



Open Source distributed document DB
for an enterprise

TECHNICAL OVERVIEW

PROJECT BAGRI: OVERVIEW

- Distributed system for storage and real-time processing of high volumes of semi-structured data documents
- Built on top of distributed cache solution like Coherence, Hazelcast, Infinispan, etc
- Designed to comply with enterprise level requirements providing high availability/fault tolerance/horizontal scalability/ACID transactions capabilities right out of the box
- Uses XQuery language for data processing
- Provides standard XQJ driver for client access
- Open sourced under Apache 2 license

THE MOST VALUABLE FEATURES

- Multi-document ACID transactions handling via multi-level concurrency control
- Does not require pre-registered schema, collects meta-data on the fly
- Can be used for any kind of semi-structured data processing
- XQuery requests are being transformed and optimized into queries against distributed cache
- Queries are performed in parallel on cache nodes and partitions
- Responses are streamed to clients asynchronously preventing client resources overload

UNDERLYING DISTRIBUTED CACHE PLATFORM

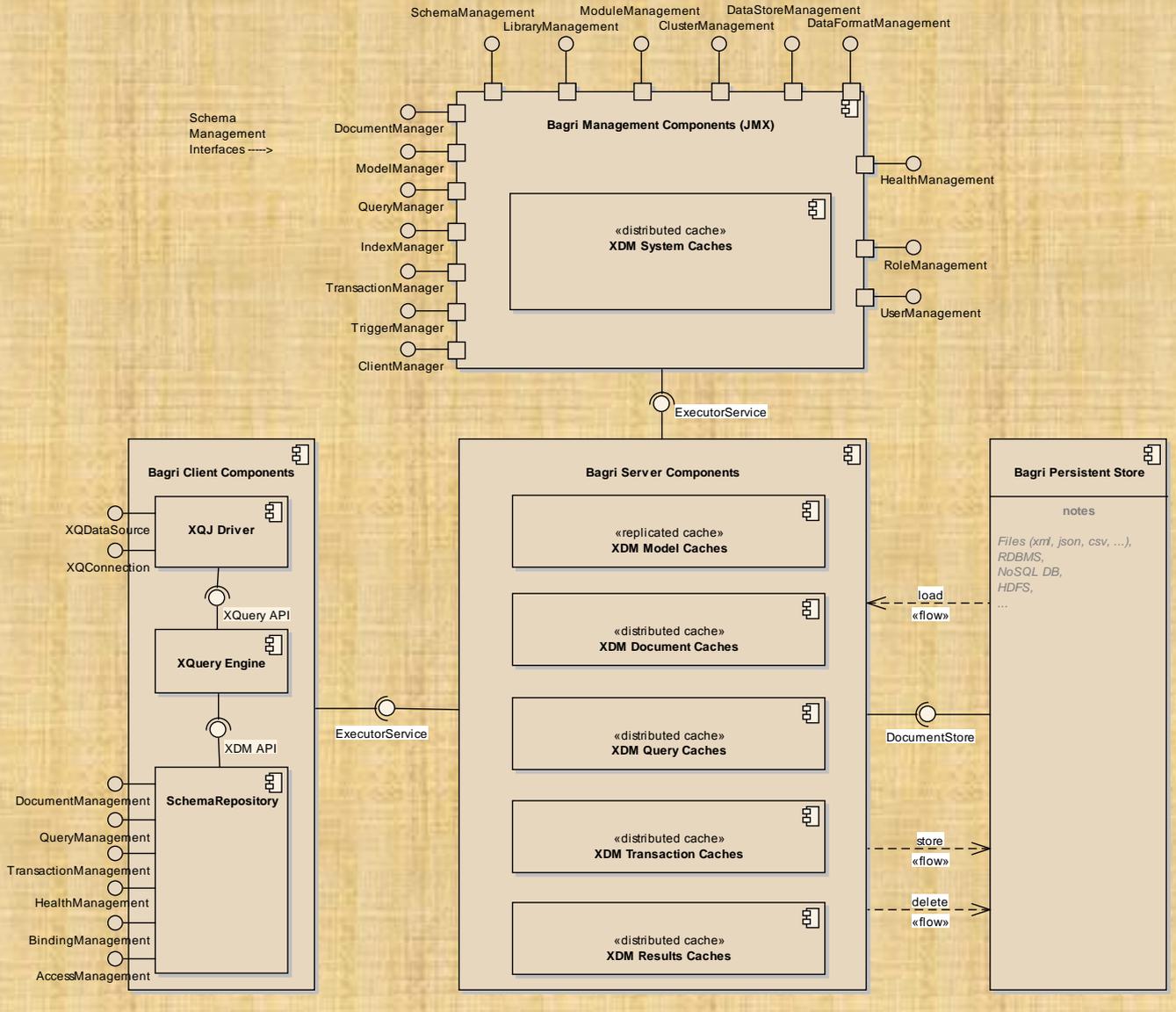
- Standalone Java SE app with built-in clustering capabilities
- Data distributed between cache partitions by consistent hashing algorithm applied on data keys
- Data can be accessed via Map interface or queried by predicates
- Used both as data and computation grid for massive parallel data processing
- Data processed in-place on the cache nodes by distributed tasks submitted by client; queries are also distributed
- Data can be loaded/stored in any kind of persistent store via simple and extensible CacheStore interface
- Clients access server cache interfaces via client proxies

HOW DISTRIBUTED CACHE CAPABILITIES ARE USED IN BAGRI

- All documents are stored in Schemas. Schema is like a database in RDBMS
- Every schema is handled by dedicated Distributed Cache cluster
- Document meta-data (unique paths) are stored in caches and replicated between all cluster nodes
- Document data are sliced and stored in distributed caches in share-nothing fashion
- Other distributed caches store indexed values, compiled queries, in-fly transactions, query results and other internal stuff
- Client connects to server via underlying cache mechanisms
- Client requests are wrapped to distributed tasks and processed on the server side
- Results are returned back via dedicated channel (queue) established between client and server

BAGRI COMPONENTS

cmp Bagri Components



Bagri deployment layers are:

- Clustered data servers
- Administration server(s)
- Persistent Store
- Clients

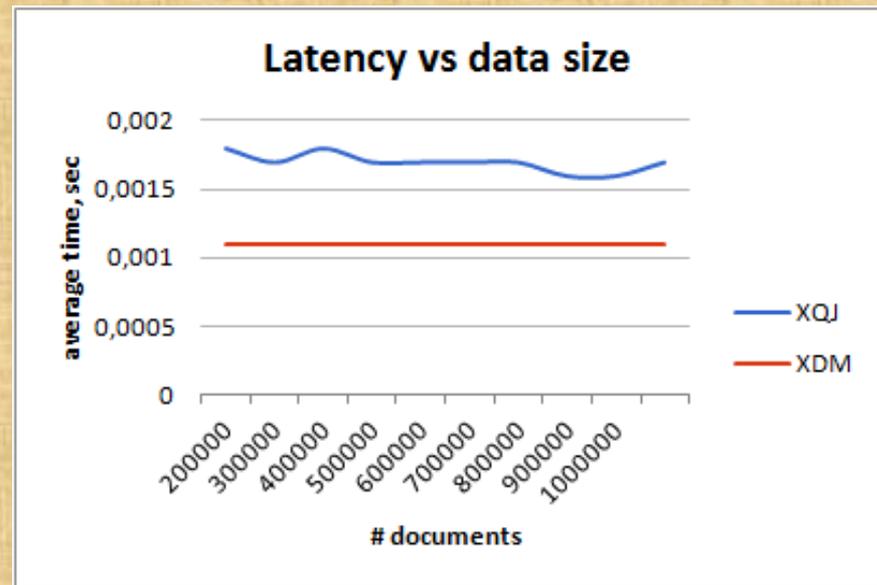
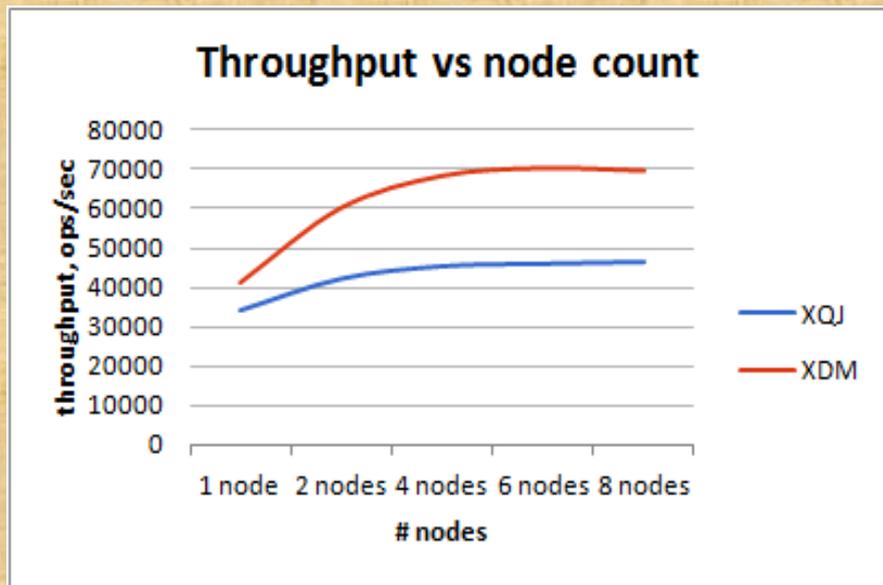


OTHER UNIQUE BAGRI FEATURES

- XQJ (JSR-225) and XDM (proprietary) client interfaces
- Full access control, role based security
- Wide indexes capabilities: unique, range, case-insensitive, XPath (wildcards)
- Server-side modules implemented in XQuery or in Java
- Custom triggers intercepting all document-level operations
- Java Binding allows client/server handling of documents as POJOs
- All document changes are versioned
- Document formats supported: XML, JSON, POJO, Map, ...
- Open extension APIs to connect external data formats and document stores
- ...

PERFORMANCE ANALYSIS

- The system demonstrates linear scalability with no performance degradation on volume increase

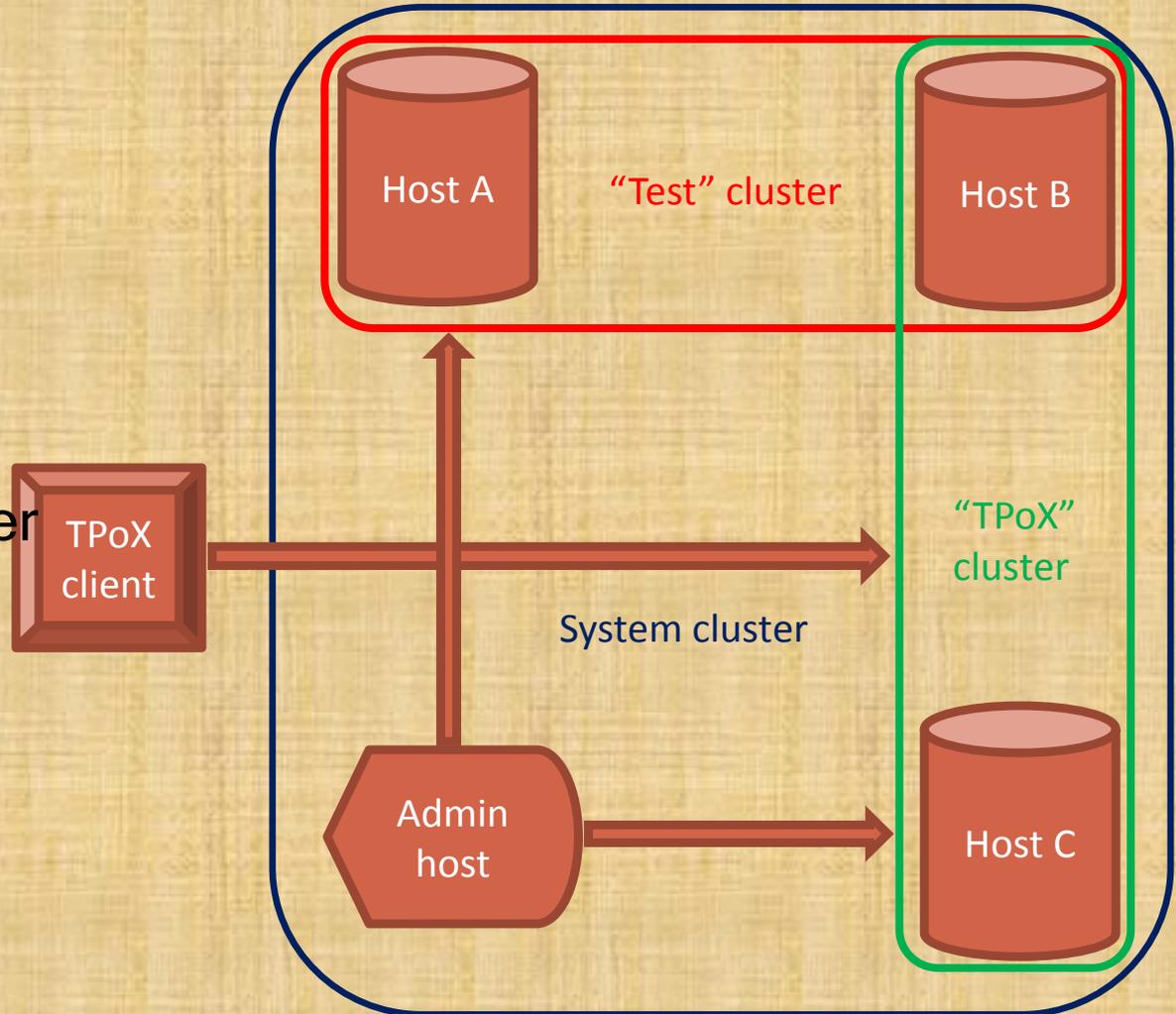


Servers: 2..8 nodes started on 2 metal boxes; 20 cores; 4G per server
Client: 1 metal box, 24 cores, 2G
Network: 10G

Throughput: 40..70K query/sec
Latency: 1,1..1,7 ms
XDM (direct) interface is 50% faster

DEPLOYMENT MODELS

- Standalone (embedded) mode
- Client-server (grid) mode
- Managed schema clusters
- Administration server (optional)

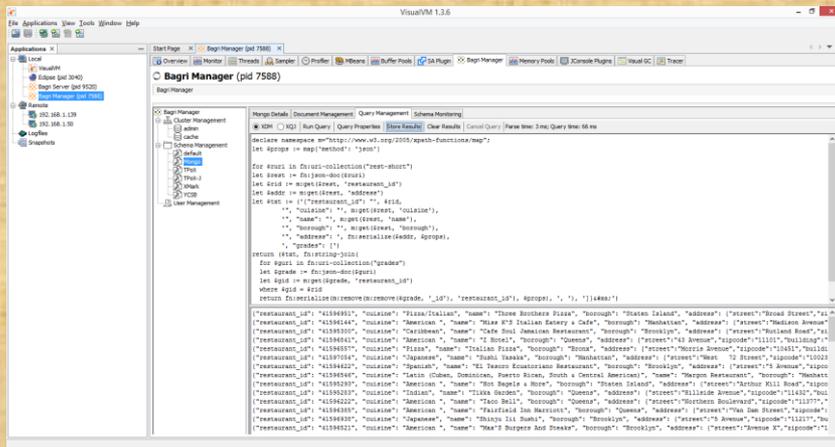
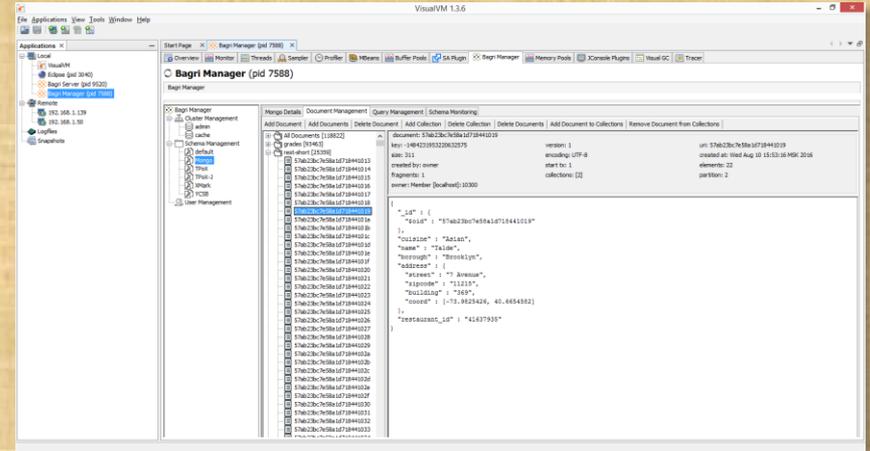


MANAGEMENT & MONITORING

- Management layer delivered as a dedicated Distributed Cache cluster
- All functionality exposed via set of MBeans deployed on administration server
- Rich API allows management and monitoring of every piece of system functionality:
 - Set of MBeans to handle Roles, Users, Nodes, Schemas, Modules, Libraries, Extensions
 - Set of MBeans to manage and monitor statistics on Schema details: Model, Documents, Indices, Queries, Transactions, Triggers, Clients..
- One central place to manage schemas, schema resources and schema access rights
- Can be used from any standard JMX console or via API

MANAGEMENT UI

- Implemented as a custom plugin for VisualVM tool
- Can be used to visually configure common and schema-related resources
- Provides explorer-style interface to work with documents and collections



- Provides query console to perform XQueries and see query results
- Actively developed, open for new features and suggestions



BAGRI COMPETITIVE ADVANTAGES

- True horizontal scalability, transparent redundancy & high availability inherited from the underlying distributed cache platform
- An ability to handle any high data volumes in real-time with memory access speed
- The well-known XQuery language is used for any kind of data manipulation and transformation
- Standard syntax for joins, sorting, grouping and many other valuable features; driven by industrial community
- Multi-document ACID transactions supporting all standard transaction isolation levels
- ...

CONTACTS

- Bagri site: <http://bagridb.com>
- Bagri project repository:
<http://www.github.com/dsukhoroslov/bagri>
- Bagri extensions project repository:
<http://www.github.com/dsukhoroslov/bagri-extensions>
- Bagri Google Group:
<https://groups.google.com/forum/#!forum/bagridb>
- Mail to: support@bagridb.com

Thank You!