IBM Software

High-Volume Performance Test Framework using Big Data

Mike Yesudas, Girish Menon, Satheesh Nair

IBM Corporation

*IBM Standard disclaimer applies

Disclaimer

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Background

- POC (Proof of Concept) to scale IBM-Sterling Order Management Application to peak hour volume
  - Largest IBM retail transaction volume tested
  - 7 million order lines per hour (2000 order lines per second)
  - Includes creating - Initial Load, Test data generation and Stubs

- Challenges
  - Limitations with commercial tools and cost
  - Cost of infrastructure and scripting
  - Tight timelines and limited resources
  - Dynamic, data-driven test scenarios
  - Achieve sub-second response times

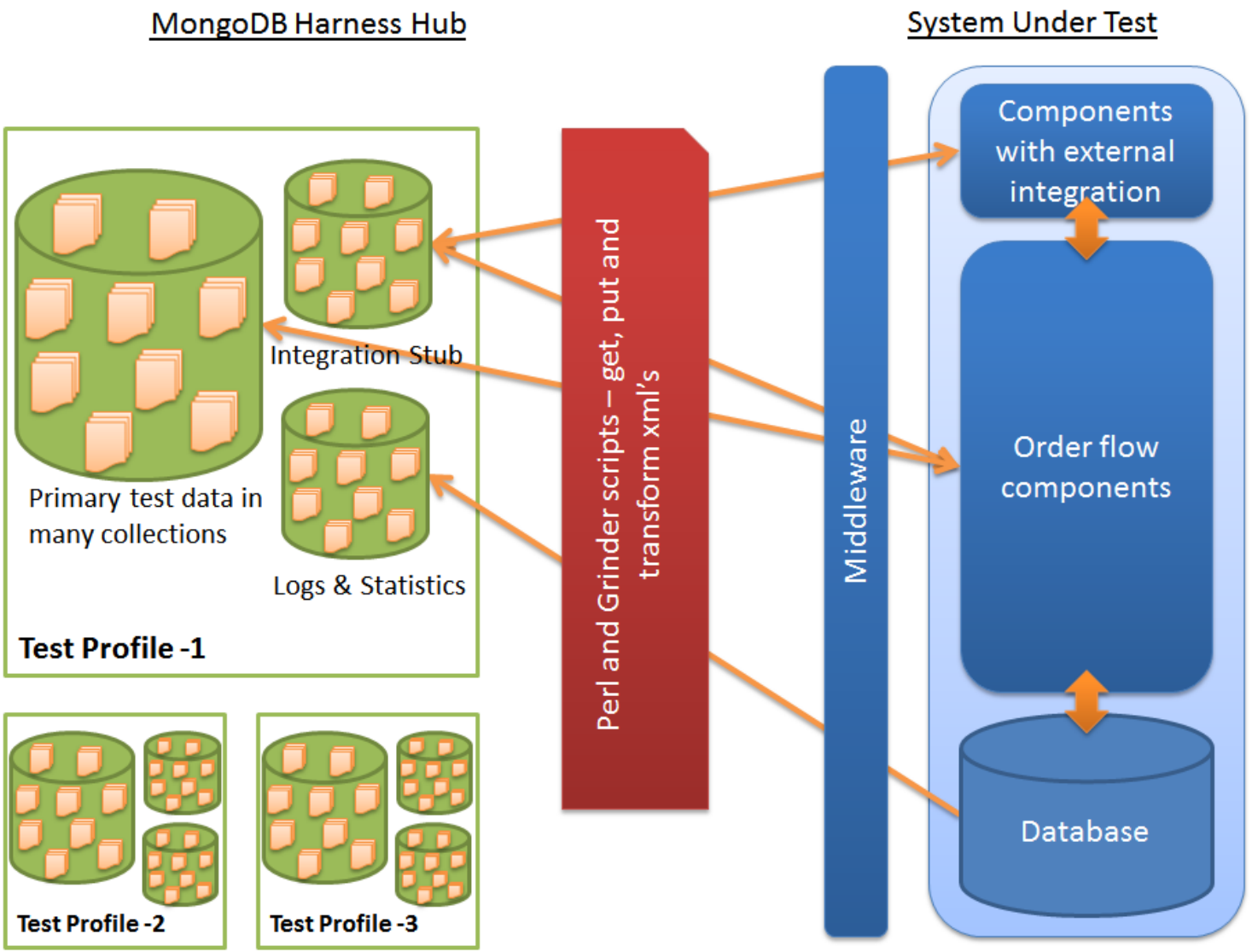
- Test approach
  - Avoid complex scripting logic for interfaces and stubs
  - Shift from code-based to content-based testing – Big Data
  - Team used MongoDB as the test harness database
  - Perl, Python and Java were used for scripting/coding

MongoDB and the Framework

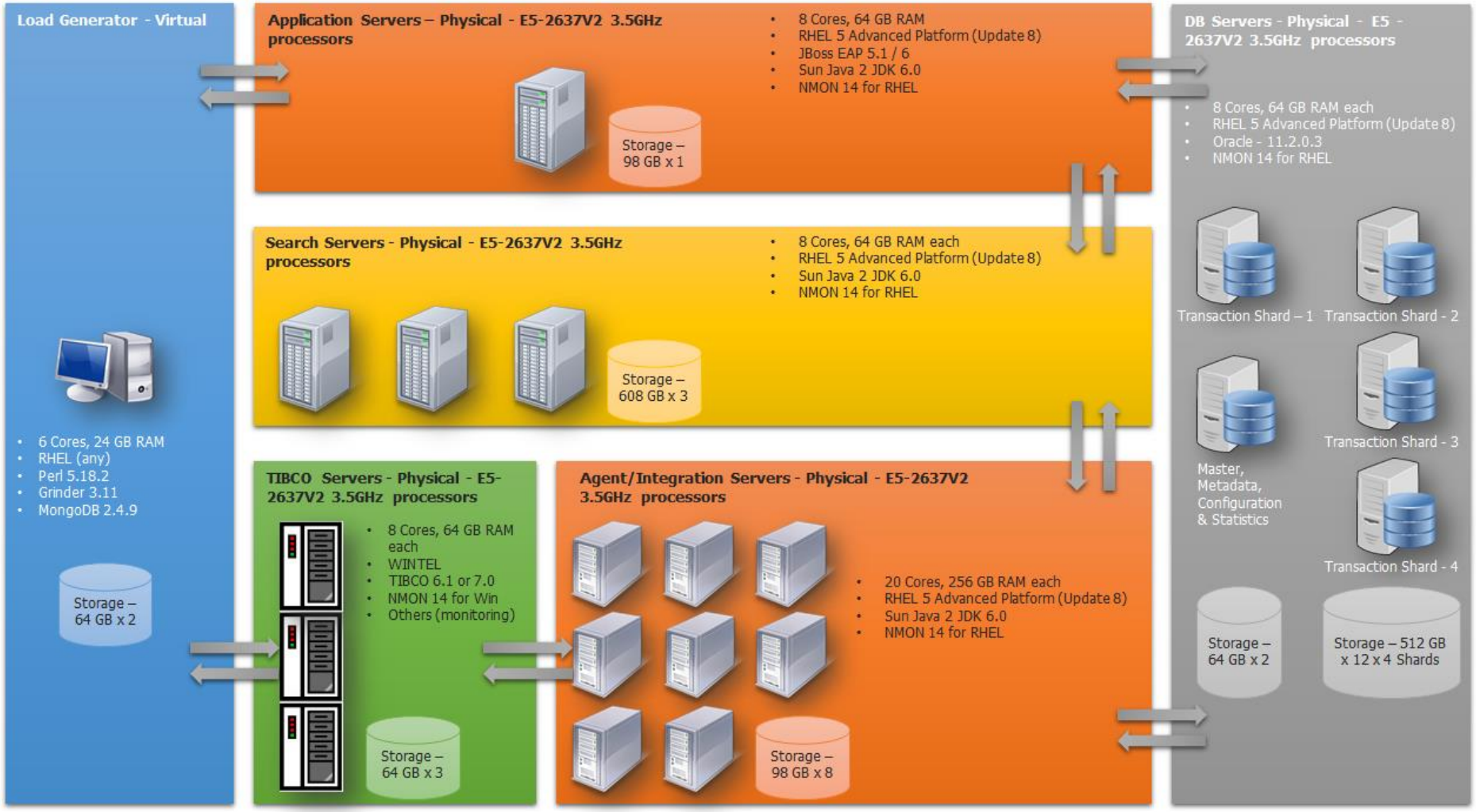
- MongoDB is an open-source and a leading NoSQL database
  - Document Oriented Storage and document based-querying
  - Map/Reduce for aggregation and data processing
  - Quick install, setup and faster scripting
  - Supports real-time insert/update of documents

- Framework
  - IBM's strategy of reuse of process, methodology and technical assets
  - Framework makes it easy to adopt and adapt
  - Enables TaaS (Testing as a Service) – deploy on cloud

Test Planning & Execution



Test Planning & Execution



Test Results & Lessons Learned

- **Store and process large XMLs** - enables generation and manipulation of input/output xmls
- **Performance test logic** is built in data , rather than in scripts
- **Quick scripting** in any language (Perl, Java, JavaScript, etc.)
- **Real-time control over stubs** (response data, response time and format) – for e.g. substitutions when inventory runs out during a floor kill - at random and pre-defined
- **Predict** test profile, **validate** accuracy of tests and manipulate tests **during runtime**
- Provide **real-time updates** of response times, errors etc. using standard big data analytical tools – output xmls stored and verified for transaction process verification
- Enables **more controlled performance tests** – detect errors, re-runs with minimal time/effort
- Enables **data reuse** in other projects (with minimal updates)
- **Storing performance outputs** like – log files, heap and thread dumps, statistics details, etc. and analyze side-by-side with test output data

Questions?

7 million order lines far exceeds the peak e-commerce traffic volume of most retailers in the world (with a few exceptions in China). Given that we have not encountered any bottlenecks with our test framework, do you think our strategy can be adopted for high volume transactions in Finance, Insurance or Telecom.

We think that every product vendor should supply the process of testing as a service (TaaS). That will be a necessity in the coming days for all enterprise products.