Piacente Cristian 866020

Assignment A10 – Notes and conceptual map of the guest lecture

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

The guest lecture will be evaluated as assignment, based on
• Summary notes (1 or 2 pages at most)
• Conceptual map (single slide)

Your goal
• Aim to provide evidence of your active participation in the lectures

Grading
• Summary notes: evaluation scale
thorough / evidently partial / poor / missed
• Conceptual map:
well thought / limited / missed

MOVIRI INTRODUCTION

Moviri is a **global** technology company with a presence in Italy (Milan Bovisa and Padua), Los Angeles, Boston, and Singapore. Originally established as a tech hub within the Politecnico di Milano in 2000, Moviri is no longer a startup. It has evolved into a consultancy specializing in **Performance Engineering, Analytics, Cybersecurity, and the Internet of Things**, while also acting as a startup accelerator (part of product companies). Originally named Neptuny, the company was acquired by BMC in 2010 and renamed Moviri. Its history also includes other companies like ContentWise (incubated in 2007, spun off in 2015), Cleafy, Arduino, and Akamas.
The incubation of the product companies ContentWise and Akamas occurred within Moviri, whereas Cleafy and Arduino were investments acquired within the Moviri group.

PERFORMANCE ENGINEERING

Performance Engineering is divided into **three services**:

- **Planning & Control**: for instance, in an e-commerce company, there is a logic of machine capacity to support the software delivered to the customer, which helps determine if the IT infrastructure is calibrated to the real use of the application
    - Includes **Capacity Management**
- **Design & Validation**: testing and validation of software
    - Includes **Performance Testing**
- **Observability**: monitoring and observation of IT infrastructure and service performance, such as a Telco company managing internet lines. With tools like Splunk, logs from various heterogeneous systems can be collected and temporally correlated to identify issues
    - Definition: “In control theory, observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs”
    - Necessary because many factors come into play when a system goes into production
    - Achieved in the IT world through three pillars
        - Metrics: CPU, memory, thread usage, etc.
        - Traces: observing how the code is functioning
        - Logs: actions the application is performing, inserted by programmers

Solutions include:

- **Peak Demand Events**: a critical aspect, Performance Engineering must prevent situations where the system collapses due to excessive requests (e.g. during Black Friday for an e-commerce site)
- **AI-Powered Performance Engineering**: systems use Artificial Intelligence to optimize, which can be cloud-based to select, for example, the optimal configuration of AWS services that reduce costs and maximize performance
- **Datacenter & Cloud Optimization**: optimization of services that can be hybrid (partially in the cloud)
- **IoT & Client Side Performance**: limited infrastructure (embedded devices), the application must be optimized beyond imagination
- **Kubernetes & Microservices**
- **Agile, DevOps & CI/CD Automation**
- **Open Source Performance Framework**

Practical example from Moviri's spin-off: **Akamas**, an optimizer that includes a smart system to optimize hyperparameters governing the operation of AWS, Docker, Java, MongoDB, and even Linux.

Piacente Cristian 866020

Assignment A10 – Notes and conceptual map of the guest lecture

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca

Akamas can optimize both individual microservices and orchestrate microservices through Reinforcement Learning.

CAPACITY MANAGEMENT

Capacity in terms of an object that must deliver a product, whether material or immaterial, refers to the level of productivity, or how much it can deliver over time.

Capacity themes are familiar to us all, from everyday experiences (e.g. a full bus).

From an IT perspective: "The goal of the Capacity Management **process** is to ensure that sufficient **IT capacity** in all areas of IT can be provided **in a timely manner**, and that it is **cost-justified** and aligned to the agreed **current and future needs of the business**".

For effective capacity planning, a solid and precise understanding of how applications in my data center utilize the resources provided (CPU, memory, storage, data center cooling, licenses, etc.) is required.
This involves gathering information through monitoring to understand how each application and system is using resources to predict and prevent saturation.

The complexity lies in monitoring all applications interacting with numerous technological stacks, collecting information, standardizing it, bringing it into a single database, and using it for analysis to extract all necessary information regarding capacity.

A manager who neglects capacity management fears the question: "Could you tell me how much it costs to run this application? Because we were thinking of cutting it".

For example, considering a company **acquisition**, the acquired company will integrate all its clients into the data center of the acquiring company. A question posed to the manager might be: "Starting January 1, 2025, our user base will increase by 50% due to the finalization of an acquisition: can our systems handle it, or what kind of upgrades are necessary to support the new application load?".
Capacity management is crucial to determine if part of the users can be accommodated immediately and to account for **nonlinear correlation effects** between memory use and CPU use, which depends on what the application does.

This opens up many possibilities for analysis, including on themes like green IT (e.g., obtaining the CO_2 emitted by a service).

PERFORMANCE TESTING

Why should performance testing be conducted on an application?

There are many good reasons, certainly because an application that does not perform well does not reflect well on us, resulting in a negative digital experience for the customer and a loss of productivity for the company. Competitors gain advantage, leading to profit loss. If you are a significant brand presenting a digital service of a certain level and experience a performance degradation following an update, it not only looks bad but there is also a risk of making the news if the service fails, resulting in a business reputation damage.

Here's what happens if the application is not performing well:

- **Bounce Rate**: increase in bounce probability (the number of visitors who view only one page divided by the total site visitors)
- **Negative Publicity**
- **Brand Reputation**
- **Forever Lost**: the customer never returns
- **Productivity Loss**

An example: a 0.1 second delay in Amazon's page loading could lead to a 1% loss in revenue.

Piacente Cristian 866020

Assignment A10 – Notes and conceptual map of the guest lecture

Software Quality, Academic Year 2023-2024, University of Milan – Bicocca