|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Features | GraphDB EE | RDFOX | AllegroGraph | Virtuoso |
| 1 | Manage unlimited number of RDF statements | Link:  <https://www.ontotext.com/products/graphdb/>,  <https://www.w3.org/wiki/LargeTripleStores#GraphDB.E2.84.A2_by_Ontotext_.2817B.29> | Link:  [https://www.w3.org/wiki/LargeTripleStores#RDFox\_.2819.5B.29](https://www.w3.org/wiki/LargeTripleStores%23RDFox_.2819.5B.29),  <https://www.w3.org/wiki/LargeTripleStores#RDFox_.2819.5B.29> | Link:  <https://allegrograph.com/products/allegrograph/>,  <https://www.w3.org/wiki/LargeTripleStores#AllegroGraph_.281.2BT.29> | Link:  <http://vos.openlinksw.com/owiki/wiki/VOS/VOSRDFFAQ> |
| 2 | Full Sparql 1.1 support | Link:  <https://graphdb.ontotext.com/documentation/standard/sparql-compliance.html> | * ( Full support update but not query )   Link:  [https://docs.oxfordsemantic.tech/querying.html#querying](https://docs.oxfordsemantic.tech/querying.html%23querying) | Link:  <https://franz.com/agraph/support/documentation/6.1.3/sparql-reference.html> | Link:  <https://virtuoso.openlinksw.com/features/#1100703a692f497c624c054916e4cb68> |
| 3 | Materialization reasoning | Link:  [https://graphdb.ontotext.com/documentation/standard/introduction-to-semantic-web.html#introduction-to-semantic-web-reasoning-strategies](https://graphdb.ontotext.com/documentation/standard/introduction-to-semantic-web.html%23introduction-to-semantic-web-reasoning-strategies) | Link:  <https://docs.oxfordsemantic.tech/3.1/reasoning-in-rdfox.html>,  [https://docs.oxfordsemantic.tech/3.1/reasoning-in-rdfox.html#materialization-based-reasoning](https://docs.oxfordsemantic.tech/3.1/reasoning-in-rdfox.html%23materialization-based-reasoning) | * (But Dynamic Materialization)   Link:  <https://franz.com/agraph/support/documentation/current/materializer.html>,  <https://franz.com/agraph/support/documentation/current/reasoner-tutorial.html>,  <https://allegrograph.com/allegrograph-rdfs-dynamic-materialization/> | Link:  <https://www.w3.org/2001/sw/wiki/OpenLink_Virtuoso>,  <https://community.openlinksw.com/t/virtuoso-openlink-reasoning-backward-or-and-forward/1887> |
| 4 | Efficient retraction of inferred statements | Link:  [https://graphdb.ontotext.com/documentation/free/reasoning.html#reasoning-retraction-of-assertions](https://graphdb.ontotext.com/documentation/free/reasoning.html%23reasoning-retraction-of-assertions) | Link:  [https://docs.oxfordsemantic.tech/3.1/reasoning-in-rdfox.html#materialization-based-reasoning](https://docs.oxfordsemantic.tech/3.1/reasoning-in-rdfox.html%23materialization-based-reasoning) | \_ | \_ |
| 5 | Full standard-compliant and optimized rulesets for RDFS, OWL2 RL and QL | Link:  [https://graphdb.ontotext.com/documentation/standard/reasoning.html#reasoning-predefined-rulesets](https://graphdb.ontotext.com/documentation/standard/reasoning.html%23reasoning-predefined-rulesets),  <https://www.w3.org/TR/owl2-profiles/#OWL_2_RL>,  <https://www.w3.org/TR/owl2-profiles/#OWL_2_QL> | Link:  <https://www.w3.org/2001/sw/wiki/RDFox>,  <https://www.researchgate.net/publication/300338060_RDFox_A_Highly-Scalable_RDF_Store>,  <https://www.w3.org/2001/sw/wiki/RDFox>  Supports RDF,RDFS, owl2 RL | Link:  <https://franz.com/agraph/support/documentation/current/reasoner-tutorial.html>,  <https://franz.com/agraph/support/documentation/current/materializer.htm>  Supports subset of RDFS and owl | Link:  <http://docs.openlinksw.com/virtuoso/rdfsparqlrulemake/>,  <https://virtuoso.openlinksw.com/features-comparison-matrix/#rdfrel> |
| 6 | Plugin API’s | Link:  <https://graphdb.ontotext.com/documentation/standard/plug-in-api.html> |  |  | Link:  <http://vos.openlinksw.com/owiki/wiki/VOS#Plugins> |
| 7 | Prolog rules and query |  |  | Link:  <https://franz.com/agraph/support/documentation/current/prolog-tutorial.html> |  |
| 8 | Geospatial Indexing | Link:  <https://graphdb.ontotext.com/documentation/standard/geo-spatial-extensions.html> |  | Link:  <https://franz.com/agraph/support/documentation/current/geospatial-nd.html> | Link:  <http://docs.openlinksw.com/virtuoso/sqlrefgeospatial/> |
| 9 | GeoSparql Support | Link:  <https://graphdb.ontotext.com/documentation/free/geosparql-support.html> |  |  | Link:  <https://github.com/openlink/virtuoso-opensource/blob/develop/7/README.GeoSPARQL.md>,  <http://vos.openlinksw.com/owiki/wiki/VOS/VirtGeoSPARQLEnhancementDocs> |
| 10 | Web based console | Link:  <https://graphdb.ontotext.com/documentation/standard/workbench-user-interface.html> | Link:  <https://medium.com/oxford-semantic-technologies/getting-started-with-rdfox-79adbbfa86b1>,  <https://docs.oxfordsemantic.tech/getting-started.html#getting-started-with-the-web-console> | Link:  <https://allegrograph.com/products/gruff/> | Link:  <https://virtuoso.openlinksw.com/features-comparison-matrix/#interui> |
| 11 | SHACL validation | Link:  <https://graphdb.ontotext.com/documentation/free/shacl-validation.html> | Link:  <https://docs.oxfordsemantic.tech/tuple-tables.html#rdfox-shacl> | Link:  <https://allegrograph.com/shacl-shapes-constraint-language-in-allegrograph/> | \_ |
| 12 | Solr connector for full text search | * (Not available in GraphDB Free and SE version )   Link:  <https://graphdb.ontotext.com/documentation/enterprise/solr-graphdb-connector.html>  NOTE: Not available in GraphDB Free and SE version | Link:  <https://docs.oxfordsemantic.tech/data-sources.html> | Link:  <https://franz.com/agraph/support/documentation/current/solr-index.html> | \_ |
| 13 | Lucene connector for full text search | * (Not available in GraphDB Free and SE version )   Link:  <https://graphdb.ontotext.com/documentation/standard/lucene-graphdb-connector.html> |  |  | \_ |
| 14 | Datalog rules for reasoning |  | Link:  <https://medium.com/oxford-semantic-technologies/datalog-basics-and-rdfox-942768327604>,  <https://www.w3.org/2001/sw/wiki/RDFox> |  |  |
| 15 | Two Phase Commit |  |  | Link:  <https://franz.com/agraph/support/documentation/current/two-phase-commit.html> | Link:  <https://virtuoso.openlinksw.com/features/#d0d66b79c6ffd0380d69bcb2ec3a75ab> |
| 16 | Query Optimizer | Link:  <https://graphdb.ontotext.com/documentation/free/data-loading-query-optimisations.html#data-loading-query-optimisations-query>,  <https://graphdb.ontotext.com/documentation/free/explain-plan.html> | Link:  <https://docs.oxfordsemantic.tech/querying.html>,  <https://docs.oxfordsemantic.tech/querying.html#monitoring-query-execution> | Link:  <https://allegrograph.com/products/allegrograph/>,  <https://allegrograph.com/benchmarks-sp2/?print=print>,  <https://franz.com/agraph/support/documentation/current/query-analysis.html> | Link:  <https://medium.com/virtuoso-blog/virtuoso-8-0-creating-a-custom-inference-rules-using-spin-vocabulary-d7a060f859ef>, |
| 17 | Elastic search connector for full text search | Link:  <https://graphdb.ontotext.com/documentation/enterprise/elasticsearch-graphdb-connector.html> |  |  | \_ |
| 18 | Spin support | Link:  <https://graphdb.ontotext.com/documentation/8.4/enterprise/loading-data-using-ontorefine.html>,  <https://www.w3.org/Submission/2011/SUBM-spin-sparql-20110222/>  See example: <http://topbraid.org/examples/spinsquare.ttl> |  | Link:  <https://franz.com/agraph/support/documentation/current/spin.html>,  <https://www.w3.org/Submission/2011/SUBM-spin-sparql-20110222/>  See example: <http://topbraid.org/examples/spinsquare.ttl> | Link:  <https://medium.com/virtuoso-blog/virtuoso-8-0-creating-a-custom-inference-rules-using-spin-vocabulary-d7a060f859ef> |
| 19 | Partitioning method | \_ | \_ | * (FedShard Technology)   Link:  <https://franz.com/agraph/support/documentation/7.0.0/agraph-introduction.html#handling-vld> | * ­ (Elastic Cluster)   Link:  <http://docs.openlinksw.com/virtuoso/clusterstcnf7/> |
| 20 | Sparql-mm-support | Link:  <https://graphdb.ontotext.com/documentation/standard/sparql-mm-support.html>,  <https://marmotta.apache.org/kiwi/sparql-mm.html> |  |  |  |
| 21 | Supports reification | Link:  <https://graphdb.ontotext.com/free/devhub/rdf-sparql-star.html> | Link:  <https://docs.oxfordsemantic.tech/3.1/reasoning-in-rdfox.html> | Link:  <https://franz.com/agraph/support/documentation/current/agraph-introduction.html> | Link:  <http://vos.openlinksw.com/owiki/wiki/VOS/VirtCartridgesReification>,  <https://www.openlinksw.com/weblog/oerling/?id=1572> |
| 22 | Sparql Endpoint | Link:  <https://graphdb.ontotext.com/documentation/free/using-graphdb-with-the-rdf4j-api.html#sparql-endpoint> | Link:  <https://www.researchgate.net/publication/300338060_RDFox_A_Highly-Scalable_RDF_Store>,  <https://docs.oxfordsemantic.tech/3.1/RDFox-shell-reference-commands.html> | Link:  <https://franz.com/agraph/support/documentation/current/sparql-reference.html> | Link:  <https://medium.com/virtuoso-blog/what-is-a-virtuoso-sparql-endpoint-and-why-is-it-important-5244df738a3e> |
| 23 | Api’s | Link:  <https://graphdb.ontotext.com/documentation/enterprise/using-graphdb-with-the-rdf4j-api.html>,  <https://graphdb.ontotext.com/documentation/enterprise/using-the-workbench-rest-api.html>,  <https://graphdb.ontotext.com/documentation/standard/quick-start-guide.html#load-data-through-sparql-or-rdf4j-api> (loading data into workbench through Sparql or rdf4j api) | Link:  <https://www.researchgate.net/publication/300338060_RDFox_A_Highly-Scalable_RDF_Store>,  <https://docs.oxfordsemantic.tech/programmatic-access-APIs.html> | Link:  <https://franz.com/agraph/support/documentation/6.4.0/python/api.html>,  <https://franz.com/agraph/support/documentation/current/#Programming-Interfaces> | Link:  <http://docs.openlinksw.com/virtuoso/rdfinsertmethodshttppost/>,  <http://vos.openlinksw.com/owiki/wiki/VOS/VirtSesame2Provider> |
| 24 | Temporal relations |  |  | Link:  <https://franz.com/agraph/support/documentation/current/magic-properties.html#sparql-magic-temporal> | \_ |
| 25 | Social Network  Analysis |  |  | Link:  <https://franz.com/agraph/support/documentation/current/magic-properties.html#sparql-magic-sna>,  <https://franz.com/agraph/support/documentation/current/magic-properties.html#Generators> | \_ |
| 26 | Multi-model Database | Link:  <https://graphdb.ontotext.com/documentation/free/mongodb-graphdb-connector.html> | Link:  [https://docs.oxfordsemantic.tech/data-sources.html#](https://docs.oxfordsemantic.tech/data-sources.html) | Link:  <https://allegrograph.com/products/allegrograph/> | Link:  <https://db-engines.com/en/system/AllegroGraph%3BGraphDB%3BVirtuoso> |
| 27 | Supports different formats for exporting Data | Link:  <https://graphdb.ontotext.com/documentation/standard/loading-data-using-ontorefine.html> | Link:  <https://docs.oxfordsemantic.tech/4.1/data-sources.html#data-sources> | Link:  <https://franz.com/agraph/support/documentation/current/agload.html#Introduction> | Link:  <http://docs.openlinksw.com/virtuoso/ch-webandxml/> |
| 28 | Hypergraph – Property Graph Permits | \_ | \_ | Link:  <https://allegrograph.com/products/allegrograph/>,  <https://neo4j.com/blog/other-graph-database-technologies/> | \_ |
| 29 | Free text Indexing | - | - | Link:  <https://franz.com/agraph/support/documentation/current/text-index.html> | Link:  <http://docs.openlinksw.com/virtuoso/rdfsparqlrulefulltext/>,  [http://docs.openlinksw.com/virtuoso/virtuosofaq8/, http://docs.openlinksw.com/virtuoso/ch-freetext/](http://docs.openlinksw.com/virtuoso/virtuosofaq8/,%20http://docs.openlinksw.com/virtuoso/ch-freetext/) |
| 30 | Concurrent execution of  queries | * (Not available in GraphDB Free and SE version )   Link:  <https://graphdb.ontotext.com/documentation/standard/graphdb-feature-comparison.html>,  <https://graphdb.ontotext.com/documentation/enterprise/ee/cluster-basics.html> | \_ | \_ | Link:  <http://vos.openlinksw.com/owiki/wiki/VOS/VOSVirtuoso6FAQ#How%20many%20concurrent%20queries%20can%20Virtuoso%20handle%3F> |
| 31 | SWRL- semantic web language |  |  | Link:  <https://www.w3.org/Submission/SWRL/>,  <https://docs.oxfordsemantic.tech/5.1/features-and-requirements.html> |  |
| 32 | DBaas service  Cloud | Link:  <https://www.ontotext.com/company/news/graphdb-cloud-dbaas/>  <https://www.ibm.com/cloud/learn/dbaas> |  |  |  |
| 33 | Dynamic Indexing | - | - | Link:  <https://allegrograph.com/products/allegrograph/> | \_ |
| 34 | Automatic Indexing | Link:  <https://graphdb.ontotext.com/documentation/free/storage.html> | Link:  <https://docs.oxfordsemantic.tech/4.0/structure-of-information.html> | Link:  <https://franz.com/agraph/support/documentation/current/triple-index.html> | Link:  <http://docs.openlinksw.com/virtuoso/rdfperfrdfscheme/> |
| 35 | ACID property Support | Link:  <https://db-engines.com/en/system/AllegroGraph%3BGraphDB>,  <https://www.ontotext.com/blog/triplestores-are-proven-as-operational-graph-databases/> | Link:  <https://docs.oxfordsemantic.tech/3.1/transactions.html> | Link:  <https://franz.com/agraph/support/documentation/current/agraph-introduction.html#ai-acid> | Link:  <http://docs.openlinksw.com/virtuoso/virtuosofaq30/> |
| 36 | Cloud Licensing | - | Link:  <https://aws.amazon.com/marketplace/pp/B08WJN3STV> | Link:  <https://aws.amazon.com/marketplace/pp/B08746KZHH>,  <https://allegrograph.com/products/allegrograph/> | Link:  <https://azuremarketplace.microsoft.com/en-us/marketplace/apps/openlinkswcom-pago.openlink-virtuoso-azure-pago-offer-20201019?tab=Overview>,  <https://aws.amazon.com/marketplace/pp/prodview-emqn2y66pu7xi> |
| 37 | Docker support | Link:  <https://hub.docker.com/r/ontotext/graphdb/> | Link:  <https://docs.oxfordsemantic.tech/docker.html> | Link:  <https://franz.com/agraph/support/documentation/current/docker.html> | Link:  <https://hub.docker.com/r/openlink/virtuoso-opensource-7> |
| 38 | MongoDB Support | Link:  <https://graphdb.ontotext.com/documentation/standard/mongodb-graphdb-connector.html> | - | Link:  <https://franz.com/agraph/support/documentation/current/mongo-interface.html> | \_ |
| 39 | Universal server |  |  |  | Link:  <https://en.wikipedia.org/wiki/Virtuoso_Universal_Server>,  <https://virtuoso.openlinksw.com/> |
| 40 | SQL support |  |  |  | Link:  <https://virtuoso.openlinksw.com/features-comparison-matrix/#interui> |
| 41 | WebDAV-Compliant Content Manager |  |  |  | Link:  <http://vos.openlinksw.com/owiki/wiki/VOS/VOSDAV>,  <https://dropdav.com/webdav-server-what-is-webdav/> |
| 42 | Federation of repositories/Queries | Link:  <https://graphdb.ontotext.com/documentation/free/internal-federation.html>,  <https://graphdb.ontotext.com/documentation/free/sparql-compliance.html#sparql-compliance-sparql-1-1-federation> | \_ | Link:  <https://franz.com/agraph/support/documentation/current/agraph-introduction.html#intro-federation> | Link:  <https://virtuoso.openlinksw.com/faq/>,  <https://thoughtsasaservice.wordpress.com/2012/05/17/sparql-1-1-and-openlink-virtuoso-first-steps-with-federated-queries/> |
| 43 | Web service deployment platform |  |  |  | Link:  <https://virtuoso.openlinksw.com/features/#5c32b0f7d6ee12ee700f64f95d992448>,  <http://docs.openlinksw.com/virtuoso/admui.webservices/> |
| 44 | Web services to RDF linked data |  |  |  | Link:  <https://virtuoso.openlinksw.com/features/#5c32b0f7d6ee12ee700f64f95d992448>,  <http://docs.openlinksw.com/virtuoso/linkeddata3simplesteps/> |
| 45 | RDF views over Native SQL Data sources |  |  |  | Link:  <https://virtuoso.openlinksw.com/features/#5c32b0f7d6ee12ee700f64f95d992448>,  <https://medium.com/virtuoso-blog/generate-relational-tables-to-rdf-relational-graphs-mappings-using-virtuosos-rdb2rdf-wizard-c4b83402599a> |
| 46 | Runtime hosting |  |  |  | Link:  <https://virtuoso.openlinksw.com/features/#5c32b0f7d6ee12ee700f64f95d992448>,  <http://docs.openlinksw.com/virtuoso/admui.runtimehosting/> |

Performance Features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GraphDB | RDFOX | AllegroGraph | Virtuoso |
| Performance feature | Benchmark used: LDBC Semantic Publishing Benchmark 2.0, Berlin SPARQL benchmark  Table 1: Loading time of the LDBC SPB-256 dataset with the default RDFS-Plus-optimized ruleset in minutes   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Editions | Rulesets | Explicit  Statements | Total  Statements | AWS  instances | Cores | Loading  Time in minutes | | 9.7 Free | RDFS-  Plus-  optimized | 237,802,643 | 385,168,491 | i3.xlarge | 1\* | 769 | | 9.7 SE/EE | RDFS-  Plus-  optimized | 237,802,643 | 385,168,491 | i3.xlarge | 2 | 692 | | 9.7 SE/EE | RDFS-  Plus-  optimized | 237,802,643 | 385,168,491 | i3.xlarge | 4 | 316 | | 9.7 SE/EE | RDFS-  Plus-  optimized | 237,802,643 | 385,168,491 | i3.2xlarge | 8 | 262 | | 9.7 SE/EE | RDFS-  Plus-  optimized | 237,802,643 | 385,168,491 | i3.4xlarge | 16 | 256 |   Table 1  Table 2: Loading time of the LDBC SPB-256 dataset with the default OWL2-RL ruleset in minutes   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Editions | Rulesets | Explicit  Statements | Total  Statements | AWS  instances | Cores | Loading  Time in minutes | | 9.7 SE/EE | OWL2-RL | 237,802,643 | 752,341,659 | i3.large | 2 | 1606 | | 9.7 SE/EE | OWL2-RL | 237,802,643 | 752,341,659 | i3.xlarge | 4 | 880 | | 9.7 SE/EE | OWL2-RL | 237,802,643 | 752,341,659 | i3.2xlarge | 8 | 652 | | 9.7 SE/EE | OWL2-RL | 237,802,643 | 752,341,659 | i3.4xlarge | 16 | 604 |   Table 2  *Berlin SPARQL benchmark:*  [BSBM](http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/) is a popular benchmark combining read queries with frequent updates. It covers a less demanding use case without reasoning, generally defined as ecommerce, describing relations between products and producers, products and offers, offers and vendors, products and reviews.  BSBM 100M query mixes per hour on AWS instance - c5d.4xlarge, local NVMe SSD with GraphDB 9.7 EE and ruleset RDFS-Plus-optimized   |  |  |  | | --- | --- | --- | | *Threads* | explore (query mixes per hour) | explore & update (query mixes per hour) | | |  | | --- | | 1 | | |  | | --- | | 10,963 | | 9,529 | | |  | | --- | | 2 | | |  | | --- | | 18,629 | | |  | | --- | | 16,806 | | | *4* | |  | | --- | | 33,751 | | |  | | --- | | 27,564 | | | *8* | |  | | --- | | 54,453 | | |  | | --- | | 37,911 | | | *12* | |  | | --- | | 67,506 | | |  | | --- | | 39,192 | | | *16* | |  |  | | --- | --- | |  | 72,660 | | |  | | --- | | 39,474 | |   *Link:*  [*https://graphdb.ontotext.com/documentation/enterprise/benchmark.html#ldbc-semantic-publishing-benchmark-2-0*](https://graphdb.ontotext.com/documentation/enterprise/benchmark.html#ldbc-semantic-publishing-benchmark-2-0)*,*  [*https://www.w3.org/wiki/LargeTripleStores#GraphDB.E2.84.A2\_by\_Ontotext\_.2817B.29*](https://www.w3.org/wiki/LargeTripleStores#GraphDB.E2.84.A2_by_Ontotext_.2817B.29)*,*  [*https://graphdb.ontotext.com/documentation/standard/rules-optimisations.html*](https://graphdb.ontotext.com/documentation/standard/rules-optimisations.html)*,*  [*https://graphdb.ontotext.com/documentation/standard/benchmark.html#berlin-sparql-benchmark-bsbm*](https://graphdb.ontotext.com/documentation/standard/benchmark.html#berlin-sparql-benchmark-bsbm) | Benchmark used: LUBM 5K and WatDiv benchmark  RDFOX can store between 1 and 1.5 billion triples in 50 GB.  On a computer two Xeon E5-2650 processors with 16 physical cores it materialized LUBM 5k in only 42s, a 10x speedup compared to using a single core  RDFox also loaded 19.47B triples (WatDiv benchmark) in 11041s on 64 threads, using 1.5TB of RAM.  *Link:*  [*https://www.w3.org/wiki/LargeTripleStores#RDFox\_.2819.5B.29*](https://www.w3.org/wiki/LargeTripleStores#RDFox_.2819.5B.29) | Benchmark used: LUBM 8000, SP2 benchmark  The total number of files read in is 160,007 N-Triples files, a total of 155 GB. The total number of triples after running the queries is 1.106 billion.  The total query time was 12 minutes and 56 seconds. They currently load LUBM 8000 dataset in just over 36 minutes.  The table below shows the results of running the LUBM(8000) queries, the results are reported in seconds:   |  |  |  | | --- | --- | --- | | *Lubm query* | *Number of triples* | *Time in seconds* | | *Query 1* | *4* | *0.007* | | *Query 2* | *2,528* | *278.321* | | *Query 3* | *6* | *0.004* | | *Query 4* | *34* | *0.027* | | *Query 5* | *719* | *0.076* | | *Query 6* | *83,557,706* | *389.062* | | *Query 7* | *67* | *0.014* | | *Query 8* | *7790* | *0.484* | | *Query 9* | *2,178,420* | *96.695* | | *Query 10* | *4* | *0.009* | | *Query 11* | *224* | *0.009* | | *Query 12* | *15* | *0.029* | | *Query 13* | *37,118* | *0.030* | | *Query 14* | *63,400,587* | *36.867* |   The [SP2 benchmark](http://dbis.informatik.uni-freiburg.de/index.php?project=SP2B) comprises a data-generator for arbitrarily large documents, which builds upon the well-known DBLP scenario, and thus comes close to a real-world application scenario. The benchmark’s queries make meaningful requests against this data while also testing typical SPARQL operator constellations and RDF access patterns. With this focus, the benchmark can help tune existing SPARQL engines and detect deficiencies in them. SP2 – 25 Million Dataset Results (Query Time in seconds)  |  |  |  |  |  | | --- | --- | --- | --- | --- | | *Queries #* | *Number of results* | *Query time #1* | *Query time #2* | *Query time #3* | | *1* | *1* | *0.006* | *0.004* | *0.001* | | *2* | *1,876,999* | *25.592* | *24.483* | *24.387* | | *3a* | *594,890* | *1.955* | *1.962* | *1.954* | | *3b* | *4,075* | *0.284* | *0.280* | *0.279* | | *3c* | *0* | *0.316* | *0.304* | *0.305* | | *4* | *113634340* | *982.168* | *1016.813* | *914.367* | | *5a* | *696,681* | *42.366* | *31.138* | *30.979* | | *5b* | *696,681* | *19.642* | *19.525* | *19.535* | | *6* | *1,945,167* | *45.790* | *46.425* | *46.443* | | *7* | *5,099* | *17.211* | *15.990* | *15.923* | | *8* | *493* | *0.323* | *0.204* | *0.206* | | *9* | *4* | *45.284* | *38.185* | *38.273* | | *10* | *656* | *0.003* | *0.001* | *0.001* | | *11* | *10* | *26.430* | *9.967* | *9.959* | | *12a* | *Yes* | *0.122* | *0.097* | *0.097* | | *12b* | *Yes* | *0.019* | *0.018* | *0.018* | | *12c* | *No* | *0.001* | *0.001* | *0.001* |   The platform for the test was 2 – 4 core Intel E5520 Processors at 2.26 GHz, with 48 GB RAM, running Fedora 14.  *Too see the results for 1 million and 5 million sp2 dataset, please see:* [*https://allegrograph.com/benchmarks-sp2/*](https://allegrograph.com/benchmarks-sp2/)  *Link:*  [*https://allegrograph.com/benchmarks-lubm-8000/*](https://allegrograph.com/benchmarks-lubm-8000/)*,*  [*https://www.w3.org/wiki/LargeTripleStores#AllegroGraph\_.281.2BT.29*](https://www.w3.org/wiki/LargeTripleStores#AllegroGraph_.281.2BT.29)*,*  [*http://dbis.informatik.uni-freiburg.de/index.php?project=SP2B*](http://dbis.informatik.uni-freiburg.de/index.php?project=SP2B) | *Benchmark used: LUMB 8000, Berlin SPARQL Benchmark*  *LUBM:*  *Load speed was 160,739 triples-per-second on a single machine with 2 x Xeon 5520 and 72G RAM. Adding a second machine with 2 x Xeon machine 5410 and 16G RAM, and 1 \* 1GigE interconnect, the load rate increased to 214,188 triples-per-second.*  *Link:*  [*https://www.w3.org/wiki/LargeTripleStores*](https://www.w3.org/wiki/LargeTripleStores)*,*  [*http://vos.openlinksw.com/owiki/wiki/VOS/VOSArticleLUBMBenchmark*](http://vos.openlinksw.com/owiki/wiki/VOS/VOSArticleLUBMBenchmark)  *Berlin SPARQL Benchmark:*  *The virtuoso 500 billion triple Berlin SPARQL Benchmark dataset runs were performed on a 24-node Elastic Cluster.*  *Please follow the link to see the results:*  [*http://vos.openlinksw.com/owiki/wiki/VOS/VOSArticle150BillionTripleDatasetLOD2StoreCluster*](http://vos.openlinksw.com/owiki/wiki/VOS/VOSArticle150BillionTripleDatasetLOD2StoreCluster) |