

Performance (Testing + Engineering) ^ (Programming + Automation)Built E2E CmdLine tools To Automate Testing, Engineering, Monitoring, Scaling easy, friendly For Prod,NonProd

Languages.: **Golang**, Shell, AWK
 DB.....: **Postgres**, Oracle AWR
 Cloud.....: **GCP** certified,Kubernetes(K8S), OpenShift
 Load tool....: **K6** (JavaScript)

OpenSource....: Inclined
 APM Tools.....: Dynatrace
 CI/CD centric...: Yes
 MiddleWare....: IIB,MQ,WAS**

What I**Am:**

1. Take E2E** Ownership of tasks, Finish. Automate stuff. Explore App's Tech-Ecosystem
2. Increase Customer Satisfaction using Automation as a Catalyst.
3. Absorb Complexity, Present things in a Simple way to Customer, Management, Technical Teams.
4. Believe in Automation + CommonSense = Simple, Robust, Fast, Live Performance Tuning.
5. Reduce reliance on expensive tools., Instead built Reliable CLI tools for E2E
6. Continually Seek, Learn & Apply promising OSS** tools/programming. Ex: Golang, K6, Postgres
7. Enjoy cross functional working with teams, technologies, know app better to build Right automation
8. Team Player, Self learner , Share learning with team. Encourage, Coach, Train team

Did:

1. Built Right, Reliable, futuristic tools for test data preparation, triggering, monitoring, analysis phases in Performance Sphere - Cloud or OnPrem, where tools couldn't scale or fail with large volumes
2. Automated boiler plate. Invest time saved on App Performance Tuning or Self learn. Repeat.
3. Developed CommandLine(CLI) utilities** :
 - a. Golang: Scaling solution. ~500 Million transaction logs, 80 GB+ Size(Counts, ResponseTimes)
 - b. Simple: Consume complexity of distributed app layout and output stats simple way
 - c. Prod : RCA is easy. It's not T+15 min but LIVE across 50+ or more servers.
 - d. QA: Perf engineering/analysis is Easy, LIVE, CI/CD, Dev friendly, stats Mailed automatically.
 - e. Programmed features seen in Commercial Performance tools to give on par or better insight

Look for:

1. Golang: Scalable Tech built are Promising & OSS. Use them in Perf domain for Web,API,Mobile.
2. Challenges at work. They help me grow, become reinforced

Note : I simplified Resume below. Kept it short. if your time permits, read further, or please call me

Duration	Company	Project	Role
07/2010 - Till Date	Tata Consultancy Services, Chennai	DTCC	Performance Engineer + Automation
06/2007 - 06/2010	Mahindra Satyam, Chennai	NYSE	WebSphere Admin

Project 3: 09/2021 - Till date: DTCC: FIX Market Volatility Performance Testing

Gamestop's issue of 1/28/2021 led to DTCC's FIX TMV Team. 220 Million trade as against 100 Million regular, on that day led to FIX Engine's Out of memory.

What I do :

1. Round trip time calculation during busy hours, deal with large volume of raw logs
2. Using shell scripts, Prod logs will be transformed to recyclable data (130 - 200 Million)
3. Auto trigger, these 2x volume and analyze FIX engine behavior
4. Watch LIVE App,Host KPI metrics and logs,with no Tools and no Performance impact either
5. Automatically extract post test counts, all these reduce hours of labor , human dependency

What I learnt :

1. Simplified logs using shell, awk. but it couldn't scale to analyze transactions, so applied golang
2. Go concurrency model using channels and analyze 350K transactions in ~ 3 hrs time
3. Advanced Shell, AWK Programming to deal with huge log volume

** WAS: WebSphere Application Server

** OSS: Open Source Software

** E2E: End To End

** GUI is slow, fails often, unreliable, resource intense, Anti CI/CD

Project 2: 07/2010 - 06/2021: DTCC,USA: GTR Performance Testing

Global Recession led to GTR – Global Trade Repository , a regulatory mandate to get transparency in to Derivatives market across Globe (Prime Markets – US, Europe, Singapore, Hong Kong, Japan, Australia), where multiple market players ranging from Regulators to various trade parties use on a daily basis.

What I did :

1. Using Bash Programming, each day 3-4 performance tests with 2x volume test across US,EU,APAC etc
2. Prepare data and trigger tests with just a click using Bash Programming
3. Patrol script auto detects Test Start and End times.
4. LIVE progress of test - any runtime space issues with linux disk, oracle tablespace, error logs
5. Hung threads, Out of memory exceptions can be seen LIVE , as they happen
6. LIVE monitor of IIB, MQ, WAS, Oracle tiers, host KPI metrics .
7. Any number of tests , done in any hour, notification mail will be seen once done at no cost
8. Time to test preparation, execution, analysis and Recommendation is not after the fact, but LIVE
9. LIVE Monitor can be used in Prod env with no Performance impact

What I learnt :

1. Above steps seem oversimplified, thanks to Bash Programming learnt
2. If not for it, it takes us days to identify issues and finish tests
3. I started small, later moved to advanced Shell and AWK Programming too
4. Patrol script i designed reduces need for expensive tools like Dynatrace , Splunk
5. Dynatrace : built business transactions, dashboards for monitoring earlier to shell scripts
6. Dynatrace : Out of Memory identification of root cause done
7. Dynatrace : MQ Queue depth, Listener dashboard creation
8. AWR Assessment. it holds some decent info

Project 1: 12/2007 - 05/2010: NYSE,USA: WebAdmins Team

NYSE is a New York City based stock exchange. It is the largest stock exchange in the world in dollar volume. The webadmins team of the NYSE supports its entire Web systems and Operations infrastructure. NYSE has a host of applications for its intranet users/employees as well as customers. These applications are installed, integrated & running under various platforms/ databases/ Application servers. The project basically supports the QA and Production Application Environments.

What I did :

1. Silent Installation for WAS and Webserver installations
2. Manual generation of Heap and Javacore dumps using wsadmin(jython) and console
3. Profile Creation [Deployment Manager/Cell/Application Server/Custom]
4. Performance monitoring/Tuning of AppServers and advice on optimization measures using PMI tool.
5. Configure JDBC resources like database drivers, data sources verify the new DS
6. Troubleshoot application and Application Server errors/exceptions during startups by thorough examination of logs, Synchronization issues and resolving the same across the node

What I learnt :

1. Basics of Linux, middleware and distributed Environments
2. Heap and thread analyzer tools usage to analyze Heap and thread dumps

Education:

2007 : B.Tech : 78 %

2003 : XII : 90.2 %

2001 : X : 89.8 %

Declaration: I hereby declare that above furnished information is genuine to the best of my knowledge.