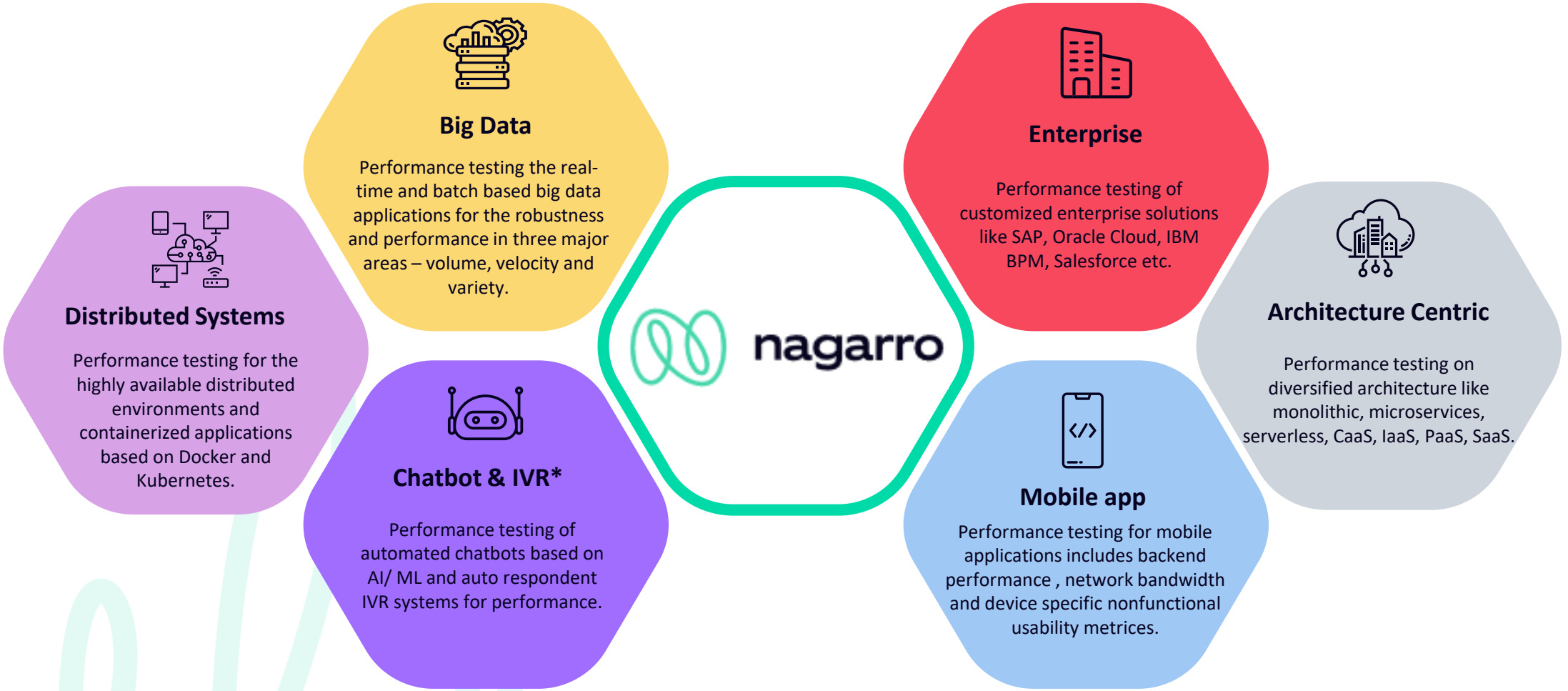
nagarro

Performance Engineering

Our services and offers

Our Performance Testing Expertise

Tech stack we cover at Nagarro



Our Performance Testing Experience in Niche Area

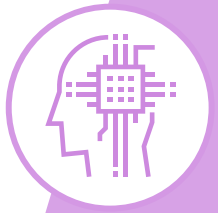
Innovative PT solutions for complex technical problems across business domains

Developed protocol agnostic framework

Developed a customized performance testing solution for a peer-to-peer video streaming application based on WebRTC. There was no open-source tool available and paid solutions were covering only limited scenarios.

Performance testing of ecommerce application

Conducted performance testing of an ecommerce application which helped the client to increase user footfall and enhanced end user experience.



Performance testing of AI/ML based application

Developed automated performance testing solution for an image recognization AI/ML application.

Performance testing of Big Data application

Developed an optimized PT solution for a traffic management application that uses historical data and real-time traffic reports to control traffic.

Performance testing of FaaS* (Serverless) application

Performed performance testing of a payment-based application developed on serverless architecture.

Performance testing of CaaS* application

Continuous performance testing for a Docker based multi-node application implemented in REST technology.

Possible Challenges in Application Performance Lifecycle

It is imperative to understand the factors affecting system performance before embarking on the task of handling them. The factors affecting may be divided into three performance large categories: project management oriented, testing and technical.

Project Planning	Performance Testing	Technical
Squeezed timeline for delivery. Low priority to performance during project planning.	Tool fitment for performance testing. Gathering realistic nonfunctional requirements.	Memory leaks , Array bound errors, inefficient buffering.
Unmanaged technical debts.	More affinity towards functional testing than performance testing.	Too many process cycles, large number of HTTP transactions.
Missing standard guidelines for code development.	Limited time and budget for performance testing.	Large no of file transactions between memory and disk. Inefficient session state management.
Tightly coupled system design with minimal focus on reusability.	Shift left or shift right dilemma.	Thread contention due to maximum concurrent users. Poor architecture sizing of peak load.
Unscalable design architecture.	Incompetence to handle complexities of emerging technologies.	Inefficient SQL statement. Lack of proper indexing on the database tables.
Ambiguous performance requirements.	Failed to model real time user load. Non-prudential approach to analyze results and logs.	Inappropriate server configurations.

3SR Approach for performance testing

01 Requirement Engineering
Brainstorming sessions with key stakeholders to refine the nonfunctional requirements. Using business domain expertise, creation of realistic workload models for performance test runs.

02 Rigorous Planning
Best fit performance test approach for tool evaluation, test environment detailing, KPI definition, risks assessment and dependency identification.

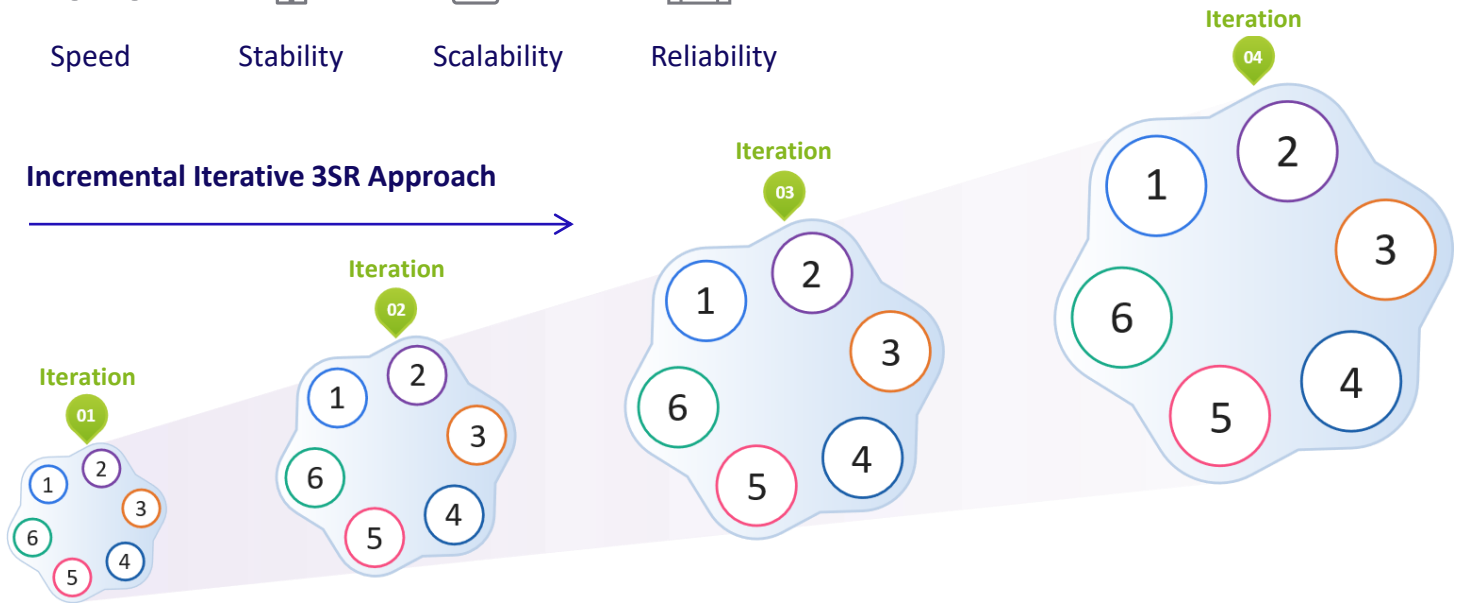
03 Robust & Realistic User Journeys
Script development representing real-time user journey. Reusable test data, mirroring real-time user transactions.

04 3SR Validation
Validation based on proven test processes, real-time load test monitoring, meaningful & insightful reporting, incremental and iterative setup with CI/CD.

Ensuring seamless user experience

Speed Stability Scalability Reliability

Incremental Iterative 3SR Approach



05 Insightful Monitoring & Analytics
Top to bottom modular approach for getting insights on performance bottlenecks using application performance management tools.

06 Dynamic Analysis & Optimization
Performance engineering and profiling activities using heap dump analysis, CPU analysis, DB tuning, thread dump analysis, vertical scaling, etc.

Performance Testing @ Nagarro using 3SR approach

Requirement Engineering
Brainstorming sessions with key stakeholders to refine the nonfunctional requirements. Using business domain expertise, creation of realistic workload models for performance test runs.

Rigorous Planning
Best-fit performance test approach for tool evaluation, test environment detailing, KPI definition, risks assessment and dependency identification.

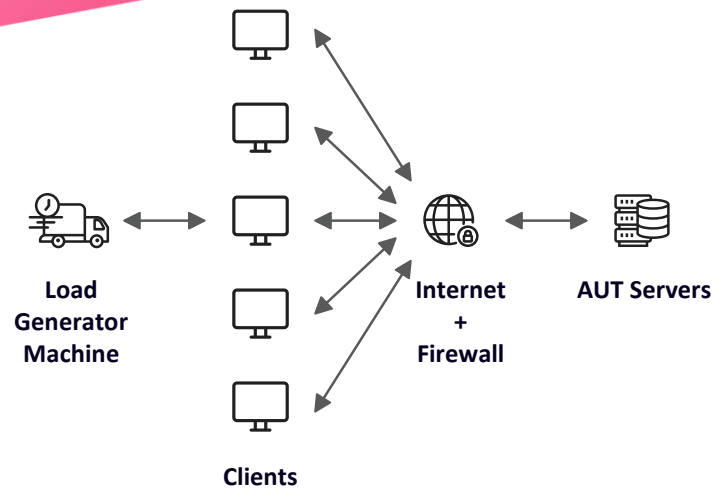
Robust & Realistic User Journeys
Script development representing real-time user journey. Reusable test data, mirroring real-time user transactions.

3SR Validation
Validation based on proven test processes, real-time load test monitoring, meaningful & insightful reporting, incremental and iterative setup with CI/CD.

Tools Evaluation

LoadStorm CLOUD LOAD TESTING TOOL
BlazeMeter
HP Loadrunner
IBM
SMARTBEAR LoadNinja
APACHE JMeter™
NeoLoad
Gatling

Load Simulation



Result Analysis

Server Side KPIs

- CPU Utilization
- Memory Utilization
- Networl I/O
- Disk I/O

End User KPIs

- Response Time
- Throughput
- Latency
- Data Transfer rate

Performance Monitoring using 3SR approach @Nagarro

Insightful Monitoring & Analytics
Top to bottom modular approach for getting insights on performance bottlenecks using application performance management tools.

Application Deep-Dive Analytics

Application architecture-based analysis for the extensibility and stability on all the modes of on-prem and cloud native environments in SaaS, PaaS, IaaS

Business Transaction Profiling

Analysis of the transaction flow through every tier of the application architecture to isolate where slowness is being caused.

User Experience Monitoring

This deals with tracking the experience of application users and identifying delivery times of CDNs and when they experience slowness, errors or downtime

Application Code Level Diagnostics

Tracing the errors through app server/web server/DB server/load balancer's log analytics for the exact RCA of performance bottleneck

Application Resource Viability

Many application issues occur due to slow network connectivity, a memory leak in the server, virtualization bottlenecks, storage hotspots, etc.

Tools

dynatrace, ExtraHop, splunk>, nesbula, Azure, ORACLE ENTERPRISE MANAGER, Compuware, New Relic., opsview, DATASTAX, APPDYNAMICS, APPENSURE, ca technologies, ZABBIX, NMAP, SiteScope, DATADOG, aws, Foglight, nGeniusONE, JENNIFER

During test runs we monitor through APM Tools based on multiple differentiators:

- **CPU Utilization -** Steal, softriq, user utilization, iowait
- **Disk** - Disk I/O, read/write execution, disc utilization, available, used, reserved for root.
- **Memory** - Utilization- available, committed, minor page fault, Kernel- Dirty, memory used by kernel, slab, kernel stack, page tables, VmallocUsed, Slab- reclaimed, unreclaimed
- **Network Stack -** TCP- bad data, timeout, timestamp, sack, fack, reno, inqueue, dropped, merged, pruned,, received, sent, failed, ecn- NoECTP, ECTPO.
- **Ipv4 Networking -** Used, received, sent, delivered, packets- received, sent, delivered, ICMP- received, sent.
- **Quality of service -** eth0(tc.world_in/out)- icmp, dns, vpn, ssh, mail, webserver, default, client.
- **Web server -** Active connection, requests, writing, idle, Connection rate- accepted, handled
  - https methods- GET, PUT, POST, Options
  - IP Protocol- IPv4, IPv6
- **DB Queries -** read/write, queries, generated

Performance Engineering using 3SR approach @Nagarro

Dynamic Analysis & Optimization
Performance engineering and profiling activities using heap dump analysis, CPU analysis, DB tuning, thread dump analysis, vertical scaling, etc.

Code Profiling Activities

- Statistical sample profiling (sampling)
- Function instrumentation profiling
- Sampling and call count instrumentation profiling
- Postmortem profiling for call count

Profiling Tools

Visual Studio
VisualVM
HEAP
ANTS Performance Profiler
CloudWatch
Android Profiler
Retrace
dotTrace
JProfiler
JFrog

What we do in Code Optimization

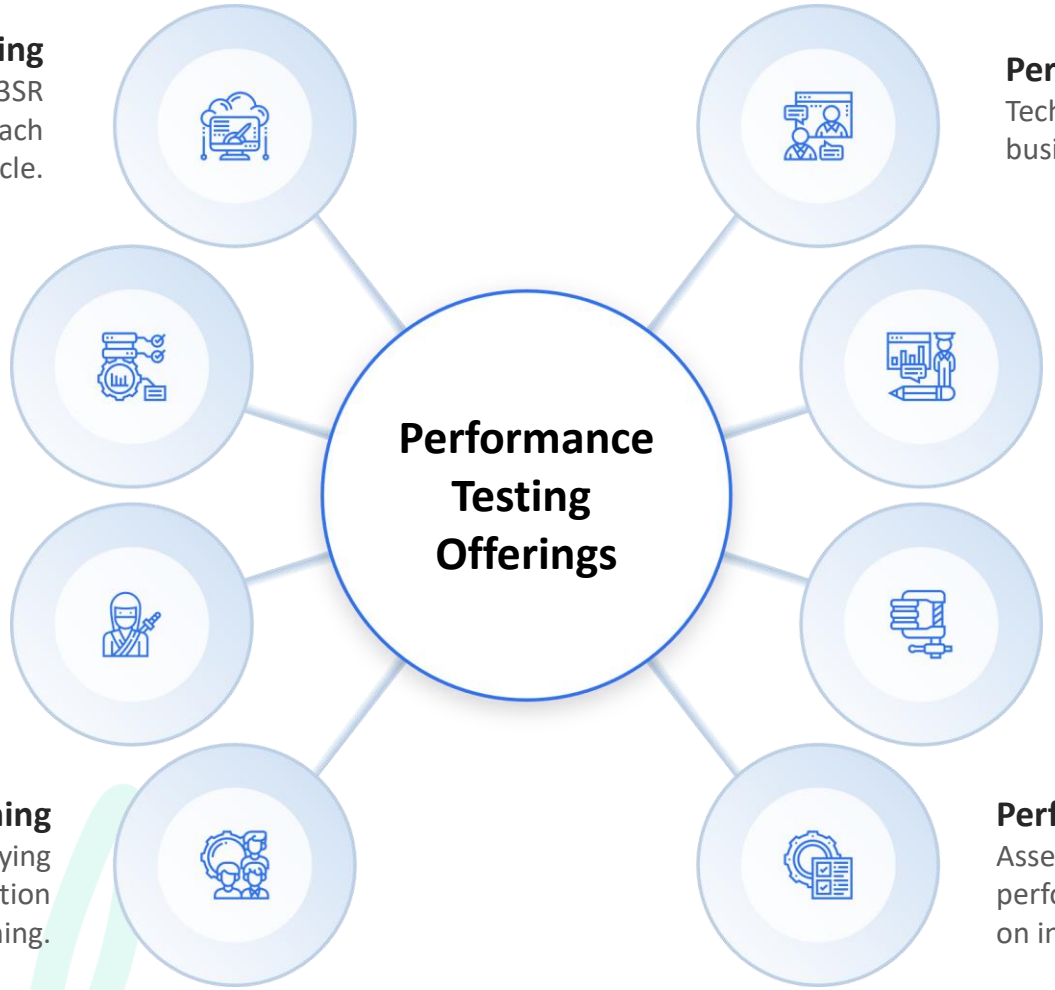
In code optimization, once we get code profiling done, then the outputs of code profiling needs to be fixed in code optimization phase. Exceptions needs to be handled, memory leaks needs to be fixed, unnecessary loops need to be removed and coding standard need to be followed.

Possible Bottlenecks

- Resource utilization (CPU usage, memory leakage)
- Thread blockage
- Garbage collection issues
- Code issue(redundant Code, unnecessary wait time/loops , sleeps
- Data base issues (long running queries, wrong DB configurations)

Performance Testing Offering

- Client-focused tailormade offerings



Performance Testing
Incremental and iterative 3SR performance testing approach during the entire project lifecycle.

Support Code Optimization
Support to the developer for performance bottlenecks insights.

Ninja
Nagarro's performance testing solution for end-to-end performance testing.

Capacity Planning
Validation of underlying infrastructure and application architecture for future planning.

Performance Test Consulting
Technical solutioning for complex business problems.

Training
Conducting knowledge sharing and training activities.

Tool Evaluation
Comparative tool analysis for best fit tool selection based on technology support and cost effectiveness.

Performance Test Evaluation
Assessment of existing performance test solutions based on industry best practices.

What differentiates us

- Key differentiators for performance testing

01

In-house Performance Testing Solution NINJA

Our Ninja performance testing solution provides end-to-end performance testing capabilities i.e., client side, server-side performance testing, log monitoring, and real-time application monitoring.

02

Our Unique Incremental Iterative 3SR Approach

Nagarro helps businesses optimize their PT&E activities with its unique, incremental, and iterative 3SR approach. Our performance engineering experts provide best fit performance testing solutions, while ensuring the key pillars like speed, stability, scalability, and reliability intact.

03

Centre of Excellence for Performance Testing

Dedicated R&D and exploration group which is continuously engaged in exploring new areas of performance testing. COE engineers work on different POCs covering niche and emerging areas of technology. We regularly publish various blogs and white papers and participate as speakers in different renowned conferences.
Thinking breakthrough sessions are organized to demonstrate innovative solutions to the clients.

04

Expert Consultants

Certified performance test engineers thriving for excellence in performance testing & engineering.
At Nagarro, we believe in continuous learning and have hands-on experience and know how on different tools of performance testing, monitoring, and profiling.

nagarro

Case Study

How we solve a complex problem?

Validation of performance, stability & scalability of web-based application

About Client:

The client is a leading customer to vendor multi-step payment solution provider.

Testing Tools

Apache JMeter

Monitoring

AWS CloudWatch, Performance Insight

Problem Statement

To validate the scalability and stability of the web-based application on the projected future user load conditions. The application was deployed on AWS platform.

Our Approach

- Testing for the concurrent users for the functionality and static content.
- Sudden incremental load testing for validation of the consistency and stability of the application components.
- Validation of concurrent database connections for the realistic user load conditions.
- RCA of the problems for the database and APIs.

Techincal Challenges

- Complex user flow for the automated script.
- Captcha integration in one of the flows to be bypassed during the testing.
- Proper application logging configuration to identify RCA.
- Test data preparation for real-time user flow simulation at scale.


Imagine what we can do together