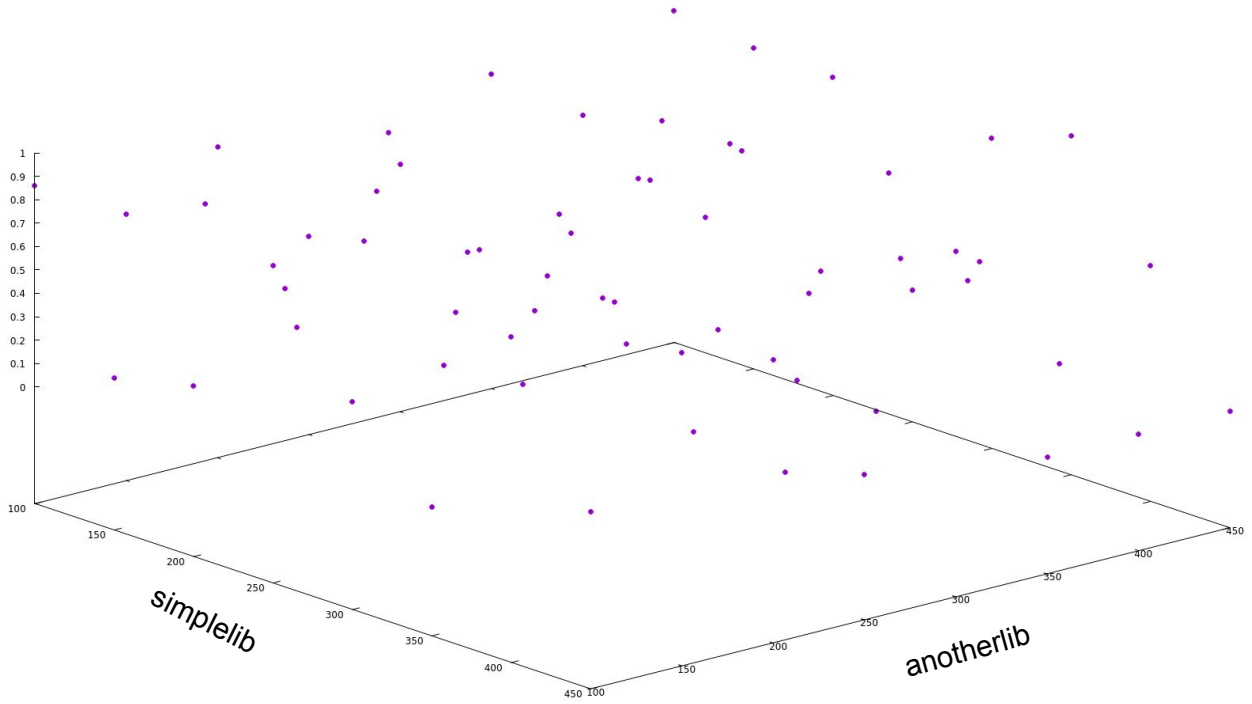
Fridolin Pokorny <fridolin@redhat.com> & Thoth team
https://thoth-station.ninja/
Red Hat AICoE, Project Thoth
2020

Thoth

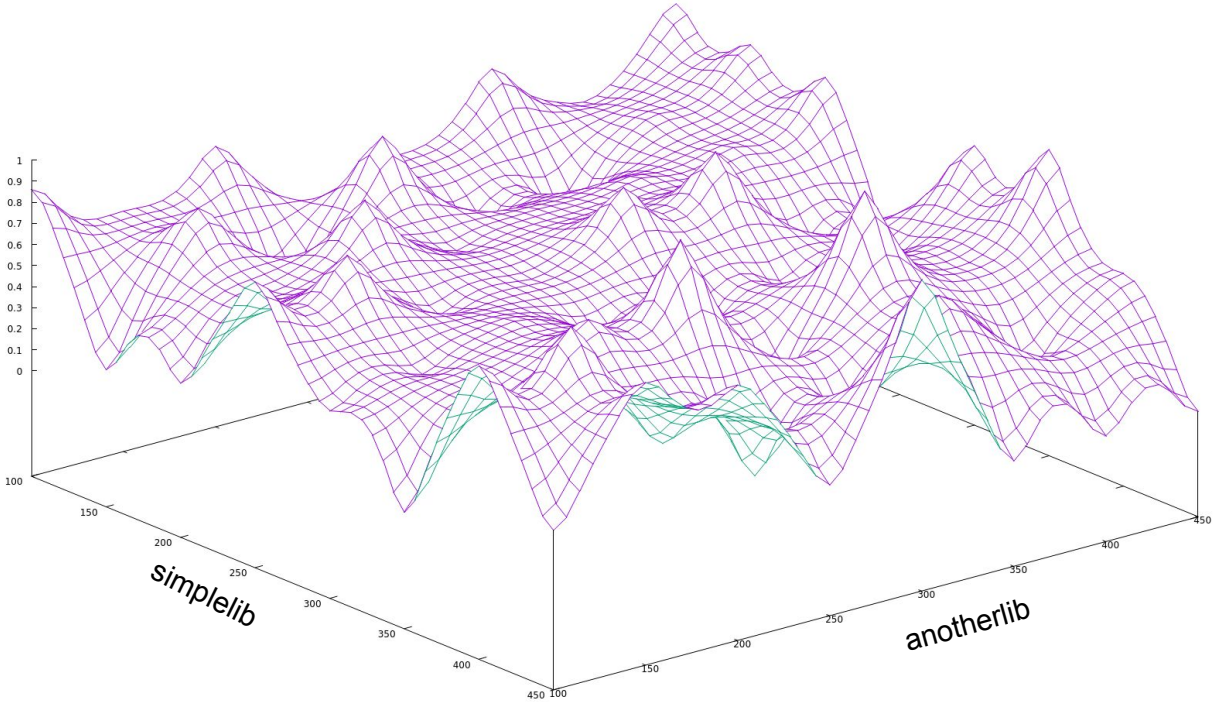
- Why Thoth?
  - Large stack of dependencies to run an application
  - Each layer in the stack introduces complexity
  - Any component on any lazyer may cause issues
  - An issue on any layer creates an observable aspect
    - Performance
    - Security
    - ...
  - How to find the right stack for user needs?



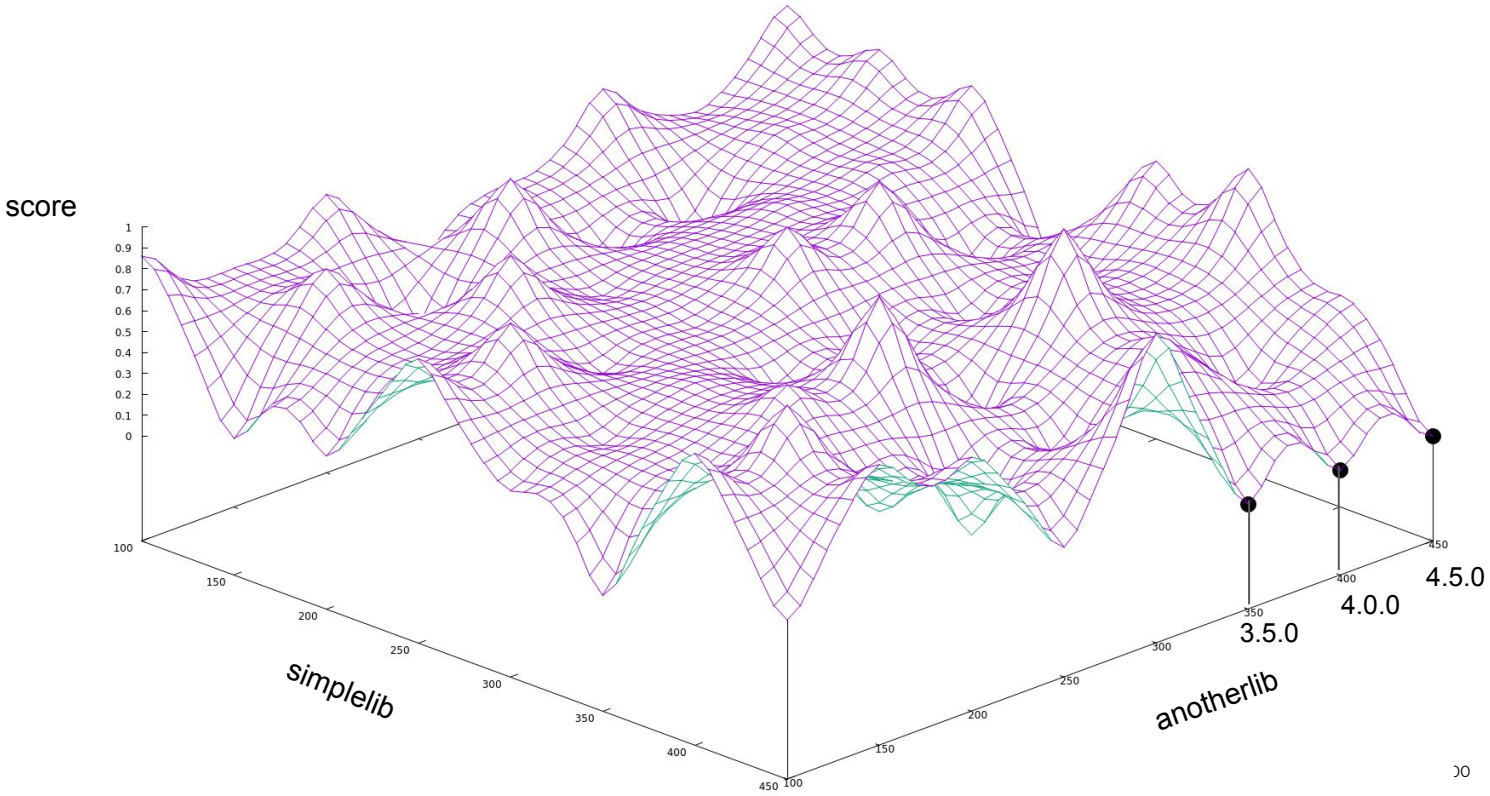
Thoth: score of the stack



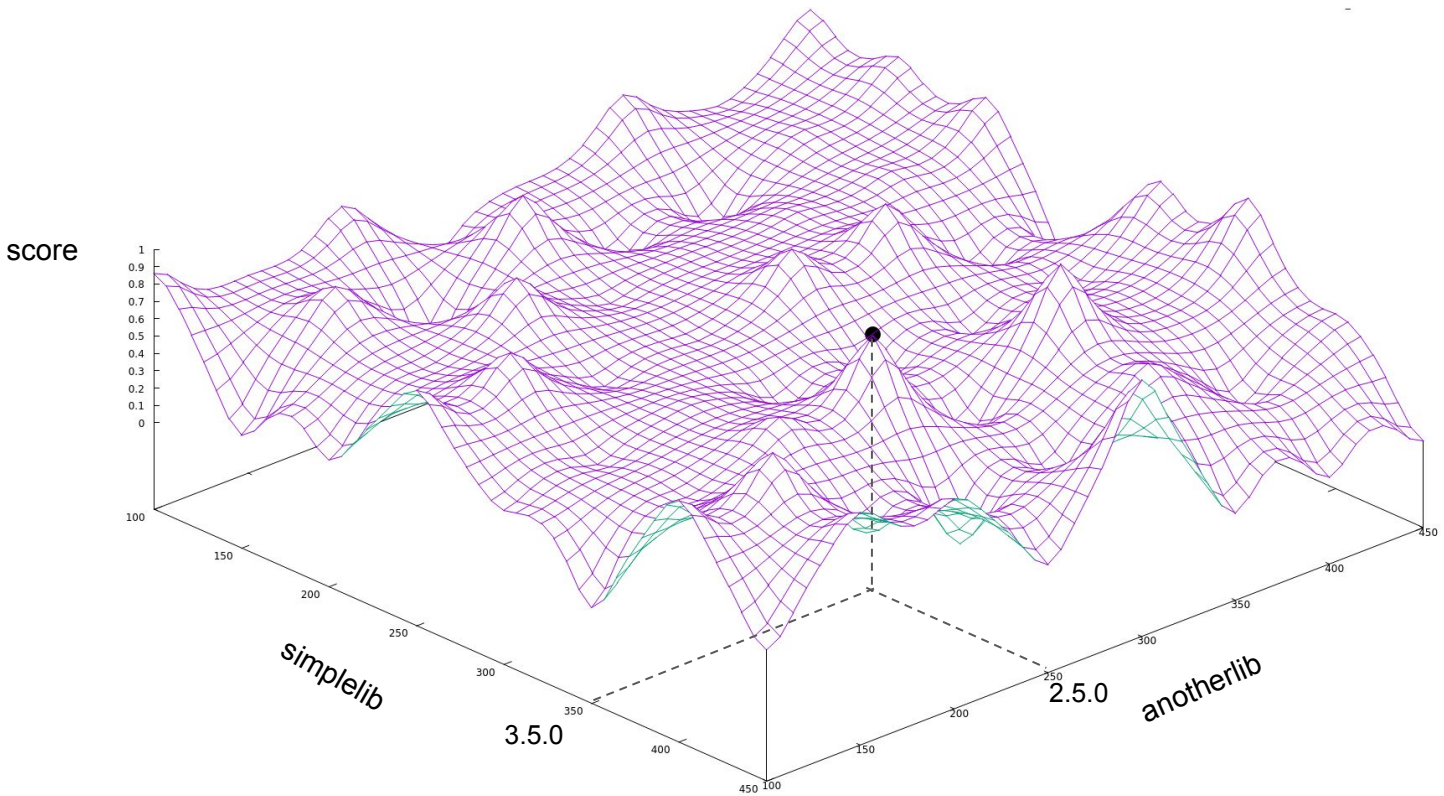
Thoth: score of the stack



Thoth: score of the stack



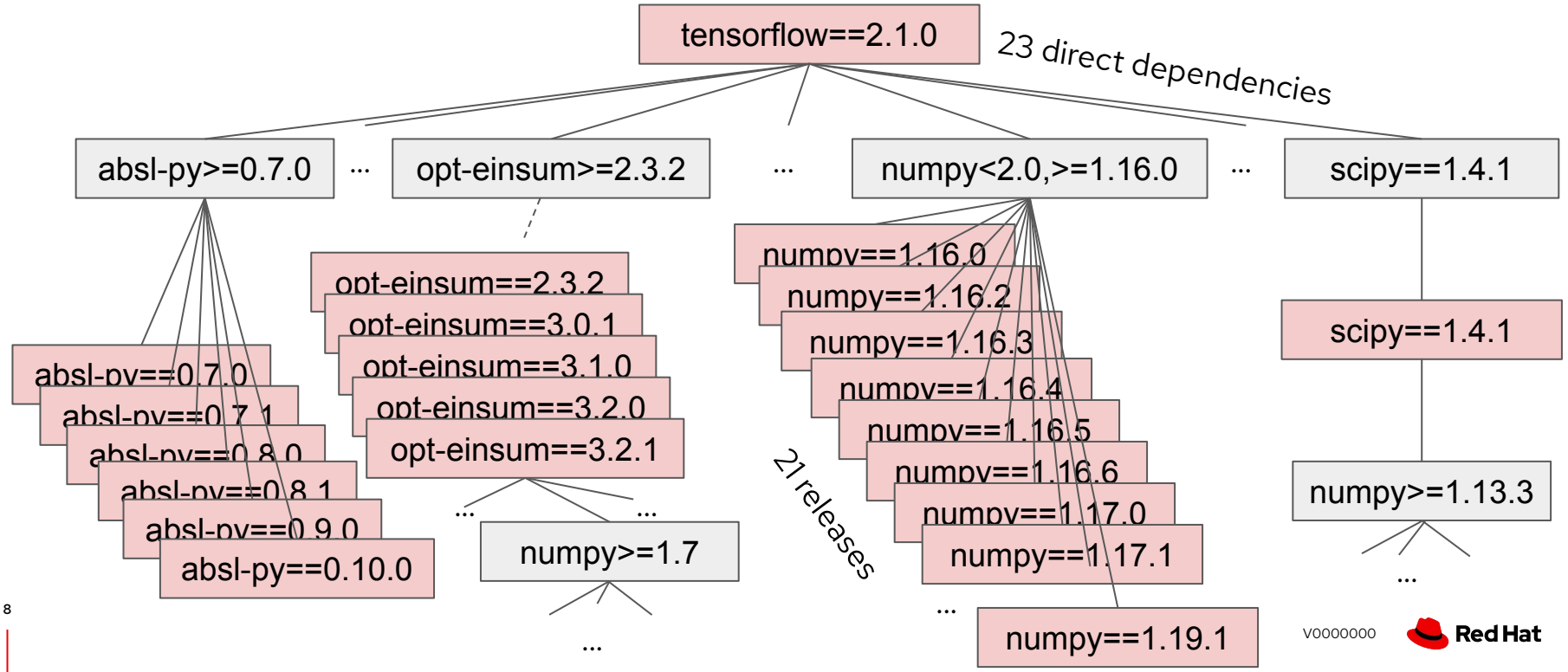
Thoth: score of the stack



Thoth

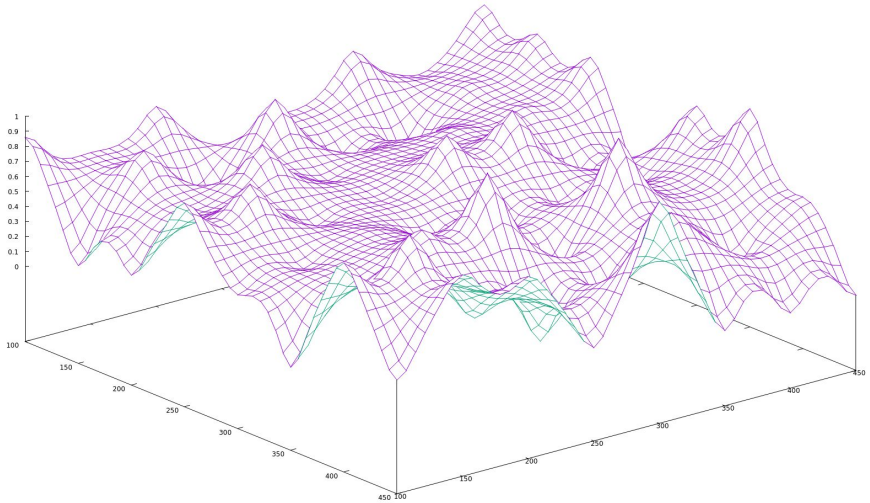
- Not just 2 dependencies but the whole dependency graph
  - Consider direct dependencies
  - Consider transitive dependencies
  - Consider runtime environment and other aspects of the software
- The state space is often too large to explore in real time
  - Number of combinations for a TensorFlow==2.1.0 stack: ~$3x10^{13}$
  - Red Hat Developer: AI software stack inspection with Thoth and TensorFlow
- Can we split this into smaller sub-problems and find a solution to the optimization problem?

Thoth: TensorFlow dependency graph



Thoth: random state space sampling

- Randomly resolve software stacks based on the dependency graph
- Check “how good” they are based on knowledge about the software



Thoth: reinforcement learning

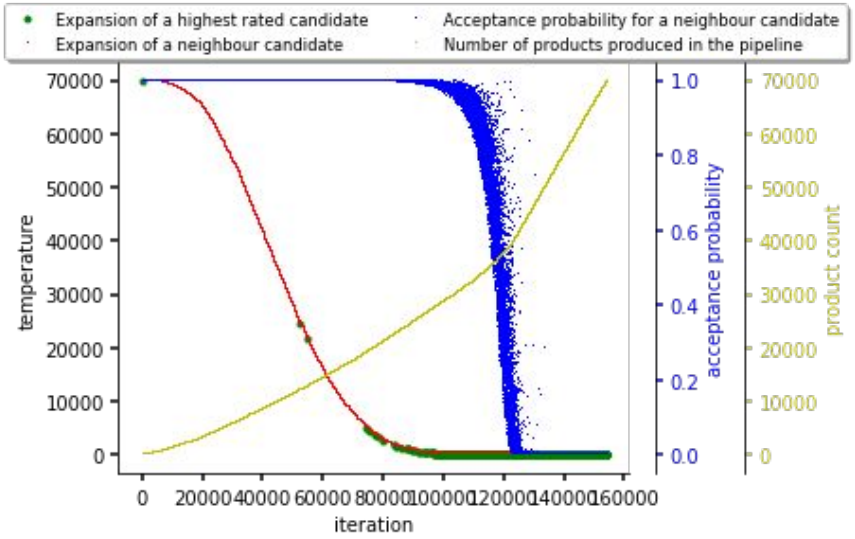
- Create a mechanism that can learn how to resolve software stacks
  - Using knowledge about the software to score software stacks
  - Learning how to navigate through the dependency graph to obtain best possible set of packages
- Gradient-free methods
  - Temporal Difference learning (TD-learning) and Monte Carlo Tree Search (MCTS)
  - Learning how to navigate through the state space and predicting score of trajectories in the dependency graph
  - Using “scoring pipeline” to score actions taken in the dependency graph and propagating a reward signal
  - Adaptive simulated annealing helps to balance exploration and exploitation

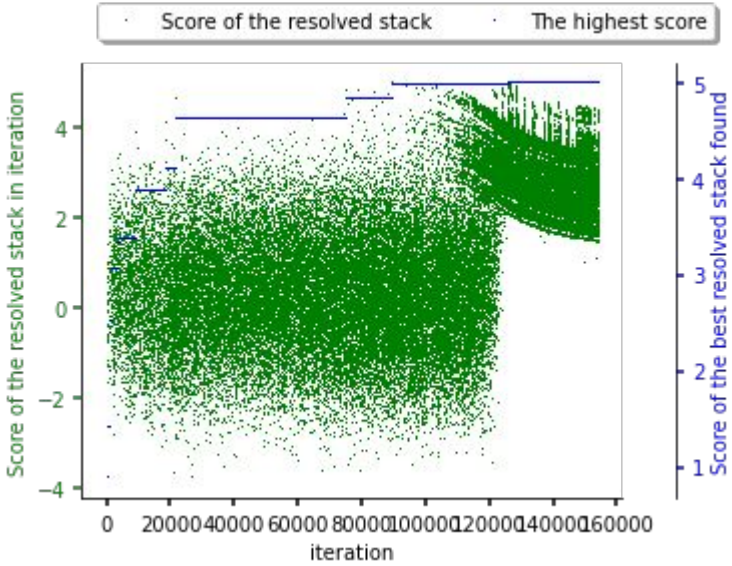
Thoth: example scenario

- Recommending a flask application
  - 7 direct packages considered in total
  - 7,861,340 possible valid software stack resolutions
  - Reinforcement learning based resolver run to resolve and score 70,000 software stacks (~1%)
  - Using randomly assigned scores for packages

Thoth: an example scenario

- Finding a software stack with a score of 4.99, the best possible candidate has 5.04



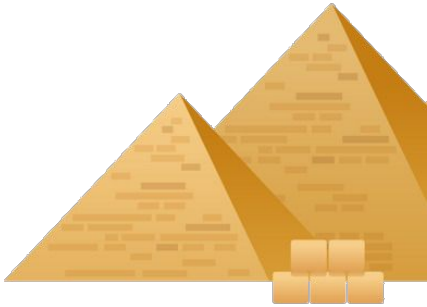


Call to action

- Use our data
  - Kaggle datasets at https://www.kaggle.com/thothstation/datasets
  - By conducting experiments on our datasets
  - By asking our services for advise
- Use our Services
  - GitHub App - https://github.com/apps/qeb-hwt
  - Thamos
  - OpenShift Pipeline Tasks
- Create Open Source software
  - Contribute a tiny feature (or a large)
  - Project Planning is done openly

Project Thoth

- AICoE, Office of the CTO
- Homepage
  - http://thoth-station.ninja/
- GitHub organization
  - http://github.com/thoth-station/
- Twitter account with updates
  - https://twitter.com/thothstation
- YouTube channel
  - https://www.youtube.com/channel/UCIUIDuq_hQ6vlzmqM59B2Lw
  - http://bit.ly/thoth-on-youtube



Thanks for your attention!

https://github.com/thoth-station

https://twitter.com/thothstation

https://www.youtube.com/channel/UCIUIDuq_hQ6vlzmqM59B2Lw

References

Website https://thoth-station.ninja/

Twitter https://twitter.com/thothstation

GitHub https://github.com/thoth-station

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat