System Architecture and Implementation Discussion

Security

Secure Sign-up using OAuth

Secure Payment Gateway with Credit Card Company

Secure User Transaction
— A very critical operation : should be thread-safe and highly consistent operation .
— Ideally use JVM language to ensure 'low-level lock is acquired' before payment transaction executed . If possible use closure's STM and Managed Mutability concept to achieve maximum immutability and thread-safety.

** We can use bcrypt+salt for encrypting / decrypting sensitive info **
Here goes my experimentation : https://phisymmetry.wordpress.com/2015/03/10/secure-mongodb-server/

API

— prefer combination of java/scala/closure for 'secure transactional api'
— asynchronously log messages (api usage, event flow, errors) - preferably to Elastic Search
— publish metrics (Java API can use Servo) - for faster Alerts and Notifications
— roll up the exceptions
— prefer loosely-coupled , asynchronous , stateless modules
— API should ensure internal core implementation properly encapsulates Visitor (Iterator) , Serializable (as required) , Transient / Volatile , List of Listeners , Observables , Behaviors

— more detailed captured in my blogpost : https://phisymmetry.wordpress.com/2012/10/19/designing-better-api/

— make the api fault-tolerant and self-healing (by leveraging Hysterix and Rx-Obseravble)
  e.g. https://github.com/pmotyka/RxJavaHystrixDemo/blob/master/MovieService/src/main/java/com/example/movie/RatingCommand.java

— Here I have documented some of the latest best practices — https://phisymmetry.wordpress.com/2016/01/18/coding-and-scripting-to-build-scalable-systems-lessons-learnt/

— API Authentication : (TODO) attach links to *best practices from StormPath and SalesForce*

— we can use node.js for 'low-latency query / lookups' and for parallelizing the query executions.
Hapi is a robust Nodejs server.

- more details on Nodejs best practices in my blog : https://phisymmetry.wordpress.com/2014/11/24/learning-mean-stack/

Data Models

We need following types of tables :
— Staging Data (denormalized streams)
— Primary Entities (normalized/ semi-normalized structures : User Profile , Transaction , Vendor , Vending Machine etc.)
— Analytics Data (pre-aggregated analytics usecases)
— Logs and Time-series Events

here is my blogpost (WIP) - Approaches for data modeling / query :

https://phisymmetry.wordpress.com/2015/05/12/data-modelling-patterns-usecases/

For Medium Dataset (GB)

Staging Data : Store say 6 months of raw data in a Mongo / Couchbase Capped Collection and then archive into S3 / HDFS.

Primary Entities :
—e.g. 'User Payments and Balance info' can be stored in a MongoDB Schema with strong consistency (otherwise into MySQL to guarantee auto-rollback and stricter consistency)

— store User Profile info, Vendor info, Vending Machine registry into MongoDB / Couchbase

RDBMS not suitable for Low-latency Analytics Analytics :

- Huge Overhead of – Record Level Locking
- Lacks notion of implicit ordering
- Overhead of buffer pool is too big
- Overhead of latching threads
- No support for ROLAP
- No notion of Shared Nothing Grid Based Architecture

quick analysis Columnar vs RDBMS - https://phisymmetry.wordpress.com/2012/11/22/why-should-we-adopt-columnstore-as-new-datawarehouse/

User {
ProfileInfo : {
  name: … , address: … , gender: ….
},
Preferences : {
  [chocobar , chips ..]
},
Balance : {
 current: 20, topup: 1
},
Feedback : {
  [vending_m1: {rating: satisfied , comments: …}, vending_m2: {rating: bad, comments: … }]

}}

Logs and Time-series Events
We can use a Capped Collection for Transaction logs

Transactions

[{userId:"", timestamp: "" , transaction_amount: "", success : ""},
{userId:"", timestamp: "" , transaction_amount: "", success : ""},
{userId:"", timestamp: "" , transaction_amount: "", success : ""}],

— pump the logs and time-series data into Elastic Search

Analytics Schema :

for sample schema refer to : https://github.com/data-mining/demo/edit/master/payrange_demo/README.md

For Big Dataset (TB)
— Cassandra for modeling all types of tables and varieties of use-cases :

*** caution ** We need to enforce 'Strong Consistency and apply Application-level locking and rollback and proper exception handling' -> user Balance Updates

1) User Profile :
— RowKey : 1001 [Name : Bill] , [Email:abc@def] , [City: New York] , [Bday: 7 May, 1980]
— RowKey : 1001 [Name : Joy] , [mnp@def]

The advantage is the KeySpace (a.k.a.) Table can have dynamic column structure in each row

2) Unstructured Logs from Mobiles

— RowKey : <session_id / mobile_phone_no > [Timestamp1 : Event_Log1] , { Timestamp2, Event_Log2]

3) **Design Column Family as per Query Pattern**

a) Vending Machine Usage data
Row-Key: VM01 : [user1 : 25] , [user2 : 30] ….

b) **Vending Machine Daily , Weekly, Monthly Aggregations** - in the same Row as different Column Families

Vending_Machine_1	Daily	Weekly

	Avg Txn Time	Max Txn Time	Min Txn Time	Avg Txn Time	Max Txn Time	Min Txn Time
Vending_Machine_2	Daily			Weekly		
	Avg Txn Time	Max Txn Time	Min Txn Time	Avg Txn Time	Max Txn Time	Min Txn Time

c) User Time with Vending machine

User1	Daily			Weekly			Monthly		
	Avg Time spent	Max Time spent	Min Time spent	Avg Time spent	Max Time spent	Min Time spent	Avg Time spent	Max Time spent	Min Time spent
User2	Daily			Weekly			Weekly		
	Avg Time spent	Max Time spent	Min Time spent	Avg Time spent	Max Time spent	Min Time spent	Avg Time spent	Max Time spent	Min Time spent

d) User purchases

User1	purchases		
	[product1: #purchaes]	[product2: #purchaes]	[product3: #purchaes]
User2	purchases		
	[product1: #purchaes]	[product2: #purchaes]	[product3: #purchaes]

e) **best-selling products**

Overall Sales : [product1: #sales] , [product2: #sales]

Sales per Provider :
Row-Key : vending_machine_provider [product1 : #sales] , [product2: #sales]

f) **Combine co-related aggregations for a specific Vending_Machine_Provider**

Vending_Machine_Provider1	SFO			Revenue		
	Machine1_inventory	Machine2_inventory	Machine3_inventory	Machine1_revenue	Machine2_revenue	Machine3_revenue
Vending_Machine_Provider2	Portland					

g) **User Demographics in an area for specific products**

RowKey - Zipcode1	**product1**		**product2**		**product3**	
	male	female	male	female	male	female
RowKey- Zipcode2	**product1**		**product2**		**product3**	
	male	female	male	female	male	female

Advanced Scenarios

- **Infer the causes/conditions which cause transaction to fail**
  - i) cause : Low Balance :
    - run a daily / 6hrs check on 'user account balance' to see if balance is very low and send alerts to users accordingly
  - ii) cause : n/w connectivity
    - if transaction doesn't succeed due to n/w connectivity within certain time - retry few times
  - iii) learn the causes from past incidences (Vending Machine issues , Transaction logs)
  - iv) offer 'small credits' to users so that they can complete transactions
-
-
-

- **Infer the cause for lower usage of Payrange in certain areas**
  - regularly calculate volatility (standard deviations) of transaction and usage data and find the anomalies by detecting outliers (say falling outside Lower Bollinger Band i.e. mean - 2*sd)
  - **correlate the outliers with 'Vending Machine provider data' and 'Products data'** to find if 'Providers have good rating' / 'Machines are functional' / 'desired products are available'
  - **analyze demographic distribution - may be old age people / non tech-savvy people prefer manual transactions**
  - analyze the user feedback to understand if food quality has dropped of hints on other issues.
  - collect data from independent surveys

- **How demographics in an area/zip relates to the transactions and transaction values**
  - As explained in the data model (g) —> we can perform statistical measurement against affect of demographics on products and places
  -
- **Analyze User Feedback and Review**
  - Provide a structured model to capture reviews / feedback / issues from Users & Vendors
    - so that by analyzing the attributes informed decision can be taken
      - say replenish inventory
      - offer favorite items to users
      - rate the Vendors

- **Vendor Recommendations**
  - perform **Collaborative Filtering** to find what users like most and accordingly share the insights with Vendors.
  -
- **Build Cluster of User Interests and Correlated Products**
- **Reward User Engagement with Top-ups**

**** Future Scope ****

> How healthy are my food items in vending Machine ?
— since 'Food Items' directly related to 'Health and Wellness'
— calculate / collect health score of food items from independent sources and user feedback
— find out if unpopular / retired items still in Vending Machine

> Eventually build a **Domain Ontology** , e.g. Vendor-VendingMachine-Product Graph

More on Data Science and Analytics

some of my crawlers - https://github.com/kaniska/data-mining-usecases

\some of my work / learning using Spark In-memory computing Workflow :

>> https://phisymmetry.wordpress.com/2015/06/10/develop-a-lightning-fast-analytics-app-using-spark/

>> https://github.com/kaniska/spark-analytics

>> my blog on recent usecases and advancement in data science : https://phisymmetry.wordpress.com/2015/08/02/advancements-in-data-analytics/

>> my work on Algorithmic Trading Forecasting (ongoing MS - CS) https://github.com/kaniska/AlgorithmicTrading

>> some more work on KBAI - https://github.com/kaniska/Knowledge-Based-Artificial-Intelligence

>> some R Analytics Work as part of 'practical data science' course from Coursera : https://github.com/kaniska/R_Workshop

>> worked on creating a library for comparing Machine Learning Algos : https://github.com/kaniska/machine_learning_library

>> some work on realtime-analytics using Elastic Search : https://github.com/kaniska/elasticsearch-percolator-real-time-analytics

>> my old work on MongoDB Analytics - https://github.com/kaniska/project-mongo-logviewer , my recent training on Couchbase Analytics : https://github.com/kaniska/couchbase_analytics

Enterprise Data Search

>> earlier built Enterprise Search Apps with faceted widgets - leveraging https://github.com/kaniska/enterprise-search-solr-cassandra

built Drugs Search - API by boosting the Ranks of Queries
>> https://phisymmetry.wordpress.com/2014/01/18/solr-ranking-by-query-boosting-and-other-methods/

Data Ingestion

Here goes the compilation of some of my thoughts — https://phisymmetry.wordpress.com/2015/09/26/real-time-streaming-etl-and-real-time-bi-part-2/ (WIP)

Approach A: Offline batch processing

Strategy - (a)

Traditional Batch processing+ETL

—> **(1)** Flurry / Data Feeds

—> **(2)** Boomi Integrator / DataTorrent Apex / Homegrown Consistent Batch Scheduler

: we need to maintain a **time-machine of batch data** (Batch-1 started T1 and successfully ended at

T2 , so Batch-2 should start from T2 onwards)

: if a Batch fails - retry (1) + (2) to fetch data within the time-window (for previously failed batch)

—> **(3)** Staging : if we use off-line batch-processing ETL , then data need to be staged into HDFS for large datasets or into Cassandra (**Primary Datastore**) .

—> For online real-time ETL Storm can be used for big data compute-merge or Spring Batch/Integration for medium dataset ==> create/update pre-aggregate tables in Analytics Datastore

—> **(4)** **—> 4.a** Cassandra (**Analytics Datastore**)

—> 4.b Elastic Search (**Search Store**) - *Geo JSON / Term Cluster / Timeseriese in Kibana*

—> **(5)** Machine Learning using Spark Workflow / python + H2O

Strategy - (b)

If there is any use case for syncing up data between mobile and server periodically , then we can use

Couchbase Mobile Client which can auto-sync up with Couchbase Server (whenever connection available) without requiring any separate data-scheduler / investor / integrator.

Raw data persisted in Couchbase (Staging Datastore) can be pumped into Analytics Datastore (Data Mart) using ETL pipeline like —> Kafka + Storm + Cassandra / Redis

Approach B: large-scale continuous real-time data

Strategy - (a) **Mobile push Notification** -> AWS SQS / Apache Kafka

-> Real-time Analysis using Spark

-> store **raw data** and **pre-computed aggregations** in Cassandra

—> use Redis for small set of lookups (post-analysis results)

Strategy - (b) **Continuous merge-sync using Lambda Architecture**

- For fast real-time feeds if both historical data and current data always need to be merged and synced , Apache Samza (as used by Linkedin) can be used as **implementation of Lambda Architecture**
- **Samza** offers a great ‘consistent view of entire data set - specially in a sliding window’

my work for dynamic object persistence : https://github.com/kaniska/Dynamic-Object-Persistence

*** Alongside the 'Data Ingestion Software' **— we should also use fast Non-blocking backend servers like Nginx / Netty**

System and Apps Performance Improvement

some of my work on performance improvement :
Challenges with streaming large datasets
>> this was a very interesting work : https://phisymmetry.wordpress.com/2013/02/10/how-to-surmount-challenges-in-streaming-massive-data-sets/

Bigdata performance improvement :
>> https://phisymmetry.wordpress.com/2012/11/22/some-thoughts-on-improving-hbase-performance/
>> https://phisymmetry.wordpress.com/2014/02/15/hadoop-performance-analysis-and-tuning/

Building Scalable Distributed Systems

CAP : Understanding at least one use case of how Distributed Database System works - helps understand how CAP theorem works .

here goes my experience — https://phisymmetry.wordpress.com/2015/03/05/mongodb-fault-tolerance-strategy/

Quick summary of my experience of building Multi-Tenant SaaS App :

https://phisymmetry.wordpress.com/2012/02/12/building-multi-tenant-platform-for-data-acquisition-analysis-visualization/

some more details on building Enterprise Data Platform : https://phisymmetry.wordpress.com/2014/05/18/enterprise-data-analysis-platform-as-a-service/

Data Orchestration and Job Workflow

Bigdata Pipeline : https://github.com/kaniska/bigdata_analysis_pipeline (brief steps)

some of my work on building Job Pipelines using MEAN Stack : https://phisymmetry.wordpress.com/2015/01/18/data-automation-dynamic-workflow-enabler/

my work on data-mining :

Analytics Dashboards

Its very important to Actionable Analytics - not just 'Static Dashboards'

Here is a brief explanation of my work on — 'Feedback Loop through Actionable Analytics' be established — **https://phisymmetry.wordpress.com/2015/01/18/continuous-automated-actionable-analytics/** (WIP)

Now lets focus on 'How to build Quickfire Dashboards'

(1) Connect Analytics Datastore to Tableau
— Tableau + MongoDB : http://www.tableau.com/about/blog/2015/6/tableau-mongodb-visual-analytics-json-speed-thought-39557
— Tableau + Cassandra : http://www.datastax.com/dev/blog/datastax-odbc-cql-connector-apache-cassandra-datastax-enterprise

(2) **Ad-hoc SQL Query using PrestoDB** against MongoDB / Cassandra
— PrestoDB helps fire SQL Query against any Big datastore (Hive / Mongo Cassandra)

(3) **Custom D3 Charts**
— e.g. https://tomneyland.github.io/angular-dc/example/stocks/nasdaq.html , http://adilmoujahid.com/posts/2015/01/interactive-data-visualization-d3-dc-python-mongodb/ , https://github.com/data-mining/DonorsChoose_Visualization/blob/master/static/js/graphs.js

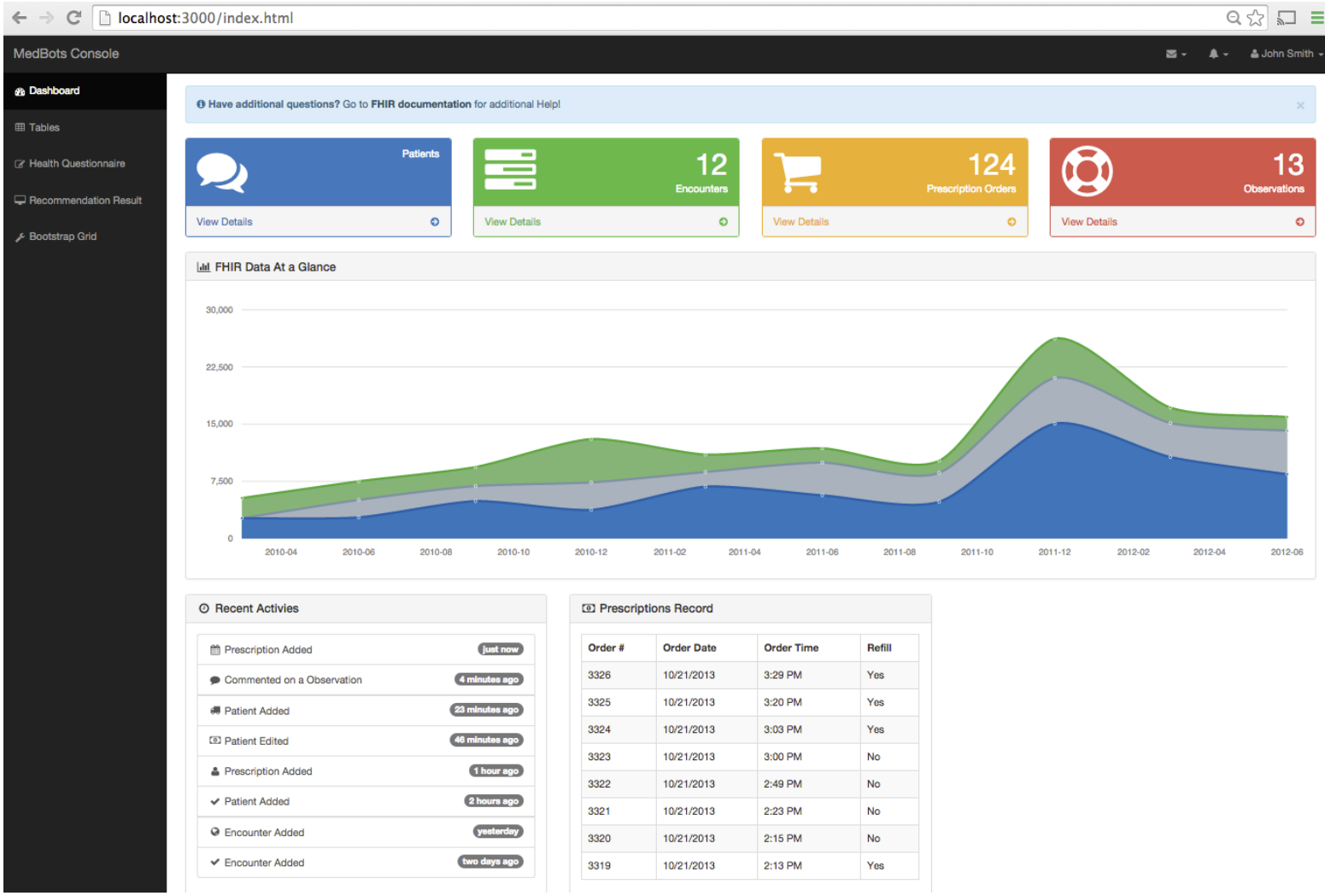
(4) I love Kibana —> **Kibana 4 on Elastic Search** supports all possible Charts including GeoJson and easy to build adhoc dashboards

(5) **Angular-Js Seed Dashboard**

- It provides out-of-the box charts and dashboards where one can feed his data points
-

I created following DRUGS Recommendation dashboard using the Angular-Js Seed Dashboard: https://github.com/masters-computer-science/medbots

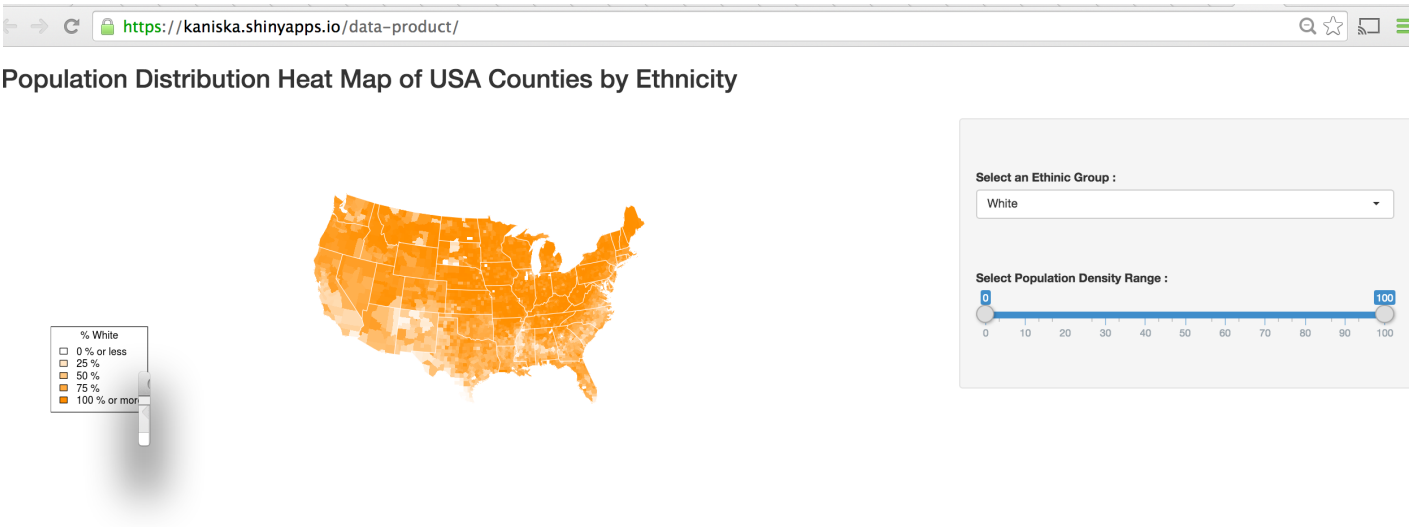
also created 'Comprehensive Product Dashboard' : https://github.com/kaniska/quickfire_product_dashboard — this is what I prefer to use for a quick POC / Demo



(5) Shinyapps - Analytics on R Data

https://github.com/masters-computer-science/data-product

https://kaniska.shinyapps.io/data-product/



— —

My quick analysis of Analytics Software (bit outdated) — **https://phisymmetry.wordpress.com/2013/02/04/approaches-for-fast-ad-hoc-query-on-current-and-historical-business-data/**

—-

Reflections on how some Key Technologies are leveraged by major Distributed Analytics Databases :

https://phisymmetry.wordpress.com/2013/02/11/welcome-to-new-age-of-analytics-renaissance/