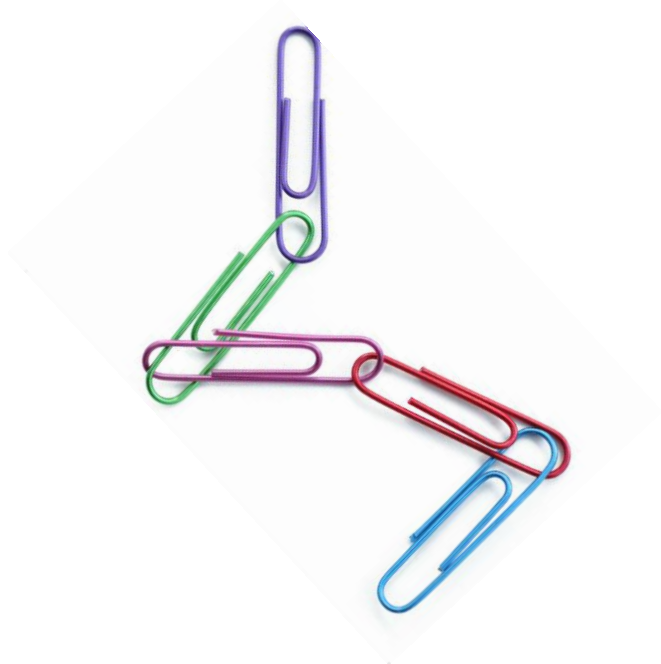
YACE



Yet Another Crossing Engine

Developed with Apache NiFi

3/16/2017

HTTPS://GITHUB.COM/AASHAR/NIFI4TRADING/TREE/V0.0.3/NIFI-FIX-BUNDLE

Niagara Files!!!

An Active Apache project

Flow-based programming

Web-based, visual data flow designer

Distributed processing with clustering

Excellent connectivity for information consumption, processing, and distribution

Extensible for custom streams processing applications development, and hosting

TDD

Traditional SDLC processes can be built with templates

Good operational functionalities with real-time flow monitoring and tracing

100% Java

QuickFIX/J

Leading open-source FIX “Engine”

Support customizations

100% Java, other implementations in C++, Dot.Net, etc. are available

Integrated with many open-source service platforms

YACE

Crossing

- Price / Time Priority
- Continuous crossing or scheduled crosses

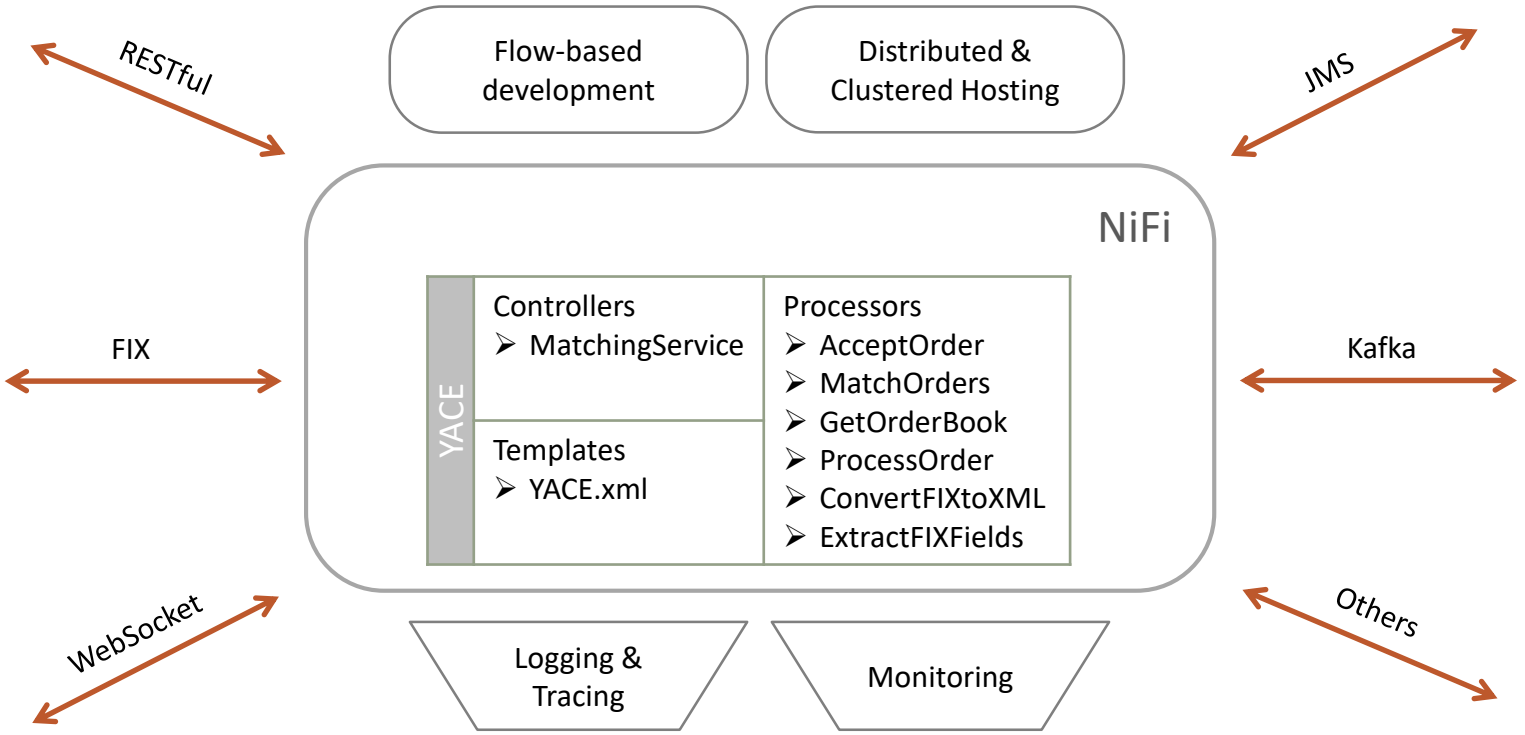
Based on QuickFIX example, ordermatch, with following changes:

- Run in an engine-free or a server-less mode
- Order placement uses binary search for better performance
- Support JSON serialization instead of FIX. FIX to JSON transformations can be handled by other components in NiFi.
- Other serializations can offer benefits like improved performance, payload versioning, etc.

Order book implemented as Java collections. The other options are –

- External caching / persistence solutions – preferred for availability and performance
- NiFi distributed cache
  - Good – NiFi implementation, support access across the cluster
  - Bad – Sizing limitation, single data structure

YACE Architecture



Demo

Show & Tell

&

Play

References

NiFi: http://nifi.apache.org/

QuickFIX/J: http://www.quickfixj.org/

GitHub: https://github.com/aashar/NiFi4Trading/tree/v0.0.3/nifi-fix-bundle

Blog: https://aasharsite.wordpress.com/

Q / A

Comments, Questions, Suggestions, Concerns, ...