Hemshankar Sahu

M. Tech
IIT Roorkee (CSE)

+91-72591-94386
hemshankar.sahu@gmail.com

PUBLIC PROFILES

- **Databricks Spark Summit Speaker** : https://databricks.com/speaker/hemshankar-sahu
- **Github**: https://github.com/hemshankar?tab=repositories
- **Linked-in**:https://www.linkedin.com/in/hemshankar-sahu-765abb28
- **Kaggle** : https://www.kaggle.com/ninjacoders

Major responsibilities

1. Research on new technologies to bring them in Product.
2. Design/Architect and develop highly available, low latency, cloud native SAAS applications
    a. Design HLD/LLD
    b. API Specification
    c. State Diagrams
    d. Sequence Diagrams
    e. SLA definitions
3. Create POCs (Proof of concepts)
4. Develop/Code/Deploy products
5. Review and debug code.
6. Write, suggest internal tools to increase productivity
7. Write Unit/Integration/Performance Tests
8. Assist QA to make Test Specification
9. Attend meetings with customers to resolve critical issues.
10. Represent my organization at various events
11. Assist peers and juniors to understand Tech Stack, Project, features, design, code reviews etc.
12. Backend development using technologies:
    a. Go, Java, Python,
    b. Kubernetes, Monitoring and Alerting, Data Pipeline, Data Warehouse, SQLs/No-SQLs
        i. Airflow, Spark
    c. Build Automation, CI-CD, MLOps,
    d. REST, GRPC, protobuf, swaggers
13. Cloud: GCP and AWS ecosystem

WORK EXPERIENCE

Staff Software Engineer at Suki

Backend Engineer

From 02/Nov/2022 till present.

Projects

Following are different projects that I worked creating a **voice based digital assistant**

- **Suki Speech Platform (SSP)**
  - Working on Suki Speech Platform to scale processing of speech to text using different ASR (Automatic Speech Recognition) for internal as well as external customers.
  - Breaking the monolith into different microservices
  - Design changes to make system 99.95 available
  - Design changes to make system more reliable and recoverable.
  - **ASR Manager**
    - Design changes to make pluggable and extendable system.
    - To support Google ASR, NVOQ ASR and Suki ASR.
    - Implement Disaster Recovery to handle any reconnects, pod restarts.
- **Suki Server**
  - A common server to coordinate between different feature microservices
  - Handle intent identifications
  - Connect to SSP platform for voice processing
  - Handle digital note related requests.

- **Suki Automatic Speech Recognition (ASR)**
  - Deploy tuned Whisper model to Kubernetes cluster
  - Expose GRPC streaming apis for sending audios and receiving transcripts
  - Manage autoscaling and load balancing.
  - Monitor latencies
  - Deploy different versions based on different requirements
    - Small, Medium and Large
  - Support for multilingual models
- **LLM Ops**
  - Design and Implement abstraction over LLMs to be used by different internal components.
  - Highly scalable and resilient with optimized latency
- **Suki Data Platform**
  - Desing and implement Internal platform to manage platform data.
  - Implement ETL and ELT like pipelines using airflow
  - Use DBT or spark based on use cases
- **ML-Ops**
  - Use Kubeflow to create pipelines for training inhouse models

        - ASR model based on OpenAI's whisper using pytorch
    - Provide a generic framework to train and evaluate the models.
    - Implement CI-CD flows to deploy models on kserve.
- **Feature Store Manager**
    - A microservice to maintain set of features for different entities with following features
        - Read Optimized for high volume data using cache and right database
        - Custom Feature resolution based of the feature requirements
        - Stores user features like prompts, ASR hints, doctor specialties etc.
- **Cloud cost optimization**
    - Explore different ways to optimize the cost
    - Dynamic scaling down the pods
    - Reducing ASR usage
    - Optimizing Bucket and log usage

Tech Stack: GO-lang, Python, Redis Streams, Kubernetes, GCP, gRPC, Protobuf, PubSub, kubeflow, Kserve, air-flow, spark

Principal Software Engineer at Informatica

Backend Engineer

From 23/March/2015 – 30/October/2022

Projects

- **Spark Execution Engine for Big Data Management**
    - Job Compilation: From Informatica Job to Spark Job
        - Compiling and converting Informatica Job XML to equivalent Spark Job.
        - Worked on a pluggable framework to add new transforming capability for generating Spark Job (Flex Framework, Informatica Patented approach)
    - Workflow generation
        - Aggregate all the tasks required to execute a Job and form a DAG out of it.
        - Later Extending it for Asynchronous Submission
    - Monitoring
        - Catching, extracting, and relaying back the sparks events
        - Persisting selected data in database
    - Security
        - Extending Spark code to achieve desirable security standards
            - Use/Enable **Kerberos**
                - Make sure correct set of delegation tokens are used/provided
            - Enable LDAP users
    - Logging framework
        - Implement Spark Listener to listen for various spark events
        - Log as required.
    - Used **Java and Scala and Spark**

- **CLAIRE Metadata Repository (CMDR)**
  - Introduce a new metadata repository as a SAAS to aggregate customers metadata from different Informatica product, which can be used by Informatica's Data Scientist and ML Engineers.
  - Aims at ensuring
    - Consumer's data is being stored, accessed, and used in accordance with Security, legal & compliance guidelines.
    - All data assets & information to enable AI & analytical use cases is available at a centralized place in a secure & governed ecosystem.
    - Teams get variety of infrastructure options, capabilities & tools to develop, tune and test solutions in seamless and integrated manner.

- **Informatica MLOps Platform**
  - Introduce a new service to facilitate Data Scientist to use Informatica Eco-system to add and deploy Trained ML model and expose a REST Endpoint.
  - Expose REST End Points to allow user to deploy/un-deploy/redeploy machine learning model.
  - Implement a python template code to be updated by user as per ML model
  - Make sure the required software is installed in the Kubernetes cluster
    - Kserve, Istio, Knative, Cert Manager
  - Implement periodic status update of deploy model.
  - Implement Log collection for deployed models
    - Using fluentbit.
  - Provide resource (CPU, RAM) consumption for each deployment for metering
  - Write Custom ML Model, dockerfile for creating docker image, yaml file for deployment

- **Machine Learning Transformation for CDI-e (Cloud Data Integration-Elastic)**
  - Research Various ML-Ops platform
    - Azure ML, SageMaker, Kubeflow, Databricks ML-Flow
  - ML Models deployed in various ML-Ops platform can be consumed directly form informatica mapping.
  - Improve performance by batching of records.
  - Tech Used: Spark, SageMaker Azure ML, Machine Learning Models, Ok-http REST Client
  - Other Responsibilities
    - Help QA to create and deploy ML Models
    - Write Doc/Wikis/Demos on How to use ML Tx.
    - Help DOC team.

- **Advanced Logging in Kubernetes cluster**
  - Explore various logging options
    - Fluent-bit, Vector, log-stash
  - Collect detailed logs from various containers/pods
  - Stream the logs to S3/Azure Blob

- **Informatica Solutions Repository (Patent Filed)**
  - A platform to store compute solutions, so that it can be used by multiple products
  - Solution should be executable at Local System, Kubernetes, Hadoop, REST Endpoint
  - Code/plugin is generated for consumers
    - Spark, Java, Scala, REST, Informatica DEI/CDI-e
  - Other Responsibilities
    - Help QA with the test Cases
    - Write Doc/Wikis/Demos on How to use ISR
    - Help DOC team.
  - Presented this idea in spark Data+AI summit
    - https://databricks.com/session_eu20/simplifying-ai-integration-on-apache-spark

- **CLAIRE Plugin for Jupyter Notebook (POC)**
  - Provide Intelligent recommendation as per customer data processing requirements.
  - A Jupyter notebook plugin for bringing Informatica Services to Data Science Eco System.
  - Bring Data Sources registered from Informatica Eco system to Notebook
  - Take ML Models developed in Notebook to Informatica Eco System.

- **Blaze application master** (now Deprecated)
  - Design and implement informatica blaze application master to run informatica mapping in Hadoop ecosystem.
  - Workflow Generation
    - Aggregate all the tasks required to execute a Job and form a DAG out of it.
  - Monitoring
    - Used hadoop timeline server
- Extend **partitioning framework**
  - For Informatica's for HBase and Complex File adapter
- **Attend meeting to solve customer issues**
  - Discuss and understand the problem
  - Provide instrumentation to find the core issue.
  - Find the fix and forward and backport the changes
- Provide tech sessions to juniors, other teams, and QA about hadoop technologies, spark engine and other internal projects.

Software Development Engineer at Enlightiks (now Practo)

Full Stack Engineer

From Oct/06/2014-Mar/20/2014.

- Information extraction from unstructured data using **JSOUP** and **open NLP** in medical domain.
- Develop **text classification** module that can be **trained** (auto/manually) for different set of classes using **Naive Bayesian network**.

    - Using **Concept Net, rest APIs,** Java, **JSON**,
- Implement various rules using **Drools** and expose them as **web services.**
- Generate complex SQL dynamically according to the User input.

Member of Technical Staff at Oracle

Full Stack Engineer

From Aug/8/2011 to Oct/03/2013

- **Oracle Enterprise Manager plug-in for managing and monitoring the Schema Objects**
    - Write **Abstract classes and Interfaces** for implementing various schema objects.
    - Write **multi threaded approach** to search keywords for file authentication
    - Implement Provision for Create, Edit, Delete and View Schema objects.
- **Implement** new feature to create a 'Sparse Disk Grop' on ASM (Automatic Storage Manager)
    - Extend the Disk group backend bean to add provision for a new Sparse type.
    - Check for various **backend validation** on the user input like correct failure group, correct version of DB and ASM and take corrective action.
    - Implement Backend methods to **Generate SQL** for creating the disk group based on the user selections and inputs (Sparse or Non-Sparse).
- **Design and develop** portal for managing "Automatic Filter Driver" (AFD).
    - Creating **backend beans and controllers** to allow user to select between different disks for provisioning and un-provisioning under AFD.
    - Show which disks are provisioned and un-provisioned under AFD in a tabular form.
    - My responsibilities were to develop back-end using **JAVA and JDBC** as well as front-end using ADF.

- Write unit test cases in **Selenium** for the new code.

TECHNICAL SKILL AND EXPERTISE

- **Cloud Administration and Architect**
    - AWS
    - Azure
    - GCP
- **Kubernetes Ecosystem**
    - Creating K8s Cluster
    - Deploying Application
    - Monitoring/logging
        - Fluent-bit
    - KServe

- **Hadoop Ecosystem**
    - **Yarn**: Creating application master

- **HDFS**: Writing HDFS adapters
- **Hive**: Writing Hive Storage handler as UDFs, and Serde
- **Security:** Kerberos, SSL, SASL, KMS (data encryption at rest), LDAP for Informatica

- **Spark**
  - RDD, Data-frames and Datasets
  - Pipelining and Narrow/Wide dependencies
  - RDD persistence
  - Spark as an application master
  - Spark Security
  - Writing custom RDDs
  - Writing UDFs
  - Optimizing Spark Processing
- **Java**
  - Asynchronous Job submission
  - Lambda functions
  - Identifying memory leak, file system leak and thread leak
  - Identifying concurrency bottleneck.
  - Worked on following design pattern
    - Factory, Singleton, Builder, Adapter, Decorator, Facade, Iterator, Observer, Strategy
  - Multithreading and parallel execution
  - Collections and Generics
  - Streams
  - Classloaders and Reflection
  - JNI and Reverse JNI

- **Scala**
  - Knowledge of following concepts
    - Higher Order Functions
    - Data and Abstraction
      - Classes, Traits
      - Companion objects
    - Types and Pattern Matching
    - Case classes and matches
    - Collections
    - Actor Model, Akka

- **C++**
  - JNI and reverse JNI
  - Sharing instance of JVM between different processes
  - Having a new in process thread in JVM for executing the CPP code

- **Build tools**
  - Maven
  - Gradle
  - SBT
  - **Go mod**

- **Machine Learning**
  - Used Stanford's NLP libraries to form a text classification module
  - Knowledge/Understanding about following algorithm/tools
    - Neural Networks
    - Deep Neural Networks
    - Clustering Algorithms
    - Bayesian Networks
    - RNNs
    - TensorFlow
    - Weka
    - SVM
    - Linear Regression
    - Image Processing
      - Using OpenCV and python libraries
    - CNNs
- **Python**
  - Just basics to use spark and openCV libraries
  - Used Flask for web service deployment
  - Created TCP server/client
  - Multithreading

- **Jupyter Notebook**
  - Created plugins for Jupyter Notebook

- **UI**
  - Javascript
  - JSP/JSF
  - HTML5
  - Play Scala Framework

- **Other technologies**
  - Deep understanding of ML-ops / Data Science Ecosystem.
    - Kubeflow, Azure ML, SageMaker

  - Web services
    - Implementing Rest Client and Server
    - Using JBoss libraries
    - Play Scala
    - Tomcat
    - Spring Boot

  - Graph Database
    - Just hands on Neo4J for personal use.

ACADEMIC PROJECTS

Slot Reservation Based Novel Base Station Scheduling Algorithm for Bandwidth Optimization in IEEE 802.16 (WiMAX) Networks.

The project aims at implementing a novel approach for the optimized bandwidth usage in IEEE 802.16 (WiMAX) standards. A novel approach is being developed and implemented for the Base Stations of the WiMAX.

As a Part of: Dissertation Master of Technology (at IIT Roorkee)
Duration: From June 2011 to June 2011 (1 year)
Language: C++
OS Used: Ubuntu – 9.10 (Linux)
Software: Ns-3 (Network Simulator- 3)

Ant Colony Simulation

The objective is to simulate an "ant colony" which is population based, general search technique for the solution of difficult combinatorial problems. The project is inspired by the pheromone trail laying behavior of real ant colonies.

As a Part of: Major Project Bachelor of Engineering
Duration: Jan 2005 to May 2005 (5 months)
Language: VB.net
Role: Designer and Developer

Application (PIPEBOOK) Query Builder

The project work is to develop a web page which can display the content of tables or views of database, by generating a query line as per the employee's requirement along with capability of saving the results to a excel file.

As a Part of: Internship at Reliance Info solutions Limited, Mumbai
Duration: 15/05/2004 to 30/6/2004 (45 days)
Language: C#.net
Role: Developer

Character Recognition System

The objective of the project is to develop an application which can recognize **human written characters**.

As a Part of : "AI" Course Project at Indian Institute of Technology, Roorkee
Duration: 3 Months (02/2010 to 5/2010)
Language: VB.net

PERSONAL PROJECTS

Workspace Manager (https://github.com/hemshankar/WorkspaceManager)

- Motivation Behind Workspace Manager
  - Tech keeps changing really fast, from native, to hadoop, to cloud, to microservices and so does the environment and workspace.
  - Generally, the setup required is achieved by initial developers working in the project using terminal, notepad, gedit, scripts etc.
  - These things can generally be not shared across the users/developers. Also, when the system crashes whole setup is gone, and needs to be recreated from scratch.

- Docs created for setup becomes outdated as the tech changes or updates, or with requirements, and if not updated properly becomes stale.

Animated Arenas (http://animatedarena.com/) (private repository)

- A webapp to see the Data-structures and algorithms in animated form
- Easy APIs allow users to add their program and the app will convert that to animation
- Features like captcha, Google and FB login, comment, and rating
- Interview question and answers panel
- Used: JSF, Tomcat, **Java-Script**, **HTML5** features (Canvas), Amazon EC2, MySql
- Created a video explaining the concept
  - https://www.youtube.com/watch?v=mf2HhkRv5Vs

Android Game: Keep Running (https://github.com/hemshankar/Game_KeepRunning)

- Running game with various characters, enemies and equipments/powers
- Used LibGDX for development
- **Implemented in scalable fashion** so that new characters can be added easily.

Participated in Kaggle

- Used python, openCV and spark
- Participated in
  - Coupon Purchase Prediction
    - https://www.kaggle.com/c/coupon-purchase-prediction
  - Rossmann Store Sales
    - https://www.kaggle.com/c/rossmann-store-sales

Web Crawler (https://bitbucket.org/hemshankar/vyasam)

- Aim was to develop a website similar to GSMArena.com
- Crawler will collect all the data from various sites and put in the database

Web automation

- Using selenium, a framework/dashboard was provided to record the web activities
- Play the recorded activities with play and pause options
- Special features like wait for user input if any error occurs.
- Provide logging facilities
- Used **Java core + Java Swings + Selenium**

EDUCATIONAL DETAILS

- ***Masters of Technology*** in *Computer Science and Engineering (**CGPA: ~8**)* from ***Indian Institute of Technology, Roorkee.***
- ***Bachelors of Engineering*** *in Computer Science and Engineering (**CGPA:* **8.32**) from ***Bhilai Institute of Technology, Durg (CG).***
- *Senior School Certificate Examination form **CBSE*** in *Science (Physics, Chemistry, and Math)* with ***81.8% marks.***

- High School Certificate Examination from CGBSE, Raipur with ***74.83 % marks***

REFERENCES

Dr. Manoj Mishra
Professor, Deptt. of E & C E
Indian Institute of Technology Roorkee
manojfec@iitr.ernet.in
+91-1332-285642
https://www.linkedin.com/in/manoj-m-94862847/