SOFTWARE TOOL ARTICLE

Biobtree: A tool to search, map and visualize bioinformatics identifiers and special keywords

Tamer Gur

European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, CB10 1SD Cambridge, UK; E-mail: tgur@ebi.ac.uk

Abstract

Due to their nature, bioinformatics datasets are often closely related to each other. For this reason, search, mapping and visualization of these relations are often performed manual or programmatically via identifiers or special keywords such as gene symbols. Although various tools exist for these situations, the growing volume of bioinformatics datasets, emerging new software tools and approaches motivates new solutions. To provide a new tool for these current cases, I present the Biobtree bioinformatics tool. Biobtree effectively fetches and indexes identifiers and special keywords with their related identifiers from supported datasets, optionally with user pre-defined datasets and provides a web interface, web services and direct B+ tree data structure–based single uniform database output. Biobtree can handle billions of identifiers and runs via a single executable file with no installation and dependency required. It also aims to provide a relatively small codebase for easy maintenance, addition of new features and extension to larger datasets. Biobtree is available to download at https://www.github.com/tamerh/biobtree.

Keywords: bioinformatics, identifiers, mapping

Introduction

Bioinformatics datasets often consist of entries, where each entry is represented by unique identifier. Depending on the dataset, each entry contains various types of information such as biological function, chemical structure or literature reference etc. In addition, entries often contain cross-referencing information to other dataset entries via identifiers. Let's take as an example entry the proto-oncogene vav protein in humans, which is encoded by the *VAV1* gene. If we display this protein on the UniProt website, we see cross references to many other databases. These cross references represent relations of databases with each other. Various tools exist to deal with such data; however, the growing volume of bioinformatics datasets, emerging new software tools and analysis approaches motivates new solutions. Biobtree presented herein is capable of improved and rapid processing of large numbers of unique identifiers of entries and related identifiers that are specified via cross-reference data.

In some datasets, in addition to unique identifiers there is information that is strongly related to entries but not necessarily unique for each entry. Species names or UniProt secondary accessions are example of this type. Information that is strongly related to the entries but not necessarily unique is a second data source for Biobtree. In Biobtree, these types are called special keywords and each of these can be related to multiple entries among the datasets.

Biobtree retrieves all these identifiers, related identifiers and special keywords from various bioinformatics resources and stores it in a single database. The data resources currently used are ChEBI [1], HGNC [2], HMDB [3], InterPro [4], Europe PMC [5] and UniProt [6]. Figure 1 shows details of these datasets.

[Figure 1 about here.]

Based on stored data, Biobtree provides search, map and visualization functionalities via provided web services or a web interface. For instance, all the UniProt proteins entries belonging to a gene name, or, all Ensembl [7] genome transcripts identifiers and ENA [8] sequence identifiers that map to a protein identifier can be accessed. These relations, determined via identifiers, are stored bidirectionally so all actions can also be done in the opposite way.

Biobtree is managed from a single executable file for each major operating system without requiring any installation or

compilation. It consists of three main phases, which will be explained in later sections. These phases are named *update*, *generate* and *web* and are controlled by a Biobtree command line interface (CLI). As a database, Biobtree uses a B+ tree data structure–based LMDB key value store. LMDB provides fast batch inserts and reads and allows effective operation on a large number of records. LMDB is embedded into Biobtree's executable binary code so it does not require a separate installation.

Update Phase

The purpose of the update phase is to retrieve dataset identifiers and special keywords from remote servers or a local disk and produce files that contains identifiers and special keywords with their referred identifiers as keys and values in a sorted order. It is essential that the produced files are sorted to make fast batch inserts to LMDB database in the next phase. The updating phase is started via the Biobtree CLI with the update command. For example, the following command starts the update phase for the hgnc dataset.

```
biobtree --d hgnc update
```

Updating reads selected datasets as a stream and saves Biobtree-related data in a series of files. An advantage of reading dataset as streams is that it does not require fully downloading the dataset to the local disk. Datasets can have different formats like XML, JSON, TSV or CSV. Biobtree has specialized parsers for each dataset and parses them to produce its output files. When Biobtree runs the first time it retrieves its configuration, license and web interface files from the source code repository. Configuration files contain Biobtree runtime settings and dataset definitions.

Integrate user dataset

User data can be integrated to Biobtree. This feature creates an alternative for data providers to serve their data. Data should be gzipped and in an xml format compliant with UniProt xml schema definition. After the file path of the data is configured in a Biobtree configuration file, updating starts similarly:

```
biobtree --d my_data update
```

Updating in multiple computers

Biobtree supports executing the update phase over multiple computers. This is useful when it is necessary to use multiple computer processors at the same time such as with large datasets. The following two commands can be run on different computers with additional idx parameter to guarantee that the produced files have unique names.

```
biobtree --d uniparc –idx 1 update
biobtree --d uniref50 –idx 2 update
```

Although Biobtree supports the updating phase occurring over multiple computers, for the next phase all the produced files have to be in a single location.

Generate Phase

The purpose of this phase is to merge all the files produced in the update phase by keeping the sorted order and generate the final key and values in the generated LMDB database. Keys consist of identifiers and special keywords and values are identifiers that are referred to by these keys with their dataset information. If the values size for each key are above a certain threshold they are saved in pages. The following command starts the generate phase.

```
biobtree generate
```

The generated database output is used in next web phase but it can be also used directly. Example source codes for using the database directly can be found in the project github page.

Web Phase

The purpose of this phase is to provide web services and a web interface via the produced output of the generate phase. The following command starts web phase

```
biobtree web
```

With this command the Biobtree web server is started instantly, serving the REST, gRPC and web interface services.

REST service

To make queries in the produced database, Biobtree provides RESTful endpoints with json-formatted responses. For example, each dataset has a unique identifier and other meta information like name, url template, etc. Dataset-unique identifiers are used in all the services to distinguish the dataset. These meta information is retrieved via the following endpoint:

http://localhost:8888/ws/meta

The following are used to query single or multiple identifiers or special keywords:

http://localhost:8888/ws/?idlist=vav_human
http://localhost:8888/ws/?idlist=vav_human,tpi1,brca2

To make a paging query for a certain identifier:

http://localhost:8888/ws/?id=vav_human&dataset=1&page=1

To make a filtering query based on a dataset:

http://localhost:8888/ws/?id=vav_human&dataset=1&filters=102

To make a paging query with active filtering:

http://localhost:8888/ws/?id=vav_human&dataset=1&filters=102&page=1

gRPC service

RESTful services are often json-based and are very convenient in json-based applications. But for non–json-based applications, it requires an extra process of serialization and deserialization. To address this, Biobtree provides a gRPC service with same functionality as its RESTful service. Sample codes for using gRPC in different languages can be found on the project github page. The following is snippet of Biobtree gRPC service definitions:

```
service BiobtreeService {

  rpc Get      (BiobtreeGetRequest)      returns (BiobtreeGetResponse);
  rpc GetPage (BiobtreeGetPageRequest) returns (BiobtreeGetPageResponse);
  rpc Filter   (BiobtreeFilterRequest)   returns (BiobtreeFilterResponse);
  rpc Meta     (BiobtreeMetaRequest)     returns (BiobtreeMetaResponse);

}
```

Web interface

The Web interface allow user to visualize the produced database via the RESTful service. Once the web phase is started it is accessed via the browser from the following address:

http://localhost:8888/ui

The Web interface provides searching of multiple identifiers and special keywords, visualizing and filtering results, and executing bulk queries. On the result page for each result, it provides a url to access the main website where the data are originally produced. Figures 2 and 3 show the main and result page of the web interface.

[Figure 2 about here.]

[Figure 3 about here.]

Benchmarks

Software benchmarks should often be taken with a grain of salt, especially where the input data is large, because benchmarks results can be affected by many factors like the operating system, storage, network speed, input data, application parameters etc. Considering these, the purpose of Biobtree benchmarks is mainly to show overall capabilities and resource usage of Biobtree. Figure 4 shows the benchmark details and results. Benchmarks were primarily computed at digitalocean London datacentres using their cpu optimized droplets with block storage volumes. Hundred thousand sample query which used in the benchmarks can be found project github page.

[Figure 4 about here.]

Discussion

The benchmarks show that Biobtree has produced LMDB database output in relatively acceptable times. Let's discuss how Biobtree behaves if UniProt provides tens of times larger data. Clearly, more disk space would have been needed.

If enough disk space is provided, the next obstacle would have happened during the update phase, because currently UniProt provides a single gzip compressed file for each dataset and Biobtree reads each file as a stream from the beginning to end. Gzip does not allow a random access to a file unless there are checkpoints defined. This characteristic of gzip prevents the processing single large file in a split manner and utilizes more computing resources if available. Two solutions can address the issue. The first could be if UniProt would allow its datasets to be capable of parallel processing, like splitting

and compressing files inside a tar archive. The second solution would be to implement a new functionality in Biobtree and save and decompress these files to local disk and make parallel access to decompressed files.

A further obstacle would have happened during the generate phase, since we would have obtained more files from update phase. The generate phase struggles to merge all these files and could cause much longer times of output generation. To address this obstacle, a new phase could be implemented that runs before the generate phase and merges files coming from the update phase.

Limitations

Although duplicate values for each key are discarded during the update phase for each dataset, the generate phase could rarely produce duplicate values. The user needs to discard these duplicate records manually if these are created. Another limitation is when querying the special keywords, they need to be fully specified including all space characters.

Future Work

The limitations can be addressed and new functionalities added in the future. For instance, different bioinformatics datasets like Ensembl [7] or ENA [8] can be integrated. Another feature would be sorting result values based on a certain criterion.

Competing Interests

No competing interests were disclosed.

Data (and Software) Availability

All source codes and binaries are availabe at https://www.github.com/tamerh/biobtree

Grant Information

The author(s) declared that no grants were involved in supporting this work.

References

[1] Janna Hastings, Gareth Owen, Adriano Dekker, Marcus Ennis, Namrata Kale, Venkatesh Muthukrishnan, Steve Turner, Neil Swainston, Pedro Mendes, Christoph Steinbeck; ChEBI in 2016: Improved services and an expanding collection of metabolites, Nucleic Acids Research, Volume 44, Issue D1, 4 January 2016, Pages D1214–D1219, https://doi.org/10.1093/nar/gkv1031

[2] Yates B, Braschi B, Gray K, Seal R, Tweedie S, Bruford E. Genenames.org: the HGNC and VGNC resources in 2017. Nucleic Acids Res. 2017 Jan 4; 45(D1):D619-625. PMID:27799471

[3] Wishart DS, Feunang YD, Marcu A, Guo AC, Liang K, et al., HMDB 4.0 — The Human Metabolome Database for 2018. Nucleic Acids Res. 2018. Jan 4;46(D1):D608-17. 29140435

[4] Alex L Mitchell, Teresa K Attwood, Patricia C Babbitt, Matthias Blum, Peer Bork, Alan Bridge, Shoshana D Brown, Hsin-Yu Chang, Sara El-Gebali, Matthew I Fraser, Julian Gough, David R Haft, Hongzhan Huang, Ivica Letunic, Rodrigo Lopez, Aurélien Luciani, Fabio Madeira, Aron Marchler-Bauer, Huaiyu Mi, Darren A Natale, Marco Necci, Gift Nuka, Christine Orengo, Arun P Pandurangan, Typhaine Paysan-Lafosse, Sebastien Pesseat, Simon C Potter, Matloob A Qureshi, Neil D Rawlings, Nicole Redaschi, Lorna J Richardson, Catherine Rivoire, Gustavo A Salazar, Amaia Sangrador-Vegas, Christian J A Sigrist, Ian Sillitoe, Granger G Sutton, Narmada Thanki, Paul D Thomas, Silvio C E Tosatto, Siew-Yit Yong, Robert D Finn; InterPro in 2019: improving coverage, classification and access to protein

sequence annotations, Nucleic Acids Research, Volume 47, Issue D1, 8 January 2019, Pages D351–D360, https://doi.org/10.1093/nar/gky1100

[5] For more information on Europe PMC, read the article: Europe PMC: a full-text literature database for the life sciences and platform for. *Nucleic Acids Res*, 2015:43. Database issue): D1042-D1048.

[6] The UniProt Consortium; UniProt: a worldwide hub of protein knowledge, Nucleic Acids Research, Volume 47, Issue D1, 8 January 2019, Pages D506–D515, https://doi.org/10.1093/nar/gky1049

[7] Daniel R Zerbino, Premanand Achuthan, Wasiu Akanni, M Ridwan Amode, Daniel Barrell, Jyothish Bhai, Konstantinos Billis, Carla Cummins, Astrid Gall, Carlos García Girón, Laurent Gil, Leo Gordon, Leanne Haggerty, Erin Haskell, Thibaut Hourlier, Osagie G Izuogu, Sophie H Janacek, Thomas Juettemann, Jimmy Kiang To, Matthew R Laird, Ilias Lavidas, Zhicheng Liu, Jane E Loveland, Thomas Maurel, William McLaren, Benjamin Moore, Jonathan Mudge, Daniel N Murphy, Victoria Newman, Michael Nuhn, Denye Ogeh, Chuang Kee Ong, Anne Parker, Mateus Patricio, Harpreet Singh Riat, Helen Schuilenburg, Dan Sheppard, Helen Sparrow, Kieron Taylor, Anja Thormann, Alessandro Vullo, Brandon Walts, Amonida Zadissa, Adam Frankish, Sarah E Hunt, Myrto Kostadima, Nicholas Langridge, Fergal J Martin, Matthieu Muffato, Emily Perry, Magali Ruffier, Dan M Staines, Stephen J Trevanion, Bronwen L Aken, Fiona Cunningham, Andrew Yates, Paul Flicek; Ensembl 2018, Nucleic Acids Research, Volume 46, Issue D1, 4 January 2018, Pages D754–D761, https://doi.org/10.1093/nar/gkx1098

[8] Peter W Harrison, Blaise Alako, Clara Amid, Ana Cerdeño-Tárraga, Iain Cleland, Sam Holt, Abdulrahman Hussein, Suran Jayathilaka, Simon Kay, Thomas Keane, Rasko Leinonen, Xin Liu, Josué Martínez-Villacorta, Annalisa Milano, Nima Pakseresht, Jeena Rajan, Kethi Reddy, Edward Richards, Marc Rosello, Nicole Silvester, Dmitriy Smirnov, Ana-Luisa Toribio, Senthilnathan Vijayaraja, Guy Cochrane; The European Nucleotide Archive in 2018, Nucleic Acids Research, Volume 47, Issue D1, 8 January 2019, Pages D84–D88, https://doi.org/10.1093/nar/gky1078

List of Figures

1 List of datasets 7
2 Beb interface main page 8
3 Web interface result page 9
4 Benchmarks results 10

Dataset	Description	File Name	Location	Format	Special Keywords
ChEBI	ChEBI reference accession data	database_accession.tsv	ftp.ebi.ac.uk/chebi/Flat_file_tab_delimited/	TSV	-
HGNC	Human gene nomenclature	hgnc_complete_set.json	ftp.ebi.ac.uk/genenames/new/json/	JSON	name,symbol
HMDB	Human metabolome database	hmdb_metabolites.zip	http://www.hmdb.ca/system/downloads/current/	XML	name, synonyms
InterPro	Protein Families	interpro.xml.gz	ftp://ftp.ebi.ac.uk/pub/databases/interpro/current	XML	-
Literature mappings	Literature pmid, pmcid and doi mappings	PMID_PMCID_DOI.csv.gz	ftp://ftp.ebi.ac.uk/pub/databases/pmc/DOI/	CSV	-
Taxonomy	NCBI Taxonomy	taxonomy.xml.gz	ftp://ftp.ebi.ac.uk/pub/databases/taxonomy/	XML	scientificName
Uniparc	UniProt Sequence Archive	uniparc_all.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/uniparc/	XML	-
UniProt reviewed	UniProt Knowledgebase reviewed	uniprot_sprot.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/knowledgebase/complete/	XML	accession
UniProt unreviewed	UniProt Knowledgebase unreviewed	uniprot_trembl.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/knowledgebase/complete/	XML	accession
Uniref50	UniProt sequence clusters	uniref50.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/uniref/uniref50/	XML	-
Uniref90	UniProt sequence clusters	uniref90.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/uniref/uniref90/	XML	-
Uniref100	UniProt sequence clusters	uniref100.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/uniref/uniref100/	XML	-

Figure 1. List of datasets

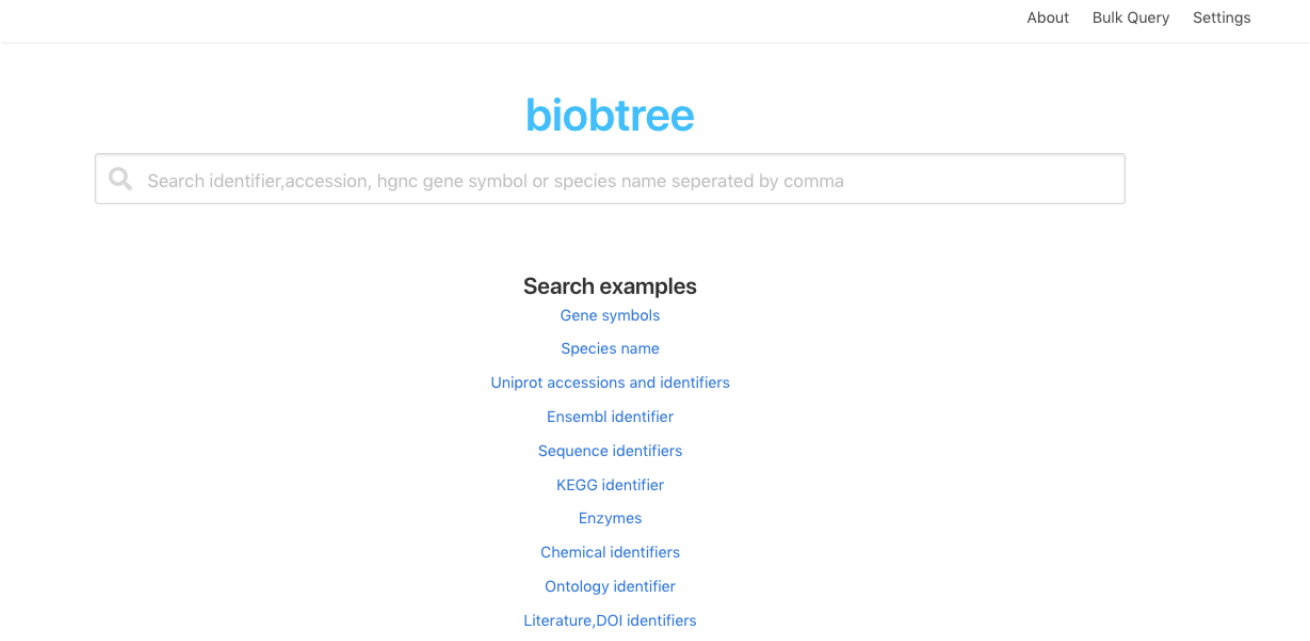

Figure 2. Web interface main page

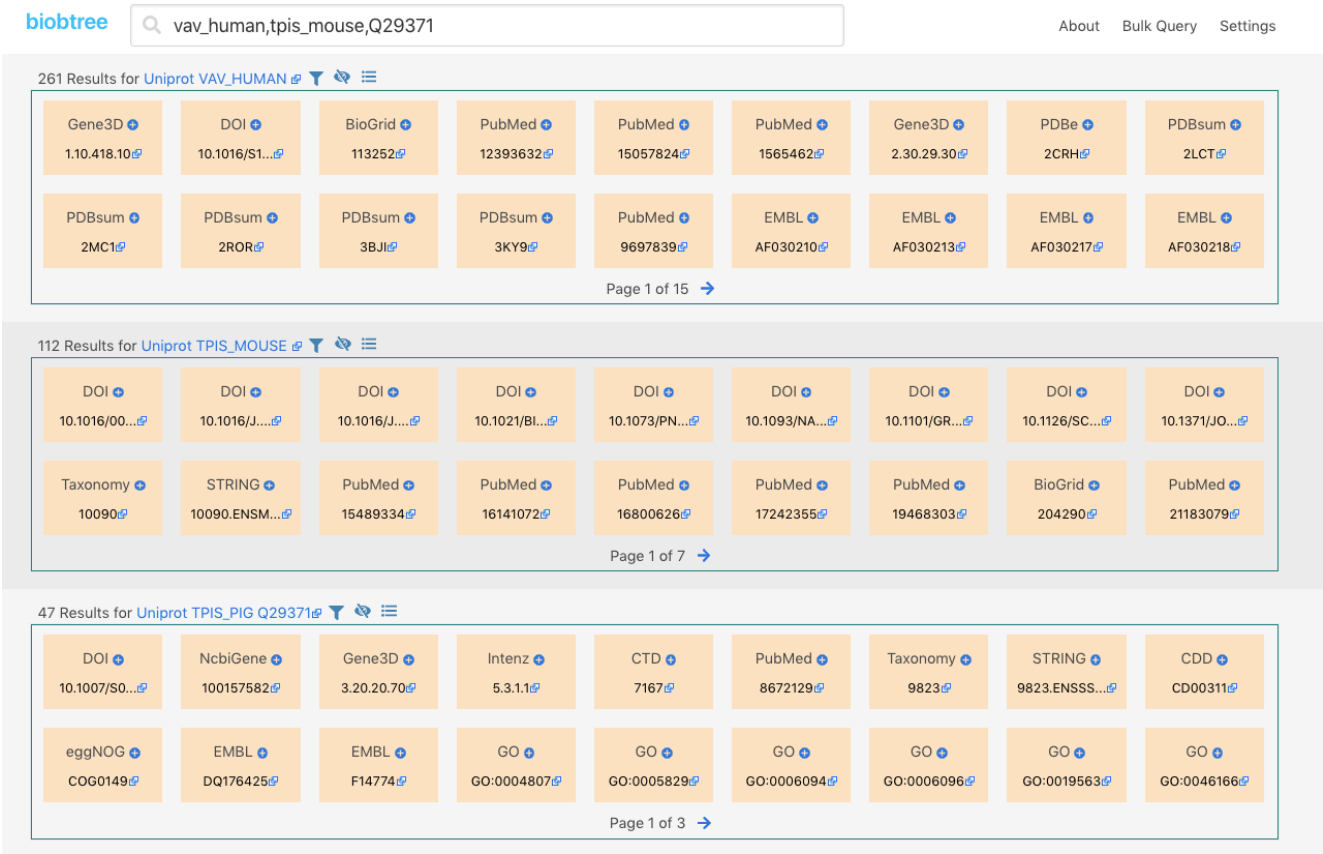

Figure 3. Web interface result page

	Benchmark-1	Benchmark-2	Benchmark-3	Benchmark-4	Benchmark-5	Benchmark-6	Benchmark-7	Benchmark-8	Benchmark-9
Description	Runs all the phases and 100K query for default datasets at macbook laptop	Runs default dataset update phase at digital ocean.	Runs uniref50 update phase at digital ocean.	Runs uniref90 update phase at digital ocean.	Runs uniref100 update phase at digital ocean.	Runs uniprot unreviewed update phase at digital ocean.	Runs uniparc update phase at digital ocean.	Runs generate phase for all data at digital ocean.	Runs 100K query against all data at digital ocean.
Operating System	macOS High Sierra	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64
CPU	4	8	8	8	8	8	8	8	8
RAM	16GB	16GB	16GB	16GB	16GB	16GB	16GB	16GB	16GB
Update Phase Avrg. CPU Usage	%309	%556	%201	%201	%204	%303	%351	-	-
Update Phase Avrg. RAM Usage	%9	%84	%39	%36	%36	%36	%37	-	-
Update Phase Elapsed Duration	13m	5m	1h24m	2h3m	2h43m	8h27m	8h37m	-	-
Update Phase Files Size	838MB	823MB	3GB	2.8GB	2.4GB	25GB	38GB	-	-
Generate Phase Avrg. Cpu Usage	%141	-	-	-	-	-	-	%164	-
Generate Phase Avrg. RAM Usage	%19	-	-	-	-	-	-	%90	-
Generate Phase Elapsed Duration	18m	-	-	-	-	-	-	8h5m	-
Generate Phase Files Size	3.8GB	-	-	-	-	-	-	392GB	-
Web Phase Avrg. Cpu Usage	%113	-	-	-	-	-	-	-	%3
Web Phase Avrg. RAM Usage	%2	-	-	-	-	-	-	-	%36
100K Consequent Query Avrg. Response Time Inside Same Machine	0.3ms	-	-	-	-	-	-	-	3ms
100K Consequent Query Avrg. Response Time From Outside Network and Machine	-	-	-	-	-	-	-	-	20ms
Total Keys	60M	-	-	-	-	-	-	3.19B	-
Total Values	131M	-	-	-	-	-	-	17.07B	-

Figure 4. Benchmarks results.

ms: Millisecond
m: Minute
h: Hour
K: Thousand
M: Million
B: Billion
MB: Megabyte
GB: Gigabyte