SAND2019-13180C

Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

LDRD
Laboratory Directed Research and Development

iDASH 2019 – Track 1, Methods

Corey M. Hudson & Nick Pattengale
Sandia National Laboratories

October 28, 2019

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Implementation Summary

- Very straightforward approach
  - Use Solidity `mapping` type to lazily assign unique integers to gene and drug names
  - For each measurement (gene, variant, drug), combine the integers via concatenation into a unique index, e.g. if CYP3A5 has index 25, and pegloticase has index 122, then (CYP3A5, 52, pegloticase) is 25 || 52 || 122 = 0x19 || 0x34 || 0x7a = 0x19347a = 1651834
  - Then use that unique index to increment outcome/relation/sideeffect count `mappings`

Implementation Summary

- Various internet wisdom was useful in Solidity peculiarities, such as
  - `atoi` (i.e. string to int), and floating point division
  - string comparison (achieve by comparing hashes), e.g.
    `keccak256(abi.encodePacked("improved")) == keccak256(abi.encodePacked(outcome));`
- `entryExists` and `query` are a naïve triple for loop, where low/high are index range if "*" and otherwise set to the specific query value

```
for (x = geneLow; x <= geneHigh; x++){
 for (y = variantLow; y <= variantHigh; y++){
  for (z = drugLow; z <= drugHigh; z++){
```

Observed Performance and Future Work

- Insertion of 40,000 sample data (10,000 entries per node)
  - ≈46Mb per node of disk space used (whereas entirety of data 2Mb)
  - ≈14min required to issue all 40K insert transactions
  - 32 min elapsed before all txns verified when ethereum deployed with --txpool.accountslots 4096 --txpool.accountqueue 1024, severely bogs down otherwise
- Tests built to check query correctness, but not performance
- Ideas not in submission
  - Z-order for keys/query
    - Instead of concatenating key elements, interleave bits
    - Can outperform standard indexed search by an order of magnitude
  - If gene and drug lists can be assumed fixed, then there may be runtime savings by precomputing index lookup tables
    - however, ≈5. 5Kb of labels (314 genes, 284 drugs), could run up against ethereum contract size limit (24Kb)