High Performance Ingestion and Analytics

Saurav Gupta
Sales Engineer

InterSystems®
Creative data technology

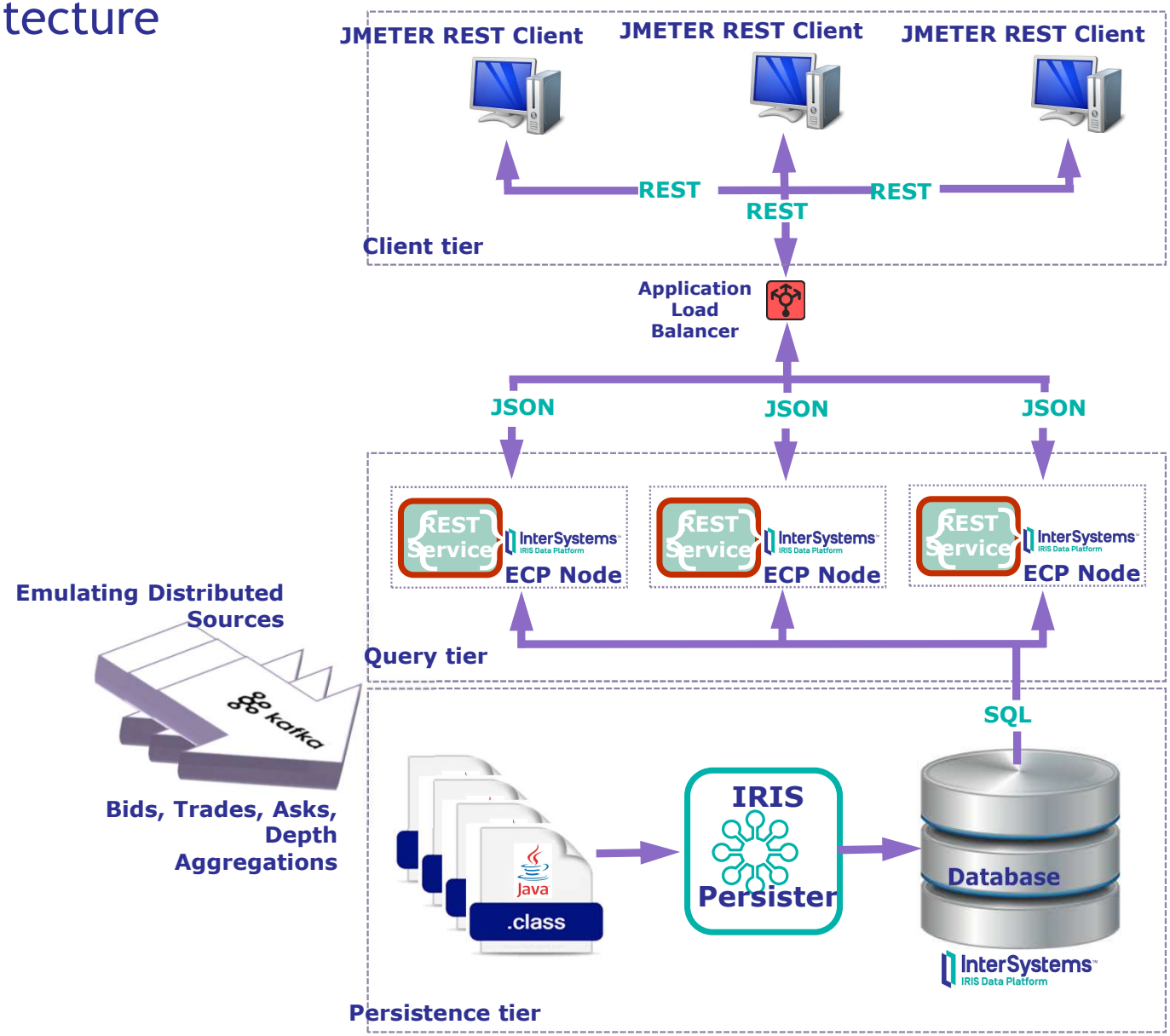
Sample Use Case – Equity Trades

- Store market data events and query real-time & historical projections of this data.
- Trades, Depth, Asks, Bids – Real time events across multiple financial instruments (volatile & non-volatile).
- Compute Aggregates (Statistics) - a trade event or a statistical update event from the source system can affect the last price, open price, high price, low, close, cumulative volume etc.
- REST APIs to access data as JSON
- Real Time access to data

Challenges

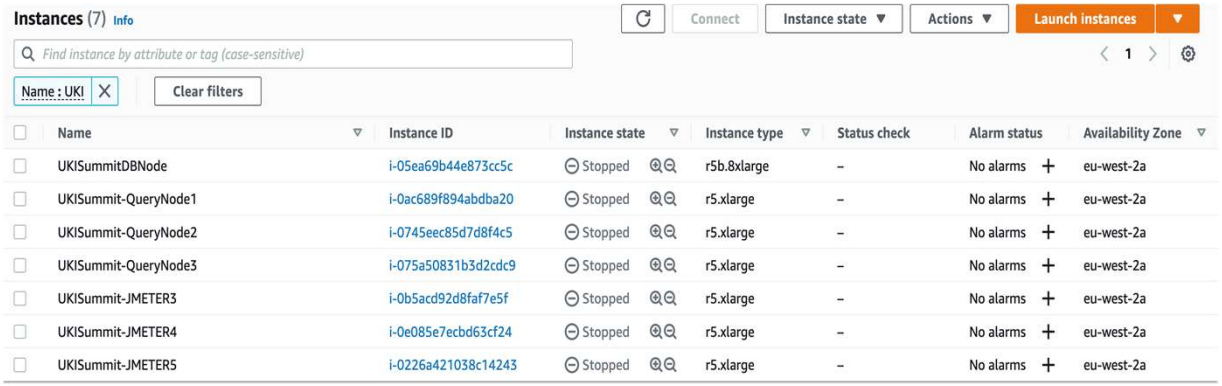
- Concurrent Ingestion and real time analytics(query)
  - 1 million inserts/second
  - 1000 REST APIs per second
- In-memory databases offer high performance but are expensive to scale
- Different databases for ingestion and query workloads creates multiple copies of data and synchronization issues
- Scaling beyond 500K Inserts/Second and keeping indexes up to date to make data available for query workloads
- Complex Architectures

Architecture



Sample Use Case Components

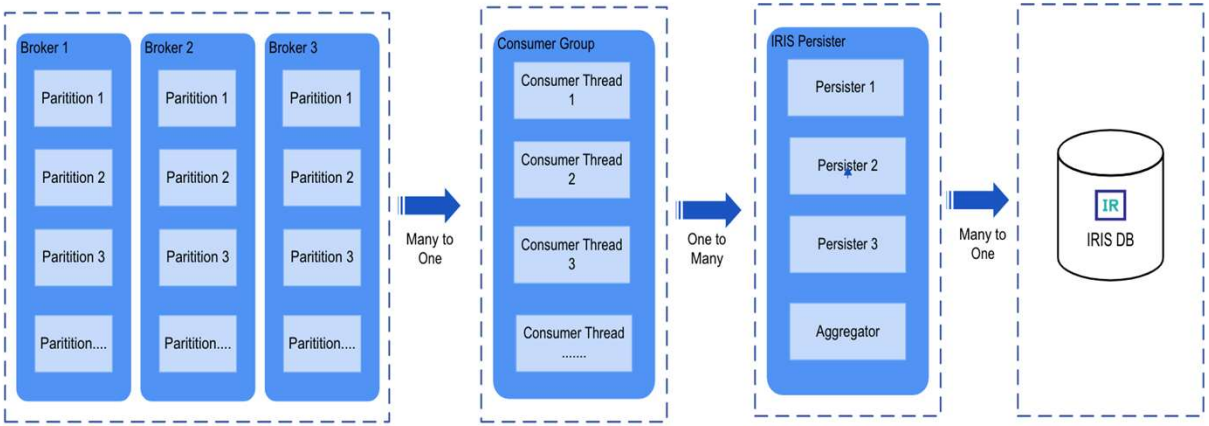
- Sample Data
  - 1 day of JSON Data
  - Average Size of message – 500 Bytes
  - No of Symbols – 140K
- Sample Data Provided on S3 Buckets loaded into KafKa Cluster (Amazon MSK)
  - 100 partitions
  - 3 brokers with each broker of 800GB size
- IRIS Persister/Loader loads data from KafKa broker into a single IRIS DB Node
- 3 IRIS Query Nodes using ECP being used for query workloads being serviced through REST APIs
- 3 JMETER REST Client Nodes being used to send concurrent REST Requests



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
UKISummitDBNode	i-05ea69b44e873cc5c	Stopped	r5b.8xlarge	–	No alarms	eu-west-2a
UKISummit-QueryNode1	i-0ac689f894abdba20	Stopped	r5.xlarge	–	No alarms	eu-west-2a
UKISummit-QueryNode2	i-0745eec85d7d8f4c5	Stopped	r5.xlarge	–	No alarms	eu-west-2a
UKISummit-QueryNode3	i-075a50831b3d2cdc9	Stopped	r5.xlarge	–	No alarms	eu-west-2a
UKISummit-JMETER3	i-0b5acd92d8faf7e5f	Stopped	r5.xlarge	–	No alarms	eu-west-2a
UKISummit-JMETER4	i-0e085e7ecbd63cf24	Stopped	r5.xlarge	–	No alarms	eu-west-2a
UKISummit-JMETER5	i-0226a421038c14243	Stopped	r5.xlarge	–	No alarms	eu-west-2a

KafKa and InterSystems IRIS Persister

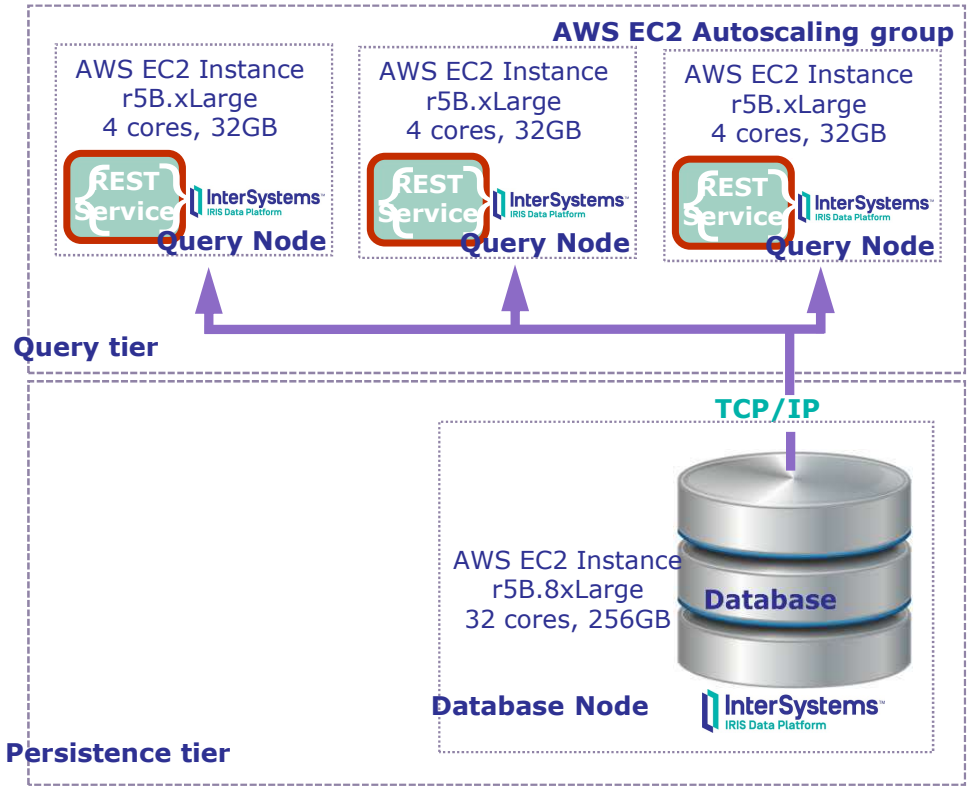
- KafKa Consumer Group
  - Related consumers with a common task
  - KafKa sends message from partitions of a topic to consumers in the consumer group
- Kafka Streams
  - Stream processing Library
  - Supports aggregations
- IRIS Persister
  - Store objects on an InterSystems IRIS DB Node
  - Multi-threaded Loader that can be used to ingest large data sets
  - The Loader consumes a data stream, serializing each record and writing each serialized record to a pool of output buffers, each of which maintains a separate connection to an InterSystems IRIS Server.



InterSystems IRIS Persister

Obtains a connection to InterSystems IRIS Server	Connections to a server can be obtained directly or by using an IRISDataSource.
Get a Schema Manager Instance	Schema Manager can be instantiated directly once a connection is available.
Getting a Schema	Define the schema using JSON, load a previously defined schema from a file, receive a schema from an external source such as a Kafka message, load a previously defined schema from the server or generate a schema from server class
Get an IRIS Persister Instance, passing schema Manager and schema as arguments	Interface that stores data in the extent of the Server's schema local implementation class
Use the IRIS Persister to store Records in the extent of the schema's InterSystems IRIS Server implementation class	Index Mode – Immediate or Deferred

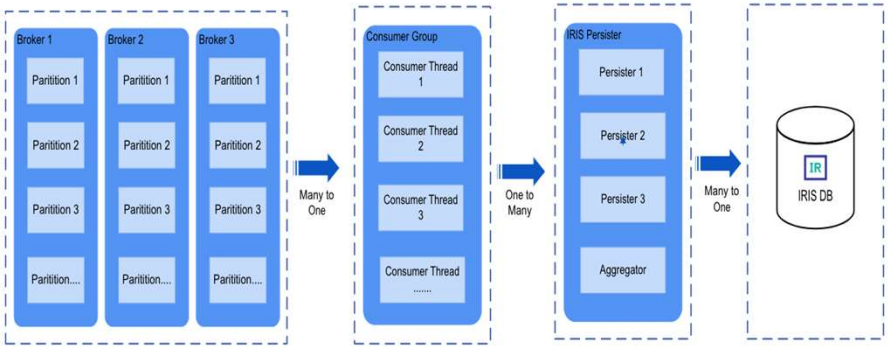
InterSystems IRIS Horizontal Scaling with Distributed Caching



- The data server continues to store, update, and serve the data. The data server also synchronizes and maintains the coherency of the query server caches to ensure that users do not receive or keep stale data, and manages locks across the cluster.
- Each query against the data is made in a namespace on one of the various query servers, each of which uses its own individual database cache to cache the results it receives; as a result, the total set of cached data is distributed across these individual caches.
- User requests can be distributed round-robin across the query servers by a load balancer
- The number of query servers in a cluster can be increased (or reduced) without requiring other reconfiguration of the cluster or operational changes, so you can easily scale as query volume increases.

Ingestion Speed

- 1 million sustained inserts/second with journaling ON for reliable persistence using a single DB Node



AWS DB EC2 Instance Type	Specifications
R5b.8xLarge	32 Cores 256GB Memory
Storage	Storage Volumes Root Device – Install Directory EBS block storage –Database volume and journaling volume

Query Speed

- 200 – 250 messages/second(REST API) on a single query node with 10 symbols per message

JMETER Node	Symbols Per Rest API	Mean RPS	Average Response Times(ms)	Maximum Response Times(ms)
1	10	450-460	16	285
2	10	450-460	16	278
3	10	450-460	16	306

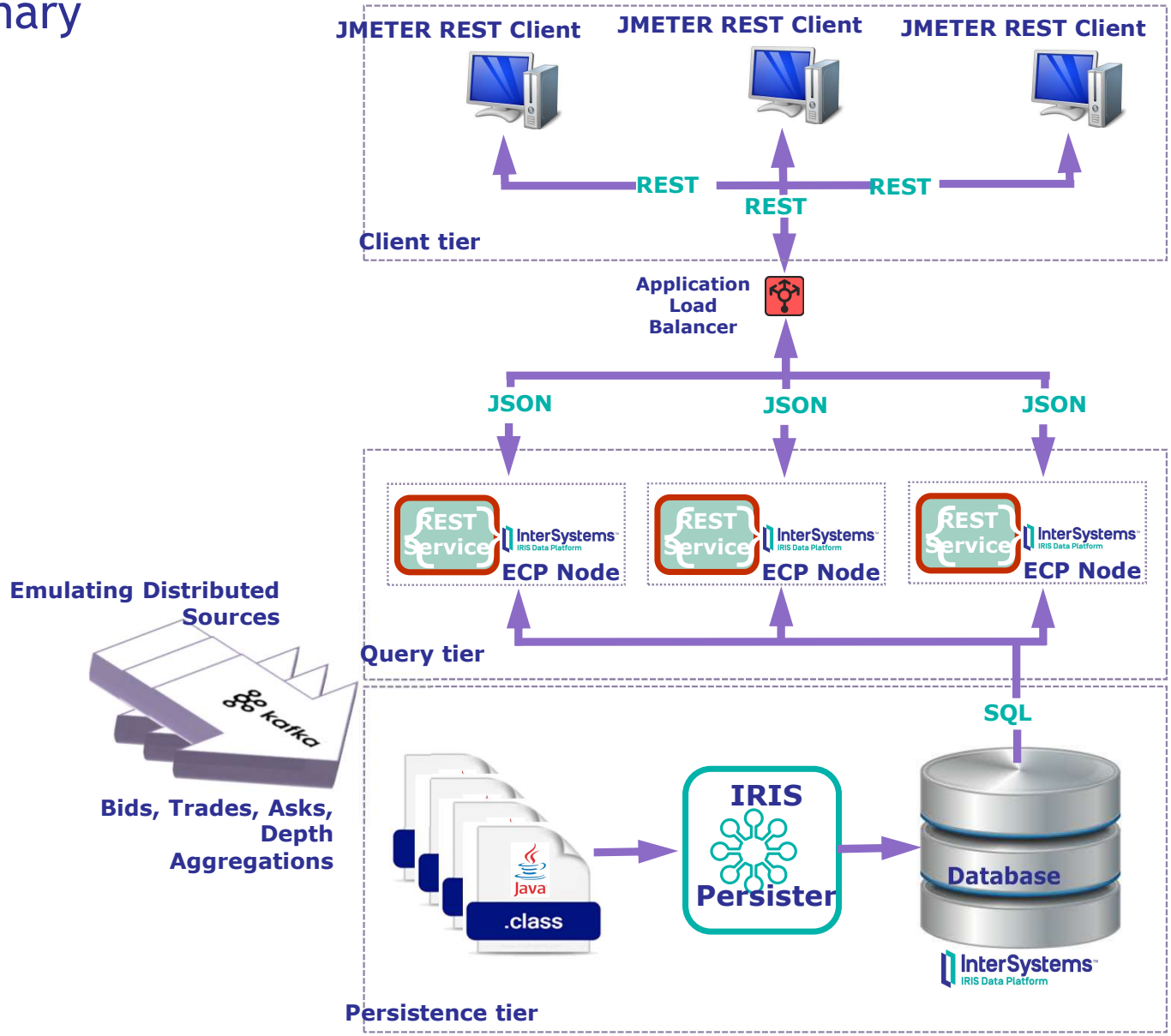
JMETER Node	Symbols Per Rest API	Mean RPS	Average Response Times(ms)	Maximum Response Times(ms)
1	30	180-190	40	593
2	30	180-190	40	631
3	30	180-190	40	592

AWS DB EC2 Instance Type	Specifications
R5b.xLarge	4 Cores 32GB Memory
Storage	Storage Volumes Root Device – Install Directory EBS block storage – Database volume and journaling volume

Demo

- Store market data events and query real-time & historical projections of this data.
- Trades, Depth, Asks, Bids - financial instruments (volatile & non-volatile).
- Compute Aggregates (Statistics) - a trade event or a statistical update event from the source system can affect the last price, open price, high price, low, close, cumulative volume etc.
- REST APIs to access data as JSON
- Real Time access to data

Summary


JMETER REST Client
JMETER REST Client
JMETER REST Client
REST
REST
REST
Client tier
Application Load Balancer
JSON
JSON
JSON
REST Service
InterSystems IRIS Data Platform
ECP Node
REST Service
InterSystems IRIS Data Platform
ECP Node
REST Service
InterSystems IRIS Data Platform
ECP Node
Emulating Distributed Sources
kafka
Bids, Trades, Asks, Depth Aggregations
Query tier
SQL
Java
.class
IRIS
Persister
Database
InterSystems IRIS Data Platform
Persistence tier

Thank you

InterSystems®
Creative data technology