RDF Triple Store Analysis

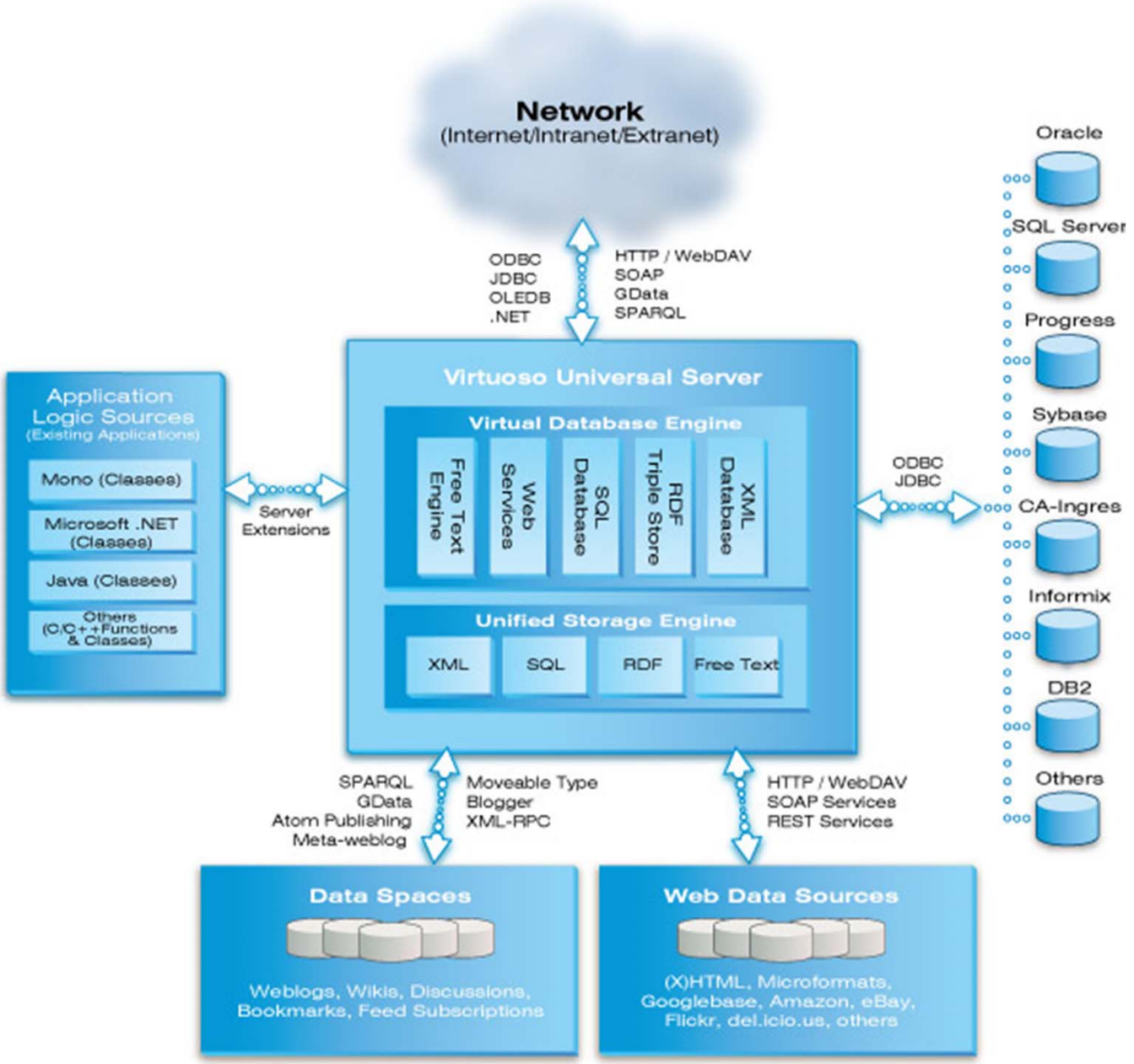
Large Triple Stores

http://www.w3.org/wiki/LargeTripleStores

1 OpenLink Virtuoso v6.1 - 15.4B+ explicit; uncounted virtual/inferred
2 BigOWLIM (12B explicit, 20B total); 100,000 queries per $1
3 AllegroGraph (20+B)
4 Garlik 4store (15B)
5 Bigdata(R) (12.7B)
6 YARS2 (7B)
7 Jena TDB (1.7B)
8 Jena SDB (650M)
9 Mulgara (500M)
10 RDF gateway (262M)
11 Jena with PostgreSQL (200M)
12 Kowari (160M)
13 3store with MySQL 3 (100M)
14 Sesame (70M)

Virtuoso

- Multi-purpose and multi-protocol (Hybrid) Data Server:
  - SQL Object-Relational, RDF, XML, and Free Text data management
  - Web Application (HTTP, SOAP, WebDAV)
  - SPARQL embedded into SQL for querying RDF data stored in Virtuoso's database
  - Java APIs (sesame, Jena)


Network
(Internet/Intranet/Extranet)
ODBC
JDBC
OLEDB
.NET
HTTP / WebDAV
SOAP
GData
SPARQL
Virtuoso Universal Server
Virtual Database Engine
Free Text Engine
Web Services
SQL Database
RDF Triple Store
XML Database
Unified Storage Engine
XML
SQL
RDF
Free Text
Application Logic Sources
(Existing Applications)
Mono (Classes)
Microsoft .NET (Classes)
Java (Classes)
Others (C/C++Functions & Classes)
Server Extensions
ODBC
JDBC
Oracle
SQL Server
Progress
Sybase
CA-Ingres
Informix
DB2
Others
SPARQL
GData
Atom Publishing
Meta-weblog
Moveable Type
Blogger
XML-RPC
HTTP / WebDAV
SOAP Services
REST Services
Data Spaces
Weblogs, Wikis, Discussions, Bookmarks, Feed Subscriptions
Web Data Sources
(X)HTML, Microformats, Googlebase, Amazon, eBay, Flickr, del.icio.us, others

Open Source vs. Closed Source

- Comparison Matrix

  http://virtuoso.openlinksw.com/features-comparison-matrix/
- Clustering & High Availability
- SPARQL GeoSpatial Extensions
- Data Replication
  - Snapshot
  - Transactional
  - Bi-Directional

Data Republication

- Replicate data between Virtuoso and non-Virtuoso servers
  - Snapshot
  - Transactional
  - Bi-Directional

Snapshot

- Non-incremental snapshot replication: the data is changed infrequently or when data is modified in large portions at a time.
- Incremental snapshot replication: when the data is changed frequently and implements incremental updates using a snapshot log.
- Bi-directional snapshot replication: allows data to be modified on both publisher and subscribers. This is useful read-write access to the replicas is needed.

Transactional Replication

- Allows subscribers to receive data in near-real time
- Ordinary transactional replication
- Bi-directional transactional replication

Virtuoso v6.1

- Open Linking Data Cloud Cache: http://lod.openlinksw.com/
- 15.4B+ explicit; uncounted virtual/inferred
- added ~3 Billion triples in ~3 hours — roughly 275Ktps (Kilotriples-per-second)

OWLIM

- Native RDF engines, implemented in Java and compliant with Sesame
- BigOWLIM 3.1: reasoning against 12.0 Billion explicit statements, + 8.4 Billion inferred statements;
- Loading and inference took 290 hours on a single server worth less than $5,000.
- BigOWLIM can deal with 1 Billion statements on a desktop machine worth $1000: it takes less than 5 hours to load the LUBM(8k) dataset at an average speed of 66 thousand statements per second.
- The "full cycle" run of LUBM(8k), including loading, inference, and query evaluation, takes 15.2 hours.

OWLIM

- Originally stands for OWL in memory
- IM only applies to SwiftOWLIM, not BigOWLIM, which uses a transactional, index-based file-storage layer.
- SwiftOWLIM and BigOWLIM are identical in terms of usage and integration for storing and managing RDF data.

OWLIM Versions

	Sesame version	Jena version	SPARQL	Instant initialisation	Advanced Features	Comment
SwiftOWLIM 2.9.x	1.2.x					The fastest OWL database. Multi-threaded inference, with transitive inference optimisation
SwiftOWLIM 3.x	2.x		√	√		The fastest OWL database engine with support for named graphs and SPARQL support
BigOWLIM 3.x	2.x	2.6.x	√	√	• Full Text Search • Geo-spatial extensions • owl:sameAs optimisation • RDF rank • RDF priming • Notifications • Replication Cluster	Ultimate scalability and fast SPARQL evaluation

SwiftOWLIM

- Designed for medium data volumes (below 100 million statements) and for prototyping
- reasoning and query evaluation are performed in main memory
- employs a persistence strategy that ensures data preservation and consistency
- the loading of data, including reasoning, is extremely fast
- easy configuration

BigOWLIM

- for handling massive volumes of data and very intensive querying activities
- file-based indices --- scale to billions of statements even on desktop machines
- special-purpose index and query optimization techniques
- optimized handling of owl:sameAs (identifier equality) to boost efficiency for data integration tasks
- efficient retraction of explicit statements and their inferences, which allows for efficient delete operations
- a range of powerful 'advanced features' including: Full text search (Node search, RDF search), ranking, selection and notifications

BigOWLIM

- **Cluster configuration** that supports load-balancing and automatic fail-over to provide even greater query performance and resilience
- **Scale-out handling of concurrent query requests:** the query processing rate scales linearly with the number of worker nodes
- **Resilience in the event of hardware/software failure:** automatic failover of cluster nodes and dynamic configuration

Storage Space (BIG)

- 1 million statements => ~200 Megabytes storage
- 1 billion statements => ~200 Gigabytes storage
- 10 billion statements => ~2 Terabytes storage

Jena TDB

- TDB has been used to load UniProt v13.4 (1.7B triples, 1.5B unique) on a single machine with 64 bit hardware (36 hours, 12k triples/s)
- pure-Java, employing memory mapped I/O, a custom implementation of B+Trees and optimized range filters for XSD value stapces (integers, decimals, dates, dateTime)

Jena TDB

- Can be accessed using both command line or Jena API
- Has some query optimazations

BSBM

Berlin SPARQL Benchmark (BSBM)

- Compare the performance of storage systems that expose SPARQL endpoints

February 2011 BSBM experiment

- 4store (version 1.1.2)
- BigData (rev. 4169)
- BigOwlim (version 3.4.3129)
- TDB (version 0.8.9)
- Virtuoso (version 7.00.3200-pthreads for Linux as of Jan 25 2011)

Data Loading Time

SUT	100M	200M
4store	26:42*	1:12:04*
BigData	1:03:47	3:24:25
BigOwlim	**17:22**	**38:36**
TDB	1:14:48	2:45:13
Virtuoso	1:49:26**	3:59:38**

* The N-Triples version of the dataset was used.

** The dataset was split into 100 respectively 200 Turtle fiiles and loaded with the DB.DBA.TTLP function consecutively.

Query Mixes per Hour

	100m	200m
4store	5589	4593
BigData	2428	1795
BigOwlim	3534	1795
TDB	2274	1443
Virtuoso	**7352**	**4669**

Number of query mixes with different parameters that are executed per hour
rate of query answering -- higher = better