DevOps with MATLAB: A Predictive Maintenance System for Streaming Data

Christine Bolliger, PhD
She/Her

Nicole Bonfatti
She/Her

Seth DeLand
He/Him
Key Takeaways

- **Incorporate familiar MATLAB capabilities**, including Predictive Maintenance and Drift Detection, in operations

- **Integrate with production systems** like data sources and dashboards, and translate those integrations from desktop to cloud servers

- **Automatically build, test, package, and deploy MATLAB code and Simulink models** with CI/CD
DevOps: Develop and Operate Production Software

Development

Operations

Predictive Maintenance

Financial Modeling

Embedded Controls
Not necessarily a conflict of interests, but certainly different interests

- Design
- New Features
- Performance
- Testing

- Robustness
- Reproducibility
- Scalability
- Monitoring
Example: Predicting Battery State-of-Health

- Fleet of electric buses
- Maintenance is expensive. **Could we do a better job predicting when batteries need replacing?**
- Started gathering telemetry data on batteries
Create a SoH prediction function using domain-specific tools for engineering data and predictive maintenance
A scalable production system running on the cloud, using industry standard tools
A scalable production system running on the cloud, using industry standard tools

Deploy MATLAB Production Server in containers on cloud infrastructure
A scalable production system running on the cloud, using industry standard tools

Automate build, test, and deploy using source control and CI/CD
A scalable production system running on the cloud, using industry standard tools

Run analytics on live streaming data
A scalable production system running on the cloud, using industry standard tools

Expose valuable metrics to business stakeholders using industry-standard dashboard integrations
A scalable production system running on the cloud, using industry standard tools

Retrain models using observed data and easily update those in production
State of health algorithm in production

Production System
- Receive sensor data as kafka stream
- Load battery model from Redis cache
- Expose metrics with Prometheus
- Save data and predictions to database

Local testing
- Mock dependencies
Write SoH prediction function to use kafka streams

Simulate streams with
- `inMemoryStream`
- `testStream`

Use production Kafka streams with `kafkaStream`

Debug locally, then deploy the same MATLAB code to production.
Automatically build, test, package, and deploy MATLAB code

```matlab
function plan = buildfile
plan = buildplan(localfunctions);
plan("packageDriftDetection").Dependencies = "test";
plan("packageSoHPrediction").Dependencies = "test";
plan("test").Dependencies = "validate";
end
```

- name: Run MATLAB buildtool
  uses: matlab-actions/run-build@v1
  with:
  tasks: packageDriftDetection packageSoHPrediction
The static data assumption rarely holds in the real world.
Developing drift detection with `detectdrift`

- Use historical data (training data) to create a baseline distribution
- Generate synthetic data to test for drift
  - This will be replaced by streaming data in the production system
Update model when drift is detected

Data labeling

Retrain

Drift Detection

\[ f_2(x) \]

Retrain model when drift is detected
Update infrastructure to periodically run the drift detection function
The Complete System
Key Takeaways

- **Incorporate familiar MATLAB capabilities**, including Predictive Maintenance and Drift Detection, in operations

- **Integrate with production systems** like data sources and dashboards, and translate those integrations from desktop to cloud servers

- **Automatically build, test, package, and deploy MATLAB code and Simulink models** with CI/CD
Learn More

How MATLAB and Simulink are used with Enterprise IT
https://www.mathworks.com/solutions/enterprise-it-systems.html

CI/CD Resources

MATLAB and Simulink in the Cloud
https://www.mathworks.com/solutions/cloud.html

Automating Machine Learning with DevOps for MATLAB and Simulink
Questions?
Attributions

- Apache, Apache Kafka, Kafka and the Kafka logo are trademarks of the Apache Software Foundation. The Apache Software Foundation has no affiliation with and does not endorse the materials provided at this event.

- The Grafana Labs Marks are trademarks of Grafana Labs, and are used with Grafana Labs’ permission. We are not affiliated with, endorsed or sponsored by Grafana Labs or its affiliates.

- Microsoft, Azure, Azure Kubernetes Service, GitHub, GitHub Actions, and their associated logos are trademarks of the Microsoft group of companies.

- Prometheus, Kubernetes, and their associated logos are registered trademarks of The Linux Foundation.

- Redis is a registered trademark of Redis Ltd. Any rights therein are reserved to Redis Ltd. Any use by MathWorks is for referential purposes only and does not indicate any sponsorship, endorsement or affiliation between Redis and MathWorks.