



Fast and Reliable Legacy Systems

Luiz Fernando Teston

FLOW ■ TRADERS

About Luiz Fernando

Lead Java Dev @ Flow Traders

20+ years writing code for living

Many domains, countries and projects



About Flow Traders

Market Maker specialized in
ETPs across major exchanges
globally

FLOW ■ TRADERS

Agenda

- Where to start
- Observe
- Facilitating changes
- Common bottlenecks
- Teaching others



Deciding where to start



Observe: people and tools



70 GB

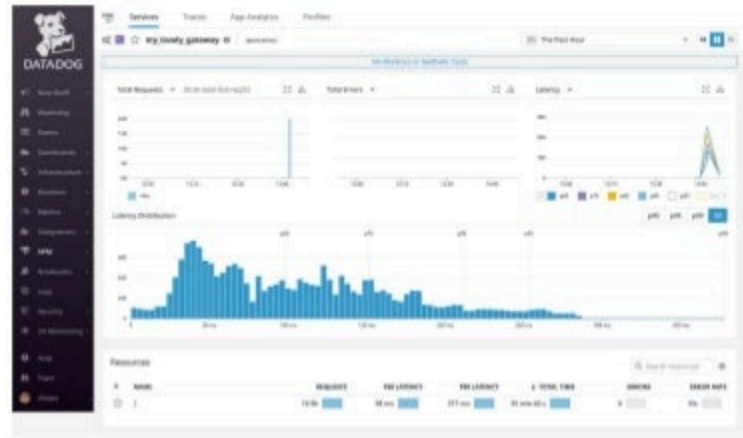
83 GB

94 GB

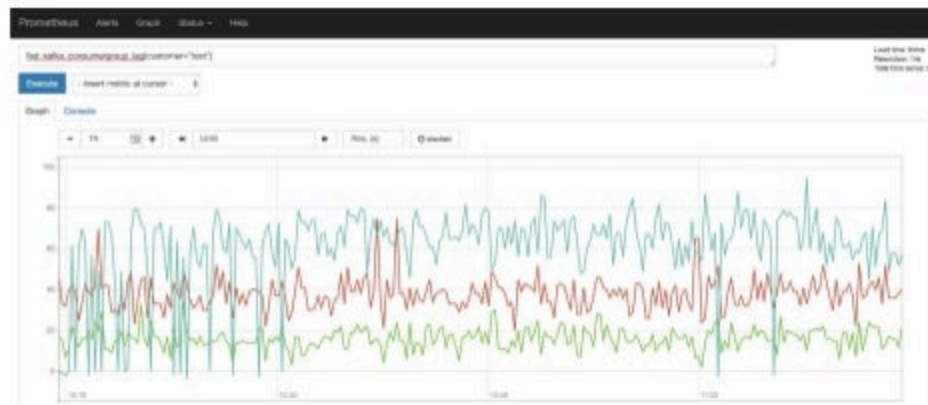
73 GB

81 GB

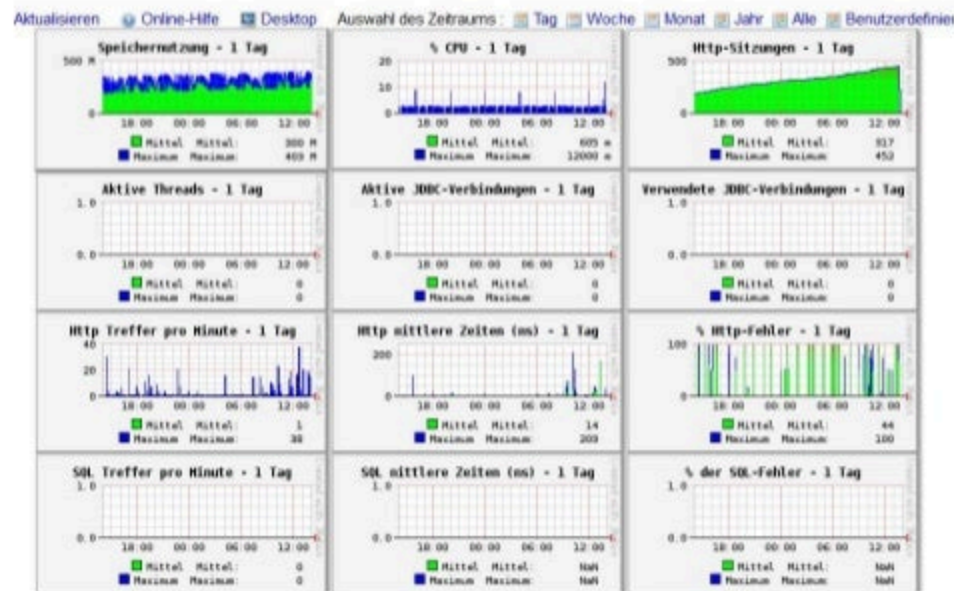
APMs



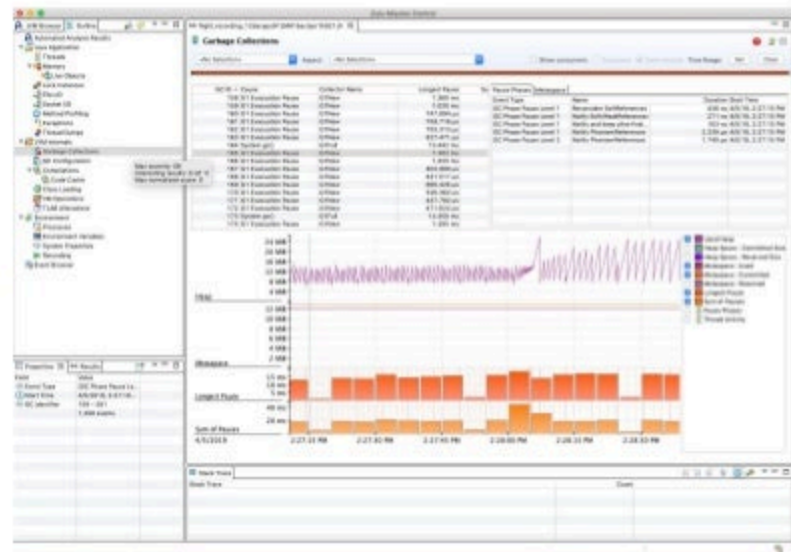
Prometheus & Grafana



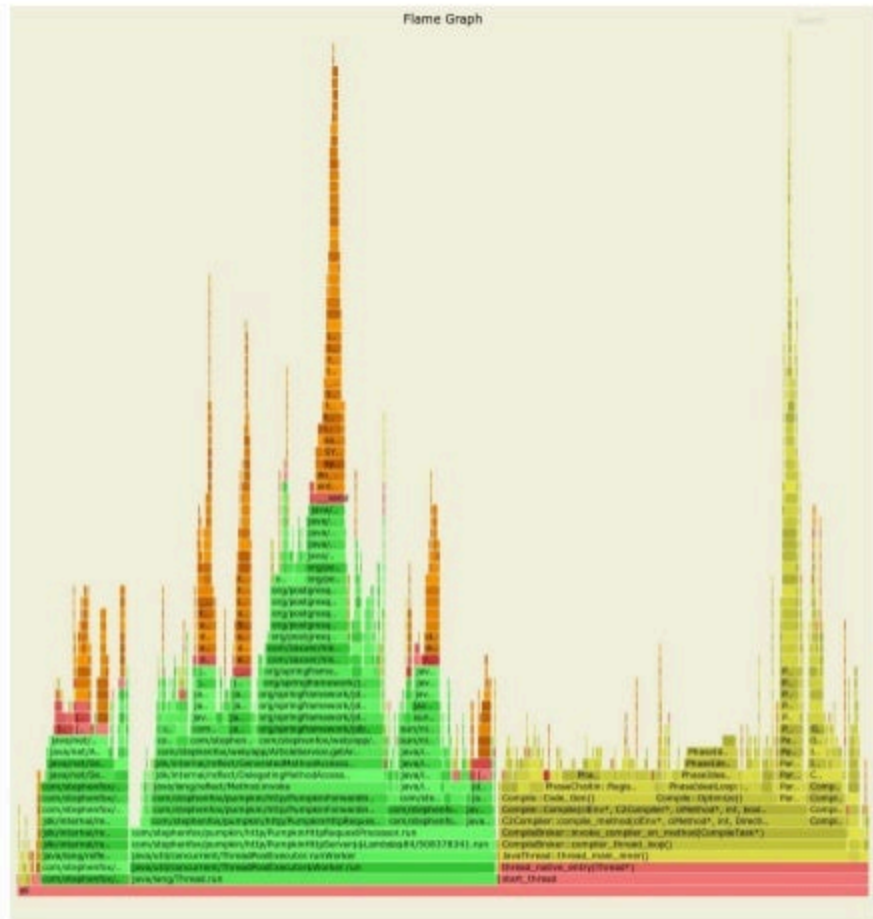
javamelody



Flight Recorder & Mission Control



Flamegraphs



4 golden rules

- Latency
- Saturation
- Errors
- Trafic



Facilitating changes



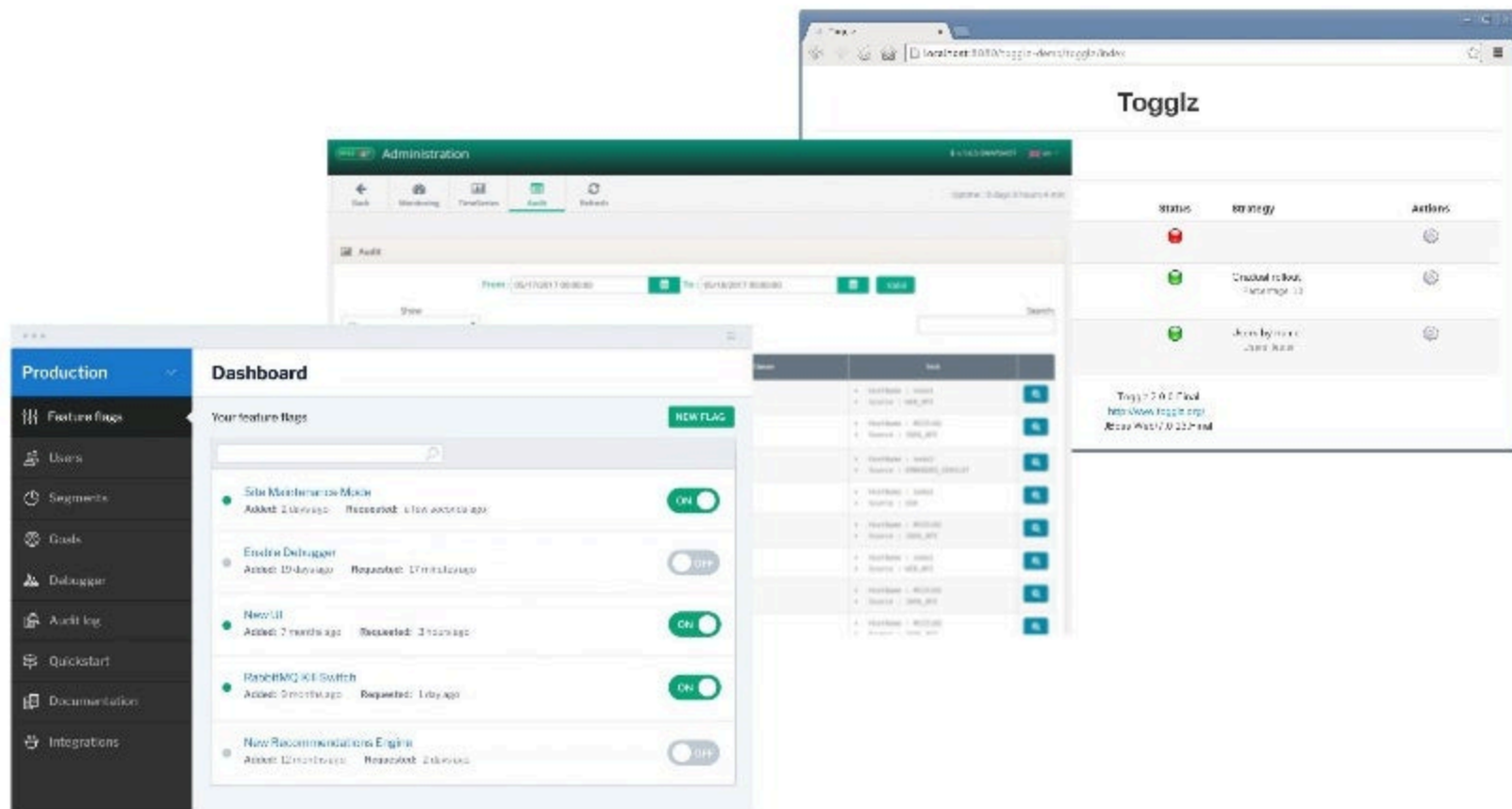
Smaller and frequent deploys



Trunk based development



Feature toggles



Blue green





Common scenarios



Database

addBatch

```
void addBatch(String sql)  
    throws SQLException
```

Adds the given SQL command to the current list of commands for this Statement object. The commands in this list can be executed as a batch by calling the method `executeBatch`.

Note: This method cannot be called on a `PreparedStatement` or `CallableStatement`.

Parameters:

`sql` - typically this is a SQL INSERT or UPDATE statement

Throws:

`SQLException` - if a database access error occurs, this method is called on a closed Statement, the driver does not support batch updates, the method is called on a `PreparedStatement` or `CallableStatement`

Since:

1.2

See Also:

`executeBatch()`, `DatabaseMetaData.supportsBatchUpdates()`



Multi threading

The `Executors` class provides factory methods for the executor services provided in this package.

Usage Examples

Here is a sketch of a network service in which threads in a thread pool service incoming requests. It uses the preconfigured `Executors.newFixedThreadPool(int)` factory method:

```
class NetworkService implements Runnable {
    private final ServerSocket serverSocket;
    private final ExecutorService pool;

    public NetworkService(int port, int poolSize)
        throws IOException {
        serverSocket = new ServerSocket(port);
        pool = Executors.newFixedThreadPool(poolSize);
    }

    public void run() { // run the service
        try {
            for (;;) {
                pool.execute(new Handler(serverSocket.accept()));
            }
        } catch (IOException ex) {
            pool.shutdown();
        }
    }
}

class Handler implements Runnable {
    private final Socket socket;
    Handler(Socket socket) { this.socket = socket; }
    public void run() {
        // read and service request on socket
    }
}
```



Fine tune with
saturation in mind



Teaching others

A photograph of several fishermen on a sandy beach at sunset. They are pulling a large green fishing net from the ocean. A blue boat is on the left. The sky is orange and yellow from the setting sun.

Recap

- Observability tools to decide where to start
- Facilitate incremental deploys
- Feature toggles, blue green to give resiliency
- Teach by example



Questions?

References

- <https://sre.google/books/>
- <https://developers.redhat.com/blog/2020/08/25/get-started-with-jdk-flight-recorder-in-openjdk-8u#>
- <https://prometheus.io/>
- <https://grafana.com/>
- <https://github.com/javamelody/javamelody>
- <https://trunkbaseddevelopment.com/>



References