TECHERA 2023

Arthur Samuel Keene

"Technology is best when it brings people together."

Madanapalle Institute of Technology & Science

(UGC- Autonomous)

Department of Computer Science & Engineering

MESSAGE FROM THE CORRESPONDENT

I feel exhilarated that the Department of Computer Science &Engineering of MITS is bringing out a magazine called TECHERA from the year 2023. This Magazine brings out the intellectual brilliance in various new techniques introduced in Information Technology industry.

``HARD WORK, SINCERITY, DEDICATION AND ENTHUSIASTIC DEVOTION TO WORK WILL FETCH YOU UNBOUND SUCCESS, MAY THE LORD SHOWER HISBLESSINGS ON YOU``.

I heartily congratulate the students and the staffs of CSE Department andWish them a grand success.

Dr. N. Vijaya Bhaskar Choudary

Correspondent

MESSAGE FROM THE PRINCIPAL

I feel delighted about the magazine “TECHERA” to be hosted by the Department of Computer Science& Engineering of MITS. On this magnanimous occasion, I congratulate all the students and faculty members of department for their great efforts and coordination in bringing out the magazine a great success.

Principal

Dr. C. Yuvaraj

MESSAGE FROM THE HEAD OF THE DEPARTMENT

TECHERA is dedicated for addressing the emerging topics and challenges in technology. **TECHERA** is to create great awareness on new innovative ideas and technologies. I wish the readersof **"TECHERA"** for their support and also can provide the useful feedback to improve the standardsof magazine.

Dr. R. Kalpana

Head of the Department/CSE

EDITORIAL DESK

The annual release of the department magazine **"TECHERA – 2023",** mark the spirit of explorationamong students in an environment of erudition.

This year's edition of "**TECHERA - 2023**" focuses on current trends in Computer Science and Information Technology which are the major rays of hope for developing a new world of science. It is a collection of information and facts, featuring the recent developments of fascinating and conceptual communication.

The editorial team owes its gratitude to all who have made "**TECHARA - 2023", a** scintillating event.

ABOUT MITS

Madanapalle Institute of Technology & Science is established in 1998 in the picturesque andpleasant environs of Madanapalle and is ideally located on a sprawling 26.17- a c r e campus on Madanapalle - Anantapur Highway (NH-205) near Angallu, about 10km away from Madanapalle.

MITS, originated under the auspices of Ratakonda Ranga Reddy Educational Academy under the proactive leadership of and **Dr. N. Vijay Bhaskar Choudary, Secretary & Correspondent** of the Academy.

MITS is governed by a progressive management that never rests on laurels and has been strivingconscientiously to develop it as one of the best centers of Academic Excellence in India. The Institution's profile is firmly based on strategies and action plans that match changing demandsof the nation and the student’s fraternity. MITS enjoys constant support and patronage of NRI'swith distinguished academic traditions and vast experience in Engineering &Technology.

ABOUT DEPARTMENT

The Department of Computer Science & Engineering was established in 1998 and had been playing a vital role in producing value based professionals ever since 1998. The department offers one 4 years undergraduate program to cater to the ever-challenging needs of technical excellence in the emerging areas of Computer Science & Engineering. The course is flexible and structured to meet the evolving needs of the IT industry. The CSE department has eminent faculty members with rich academic and industry exposure, who have pursued a Masters/Ph.D. Degree from prestigious institutions like NITs, IITs, and Central Universities within India and abroad. Many research activities in the domain of Artificial Intelligence (AI) and Machine Learning (ML) are under progress. Department of CSE has good interactions and MoUs with leading Industries for technology domain Training & Development Industries. It organizes Symposia, Exhibitions, Conferences, Seminars and Workshops for both students and Faculty belonging to various Technical Educational Institutions, Research Scholars of Research Institutes and Industries all over India.

Our students got placement offers in various top MNCs like TCS, Infosys, IBM, Tech Mahindra, Accenture, Mind Tree etc., for deserving & esteemed packages of more than 4.5 Lakhs to 24 LPA. The department also offers training in certification programs and encourages students in self-learning with MooCs such as NPTEL, Microsoft, Coursera, edX, etc. Activities in Research outcomes are presented/published in National / International Conferences / Journals. The students can become members of CSI, IEEE-CS, IEEE-PCS, IEEE-WIE, ACM & IET and participate various activities through these professional body. Department is committed to encourage students/researchers to carry out innovative research in the field of Computer Science & Engineering, keeping excellence in focus and deliver quality services to match the needs of the technical education system, industry and society. Students of CSE department are motivated to be innovative in their thinking while being strong in the Computer Science Core Knowledge. The department is also accredited by NBA(National Board of Accreditation) of All India Council for Technical Education (AICTE), New Delhi.

DEPARTMENT VISION

To excel in technical education and research in area of Computer Science &Engineering and to provide expert, proficient and knowledgeable individuals with high enthusiasm to meet the Societal challenges.

DEPARTMENT MISSION

M1: To provide an open environment to the students and faculty that promotesprofessional and personal growth.

M2: To impart strong theoretical and practical background across the computer science discipline with an emphasis on software development and research.

M3: To inculcate the skills necessary to continue their education after graduation,as well as for the societal needs.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

The Program Educational Objectives (PEOs) of the department of CSE are givenbelow:

PEO1: Gain Successful Professional career in IT industry as an efficient software engineer.

PEO2: Succeed in Master / Research programmers to gain knowledge onemerging technologies in Computer Science and Engineering.

PEO3: Grow as a responsible computing professional in their own area of interest with intellectual skills and ethics through lifelong learning approach to meet societal needs.

PROGRAM SPECIFIC OUTCOMES (PSOs)

The Computer Science and Engineering Graduates will be able to:

PSO1: Apply mathematical foundations, algorithmic principles and computing techniques in the modelling and design of computer - based systems.

PSO2: Design and develop software in the areas of relevance under realisticconstraints.

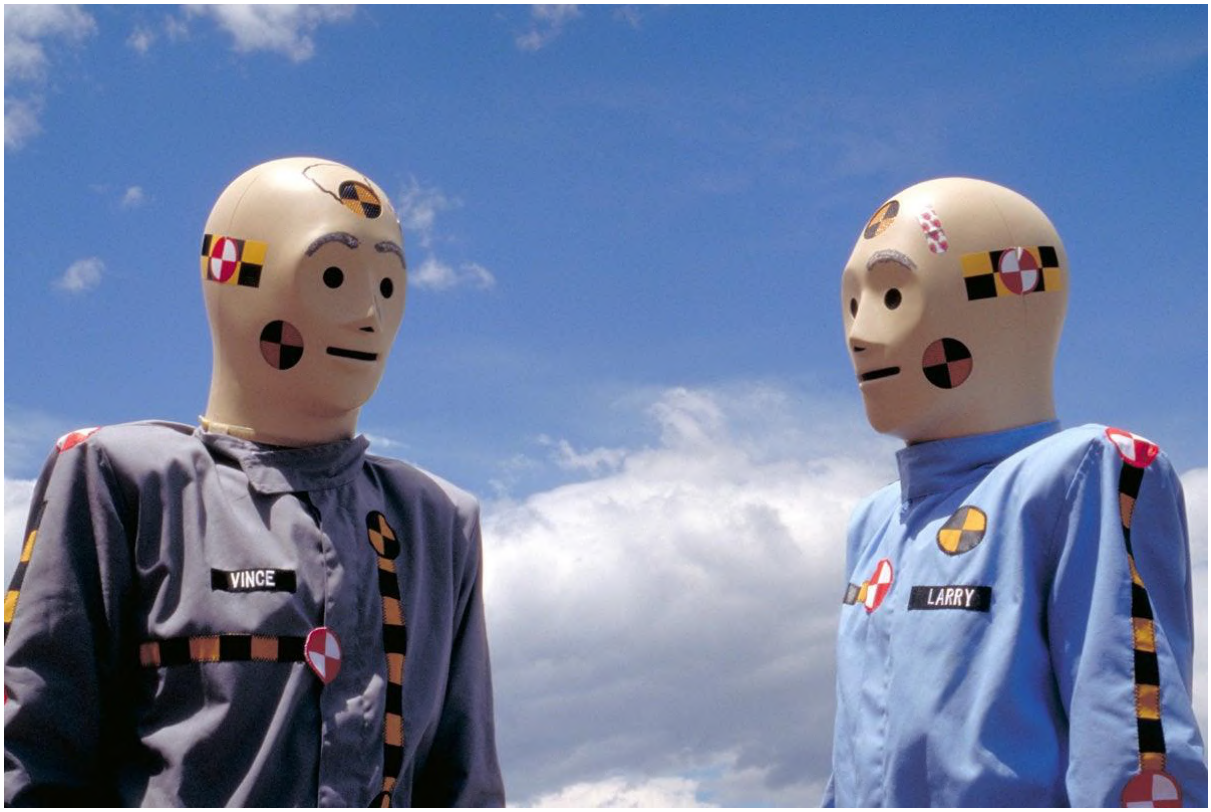
PSO3: Analyze real world problems and develop computing solutions by applyingconcepts of Computer Science.

CONTENTS

1.Using Moq: A Simple Guide to Mocking for .NET

We often need to write unit tests for code that accesses an external resource such as a database or a file file system. If such resources are not available, the only way to ensure that the tests can execute is by creating mock objects. In essence, by drawing on fake implementations of these underlying dependencies, you can test the interaction between the method being tested and its dependencies. Three of the most popular mocking frameworks for .Net developers are Rhino Mocks, Moq, and NMock.

Among these, Moq may be the most flexible and easy to use. The Moq framework provides an elegant way to set up, test, and verify mocks. This article presents a discussion of Moq and how it can be used to isolate units of code from their dependencies.



Getting started with Moq.

You can use Moq to create mock objects that simulate or mimic a real object. Moq can be used to mock both classes and interfaces. However, there are a few limitations you should be aware of. The classes to be mocked can't be static or

sealed, and the method being mocked should be marked as virtual. (Note there are workarounds to these restrictions. You could mock a static method by taking advantage of the adapter design pattern, for example.)

The first step in using Moq is to install it so that you can use it in your unit test project. You can download Moq from GitHub and add references as appropriate. However, I prefer installing Moq via NuGet because it is both easier and less likely to miss references. You can install Moq by using the following command at the NuGet command line.

Install-Package Moq

How to mock interfaces using Moq

Let's start by mocking an interface. The syntax for creating a mock object using the Mock class is given below.

Mock<TypeToMock> mockObjectType=new Mock<TypeToMock>();

Now, consider the following interface named IAuthor.


```
public interface IAuthor
{
int Id { get; set; }
string FirstName { get; set; }
string LastName { get; set; }
}
```

Using the Moq framework, you can create a mock object, set property values, specify parameters, and return values on the method calls. The following code snippet illustrates how you can create an instance from the IAuthor interface using Moq.

```
var mock = new Mock<Author>();
```

Note that the Mock class belongs to the Moq framework and contains a generic constructor that accepts the type of interface you want to create. Moq takes advantage of lambda expressions, delegates, and generics. All of this makes using the framework very intuitive.

The following code snippet shows how you can mock the IAuthor interface and provide the properties of the mocked instance with the appropriate values. Note how we use Assert to verify the values of the properties of the mocked instance.

```
var author = new Mock<IAuthor>();
author.SetupGet(p => p.Id).Returns(1);
author.SetupGet(p => p.FirstName).Returns("Joydip");
author.SetupGet(p => p.LastName).Returns("Kanjilal");
Assert.AreEqual("Joydip", author.Object.FirstName);
Assert.AreEqual("Kanjilal", author.Object.LastName);
```

How to mock methods using Moq

Let's now consider the following class named Article. The Article class contains just one method called GetPublicationDate that accepts an article Id as parameter and returns the publication date of the article.

```
public class Article
{
public virtual DateTime GetPublicationDate(int articleId)
{
throw new NotImplementedException();
}
}
```

Because the GetPublicationDate method is not yet implemented in the Article class, the method has been mocked to return the current date as the publication date, as shown in the code snippet given below.

```
var mockObj = new Mock<Article>();
mockObj.Setup(x => x.GetPublicationDate(It.IsAny<int>())).Returns((int x) => DateTime.Now);
```

The Setup method is used to define the behavior of a method that is passed to it as a parameter. In this example, it is used to define the behavior of the GetPublicationDate method. The call to It.IsAny<int>() implies that the GetPublicationDate method will accept a parameter of type integer; It refers to a static class. The Returns method is used to specify the return value of the method that is specified in the Setup method call. In this example, the Returns method is used to specify the return value of the method as the current system date.

Moq allows you to verify whether a particular method or a property was called. The following code snippet illustrates this.

```
mockObj.Verify(t => t.GetPublicationDate(It.IsAny<int>()));
```

Here we're using the Verify method to determine if the GetPublicationDate was called on the mock object.

How to mock base class methods using Moq

Consider the following piece of code. We have two classes here—the RepositoryBase class and the AuthorRepository class that extends it.

```
public abstract class RepositoryBase
{
public virtual bool IsServiceConnectionValid()
{
```

```
//Some code
}
}
public class AuthorRepository : RepositoryBase
{
public void Save()
{
if (IsServiceConnectionValid())
{
//Some code
}
}
```

Now suppose we want to check if the database connection is valid. However, we may not want to test all the code inside the IsServiceConnectionValid method. For instance, the IsServiceConnectionValid method might contain code that pertains to a third-party library. We would not want to test that, right? Here is where the CallBase method in Moq comes to the rescue.

In situations like this, where you have a method in the base class that has been overridden in the mocked type, and you need to mock the base version of the overridden method only, you can draw on CallBase. The following code snippet shows how you can create a partial mock object of the AuthorRepository class by setting the CallBase property to true.

```
var mockObj = new Mock<AuthorRepository>(){CallBase = true};
mockObj.Setup(x => x.IsServiceConnectionValid()).Returns(true);
```

The Moq framework makes it easy to create mock objects that mimic the behavior of classes and interfaces for testing, with just the functionality you need.

Aritcle by:

ABHISHEK. B

20691A0502

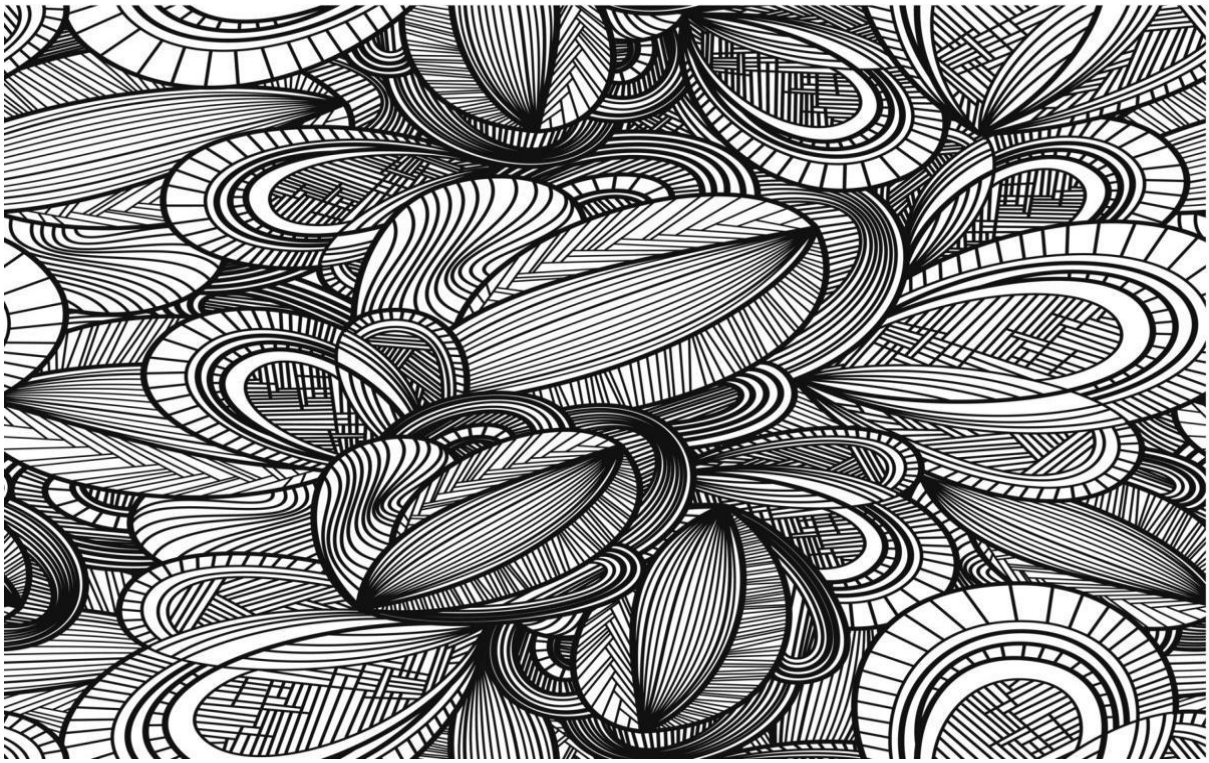
2. How to use the Memento design pattern in C#

Take advantage of the Memento design pattern to store and restore an object's state to support undo or rollbacks in your application.

We use design patterns to solve common design problems and reduce the complexities in our source code. The Memento design pattern is a behavioral design pattern that can be used to provide an undo or rollback capability in an application, or simply to reset the state of an object in an ASP.Net web application, for example. By storing an object's state to an external location called a Memento, this pattern allows that state to be restored to the object at a later time. Let's explore how we can use the Memento design pattern in C#.

Every object has its internal state. A Memento gives us a way to save that state and restore it while still abiding by the principles of encapsulation, which dictate that non-public members of an instance of a class should not be available to the outside world. This is because the Memento is available only to the object whose state it has stored.

The participants in the Memento design pattern include a Memento, an Originator, and a Caretaker. While the Memento class stores the state of the object, the Originator creates the Memento and uses it to restore the state when needed. The Caretaker is responsible only for storing the Memento—it is not supposed to alter the Memento instance.



Implementing the Memento pattern

In this section we will implement the Memento design pattern in C#. We will create a simple program that has three classes – a Calculator class, a Memento class, and the client, i.e. the Main method.

Refer to the Calculator class given below.

```
public class Calculator
{
int result;
public Calculator(int i = 0)
{
result = 0;
}
public void SetResult(int i = 0)
{
```

```
this.result = 0;
}
public void Add(int x)
{
result += x;
}
public void Subtract(int x)
{
result -= x;
}
public int GetResult()
{
return result;
}
public Memento CreateMemento()
{
Memento memento = new Memento();
memento.SetState(result);
return memento;
}
public void SaveState (Memento memento)
{
result = memento.GetState();
}
}
```

Note the CreateMemento and SetMemento methods in the Calculator class. While the former creates a Momento instance, the latter retrieves the saved state and assigns the value back to the result variable.

The Memento class

The Memento class contains two methods, SetState and GetState. While the former is used to store the state information, the latter is used to retrieve the saved state.

Nominations are open for the 2024 Best Places to Work in IT

```
public class Memento
{
int state;
public int GetState()
{
return state;
}
public void SetState(int state)
{
this.state = state;
}
}
```

The client in this example is the Main method that creates an instance of the Calculator class and makes calls to the Add and Subtract methods to perform computation. In addition, Main saves the state information at a particular checkpoint by making a call to the SaveState method. Later, this saved state is restored and the value of the result variable is displayed at the console window. This is illustrated in the code snippet given below.

```
static void Main(string[] args)
{
Calculator calculator = new Calculator();
calculator.Add(5);
calculator.Add(10);
calculator.Subtract(10);
Memento checkPoint = calculator.CreateMemento();
calculator.Add(100);
Console.WriteLine("The value of the result variable is: "+calculator.GetResult());
calculator.SaveState (checkPoint);
Console.WriteLine("The value of the result variable at first checkpoint is: " + calculator.GetResult());
Console.Read();
}
```

The complete Memento pattern example

Here is the complete program for your reference.

```
class Program
{
static void Main(string[] args)
{
Calculator calculator = new Calculator();
calculator.Add(5);
calculator.Add(10);
calculator.Subtract(10);
Memento checkPoint = calculator.CreateMemento();
```

```
calculator.Add(100);
Console.WriteLine("The value of the result variable is: "+calculator.GetResult());
calculator.SaveState (checkPoint);
Console.WriteLine("The value of the result variable at first checkpoint is: " + calculator.GetResult());
Console.Read();
}
}

public class Calculator
{
int result;
public Calculator(int i = 0)
{
result = 0;
}
public void SetResult(int i = 0)
{
this.result = 0;
}
public void Add(int x)
{
result += x;
}
public void Subtract(int x)
{
```

```
result -= x;
}
public int GetResult()
{
return result;
}
public Memento CreateMemento()
{
Memento memento = new Memento();
memento.SetState(result);
return memento;
}
public void SetMemento(Memento memento)
{
result = memento.GetState();
}
}

public class Memento
{
int state;
public int GetState()
{
return state;
}
public void SetState(int state)
```

```
{
this.state = state;
}
}
```

The Memento design pattern gives us a handy way to store and retrieve an object's state. You can take advantage of this pattern to perform an undo or a rollback. However, one of the downsides of using this pattern is that the process of saving an object's state and restoring it later can take quite some time—i.e., it may be detrimental to the application's performance. So, when using the Memento pattern, be sure to keep performance in mind. Finally, also be sure that the internal structure of your object is not exposed to the outside world.

Article by

Roopasree G

21691A05H5

3. Oracle adds machine learning features to MySQL Heatwave

Oracle is adding new machine learning features to its data analytics cloud service MySQL HeatWave.

MySQL HeatWave combines OLAP (online analytical processing), OLTP (online transaction processing), machine learning, and AI-driven automation in a single MySQL database.

The new machine learning capabilities will be added to the service's AutoML and MySQL Autopilot components, the company said when it announced the update on Thursday.

[Keep up with the latest developments in data analytics and machine learning. Subscribe to the InfoWorld First Look newsletter]

While AutoML allows developers and data analysts to build, train and deploy machine learning models within MySQL HeatWave without moving to a separate service for machine learning, MySQL Autopilot provides machine learning-based automation to HeatWave and OLTP such as auto provisioning, auto encoding, auto query plan, auto shape prediction and auto data placement, among other features.

AutoML augments time series forecasting via machine learning

The new machine learning-based capabilities added to AutoML include multivariate time series forecasting, unsupervised anomaly detection, and recommender systems, Oracle said, adding that all the new features were generally available.

“Multivariate time series forecasting can predict multiple time-ordered variables, where each variable depends both on its past value and the past values of other dependent variables. For example, it is used to build forecasting models to predict electricity demand in the winter considering the various sources of energy used to generate electricity,” said Nipun Agarwal, senior vice president of research at Oracle.

In contrast to the regular practice of having a statistician trained in time-series analysis or forecasting to select the right algorithm for the desired output, AutoML’s multivariate time series forecasting automatically preprocesses the data to select the best algorithm for the ML model and automatically tunes the model, the company said.

“The HeatWave AutoML automated forecasting pipeline uses a patented technique that consists of stages including advanced time-series preprocessing, algorithm selection and hyperparameter tuning,” said Agarwal, adding that this automation can help enterprises save time and effort as they don’t need to have trained statisticians on staff.

The multivariate time series forecasting feature, according to Constellation Research Principal Analyst Holger Muller, is unique to Oracle’s MySQL HeatWave.

“Time series forecasting, multivariate or otherwise, is not currently offered as part of a single database that offers machine learning-augmented analytics. AWS, for example, offers a separate database for time series,” Muller said.

HeatWave enhances anomaly detection

Along with multivariate time series forecasting, Oracle is adding machine-learning based "unsupervised" anomaly detection to MySQL HeatWave.

In contrast to the practice of using specific algorithms to detect specific anomalies in data, AutoML can detect different types of anomalies from unlabeled data sets, the company said, adding that this feature helps enterprise users when they don't know what anomaly types are in the dataset.

"The model generated by HeatWave AutoML provides high accuracy for all types of anomalies — local, cluster, and global. The process is completely automated, eliminating the need for data analysts to manually determine which algorithm to use, which features to select, and the optimal values of the hyperparameters," said Agarwal.

In addition, AutoML has added a recommendation engine, which it calls recommender systems, that underpins automation for algorithm selection, feature selection, and hyperparameter optimization inside MySQL HeatWave.

"With MySQL HeatWave, users can invoke the ML_TRAIN procedure, which automatically trains the model that is then stored in the MODEL_CATALOG. To predict a recommendation, users can invoke ML_PREDICT_ROW or ML_PREDICT_TABLE," said Agarwal.

Business users get MySQL HeatWave AutoML console

In addition, Oracle is adding an interactive console for business users inside HeatWave.

"The new interactive console lets business analysts build, train, run, and explain ML models using the visual interface — without using SQL commands or any

coding," Agarwal said, adding that the console makes it easier for business users to explore conditional scenarios for their enterprise.

"The addition of the interactive console is in line with enterprises trying to make machine learning accountable. The console will help business users dive into the deeper end of the pool as they want to evolve into 'citizen data scientists' to avoid getting into too much hot water," said Tony Baer, principal analyst at dbInsight.

The console has been made initially available for MySQL HeatWave on AWS.

Oracle also said that it would be adding support for storage on Amazon S3 for HeatWave on AWS to reduce cost as well improve the availability of the service.

"When data is loaded from MySQL (InnoDB storage engine) into HeatWave, a copy is made to the scale-out data management layer built on S3. When an operation requires reloading of data to HeatWave, such as during error recovery, data can be accessed in parallel by multiple HeatWave nodes and the data can be directly loaded into HeatWave without the need for any transformation," said Agarwal.

MySQL Autopilot updates

The new features added to MySQL HeatWave include two new additions to MySQL Autopilot — Auto Shape prediction advisor integration with the interactive console and auto unload.

"Within the interactive console, database users can now access the MySQL Autopilot Auto shape prediction advisor that continuously monitors the OLTP workload to recommend with an explanation the right compute shape at any given time — allowing customers to always get the best price-performance," Agarwal said.

The auto unload feature, according to the company, can recommend which tables to be unloaded based on workload history.

“Freeing up memory reduces the size of the cluster required to run a workload and saves cost,” Agarwal said, adding that both the features were in general availability.

HeatWave targets smaller data volumes

Oracle is offering a smaller shape HeatWave to attract customers with smaller sizes of data.

In contrast to the earlier size of 512GB for a standard HeatWave node, the smaller shape will have a size of 32GB with the ability to process up to 50GB for a price of $16 per month, the company said.

In addition, the company said that data processing capability for its standard 512GB HeatWave Node has been increased from 800GB to 1TB.

“With this increase and other query performance improvements, the price performance benefit of HeatWave has further increased by 15%,” said Agarwal.

Article by:

Raghavendra V

21691A05F5

4. How to reduce your devops tool sprawl

After spending the last decade investing in devops, many companies are experiencing a hangover of sorts: tool sprawl. While their software delivery processes have become more streamlined, more efficient, and more reliable, they also have many more tools to license, maintain, and manage.

Tool sprawl is often seen as a natural result of the flexibility and empowerment of dev teams to choose their own tools, but organizations now understand the need for a single, streamlined system. While flexibility to choose the right tool for the job has enabled teams to move quickly, the result is a complex web of systems and processes to deliver software.

There are three main reasons you should consider tool consolidation now:

A recession that has every organization re-examining budgets

Heightened focus on security and the impact that sprawl has on securing software supply chains and IT systems

Improved efficiency and developer experience, which is driving the recent interest in platform engineering. Consolidating toolchains directly impacts all three of these areas.

If you're a devops expert considering a tool consolidation journey, here are three areas ripe for consolidation.

Application security tooling

A recent survey by Gartner found that organizations are making a shift towards consolidating their security vendors, with the number rising from 29% in 2020 to 75% in 2022. "Security and risk management leaders are increasingly dissatisfied with the operational inefficiencies and the lack of integration of a heterogenous security stack," said John Watts, VP Analyst at Gartner. "As a result, they are consolidating the number of security vendors they use."

Between static application security testing (SAST), dynamic application security testing (DAST), software composition analysis (SCA), and the multiple other types of application security solutions available today, it's possible for organizations to have a dozen different tools in place to ensure their released software applications are free from exploitable vulnerabilities.

More point solutions, however, don't guarantee a comprehensive approach to application security. Each tool represents an additional point of complexity in your security workflow, negatively impacting developer velocity and security risk. Ultimately, security and devops teams have to use different applications and policies to attempt to keep security consistent across their component ecosystem.

Nominations are open for the 2024 Best Places to Work in IT

Package and artifact management and storage

Teams developing new products often have to use free or low-cost solutions. As software engineering and development teams grow, they naturally adopt additional tooling and technologies. Over time, this increases the number of places development teams store their artifacts, creating sprawl, impeding

automation, hindering security, and requiring manual efforts to build and release software updates.

It’s not uncommon for organizations to get to a point where they’re storing software artifacts in any number of the following locations:

Package managers such as Maven, PyPI, and NPM

Docker Hub or other container registries

GitHub, GitLab, Bitbucket or other version control systems

General-purpose storage such as Amazon S3 buckets, Google Drive, and local share drives

Storing and managing artifacts in multiple locations is great for small development projects, but when teams need to speed up releases, or share components across teams (e.g., microservice architectures), or work across geographical boundaries, the ad-hoc web of storage solutions falls flat.

Consolidating onto a single system for all dependencies, build artifacts, and their metadata allows for enhanced automation and a single place to apply your application security efforts.

Systems and data monitoring

The Moogsoft State of Availability Report indicated that, on average, engineers are in charge of overseeing 16 monitoring tools—and this number could rise to 40 when service level agreements (SLAs) become more stringent. Having such a broad selection of tools can be chaotic for your teams, and the costs associated with licensing, managing, and maintaining them are high.

Generally speaking, the more visibility you have over your processes, infrastructure, and applications, the better. But too many monitoring and logging tools generate data silos, keeping you from accessing and exploring your data when you need it. Creating a single-pane-of-glass view across your entire tech

stack not only allows for cross-functional insights, but also enhances the value of all those logs your various tools are generating.

If you've already addressed consolidating these areas, here are a few more to consider:

CI and CD tooling

Distribution and caching

Source and VCS tools

It goes without saying that you can't consolidate everything. There will always be important features or capabilities that you must maintain in your existing toolsets. But if you're serious about consolidation, consider the role a single platform can play in not only reducing the number of tools you leverage but connecting and integrating the solutions in your newly consolidated tech stack.

If you're interested in exploring tips to go about tool consolidation at your organization, check out JFrog's recent webinar on the topic. To stay up-to-date on the latest devops trends, check out JFrog's blog.

Sean Pratt is a senior devops evangelist at JFrog, where he is responsible for helping businesses understand the many benefits of devops, tools consolidation, platform engineering, and cloud-native applications.

New Tech Forum provides a venue to explore and discuss emerging enterprise technology in unprecedented depth and breadth. The selection is subjective, based on our pick of the technologies.

Article by:

SAI MOHAN SRIRAM. G

20691A3528

5.Google unveils PaLM 2 AI language model

Google has introduced PaLM (Pathways Language Model) 2, an update to its next-generation large language model with improved multilingual, coding, and reasoning capabilities.

For multilingual tasks, PaLM 2 was more heavily pre-trained on multilingual text, spanning more than 100 languages, thus improving the software's ability to understand and translate nuanced text including idioms, poems, and riddles.

In the coding realm, PaLM 2 was pre-trained on a large quantity of available source code data sets. The model "excels" at popular programming languages such as Python and JavaScript, Google said, but is also capable of generating specialized code in languages such as Fortran, Prolog, and Verilog.

PaLM 2 also digested a wider-ranging data set as part of its pre-training than its predecessor including scientific papers, web pages, and mathematical

expressions, Google said. As a result PaLM 2 demonstrates improved capabilities in common sense reasoning, mathematics, and logic.

Developers can sign up to use the PaLM 2 model or use the model in Vertex AI.

Google noted that PaLM 2 is also faster and more efficient than previous models, while coming in four sizes for a range of use cases, allowing PaLM 2 to be fine-tuned to support entire classes of products. PaLM 2 already powers more than 25 Google products and features. These include:

The Bard generative AI platform. PaLM 2 also powers the Bard coding update, helping developers code.

Workspace features for Gmail and Google Docs, and for organization in Google Sheets.

Med-Palm 2, a large language model for use in healthcare.

Sec-PaLM 2, a version of PaLM 2 for security use cases.

Duet for AI Cloud, an AI-powered collaborator.

PaLM was unveiled in April as a 540-billion-parameter, dense decoder-only Transformer model trained with the Pathways system.

Article by:

Venkata Sai Hithesh Reddy. k

20691a05l9

6. Best practices for every MongoDB deployment

MongoDB is a non-relational document database that provides support for JSON-like storage. Its flexible data model allows you to easily store unstructured data. First released in 2009, it is the most commonly used NoSQL database. It has been downloaded more than 325 million times.

MongoDB is popular with developers because it is easy to get started with. Over the years, MongoDB has introduced many features that have turned the database into a robust solution able to store terabytes of data for applications.

As with any database, developers and DBAs working with MongoDB should look at how to optimize the performance of their database, especially nowadays with cloud services, where each byte processed, transmitted, and stored costs money. The ability to get started so quickly with MongoDB means that it is easy to overlook potential problems or miss out on simple performance improvements.

MongoDB best practice #1: Enable authorization and authentication on your database right from the start

The bigger the database, the bigger the damage from a leak. There have been numerous data leaks due to the simple fact that authorization and authentication are disabled by default when deploying MongoDB for the first time. While it is not a performance tip, it is essential to enable authorization and authentication right from the start as it will save you any potential pain over time due to unauthorized access or data leakage.

When you deploy a new instance of MongoDB, the instance has no user, password, or access control by default. In recent MongoDB versions, the default IP binding changed to 127.0.0.1 and a localhost exception was added, which reduced the potential for database exposure when installing the database.

However, this is still not ideal from a security perspective. The first piece of advice is to create the admin user and restart the instance again with the authorization option enabled. This prevents any unauthorized access to the instance.

To create the admin user:

```
> use admin
switched to db admin
> db.createUser({
...   user: "zelmar",
...   pwd: "password",
...   roles : [ "root" ]
... })
Successfully added user: { "user" : "zelmar", "roles" : [ "root" ] }
```

Then, you need to enable authorization and restart the instance. If you are deploying MongoDB from the command line:

mongod --port 27017 --dbpath /data/db --auth

Or if you are deploying MongoDB using a config file, you need to include:

security:

authorization: "enabled"

MongoDB best practice #2: Don’t use ‘not recommended versions’ or ‘end-of-life versions’ in production instances and stay updated

It should seem obvious, but one of the most common issues we see with production instances is due to developers running a MongoDB version that is actually not suitable for production in the first place. This might be due to the version being out of date, such as with a retired version that should be updated to a newer iteration that contains all the necessary bug fixes.

Or it might be due to the version being too early and not yet tested enough for production use. As developers, we are normally keen to use our tools’ latest and greatest versions. We also want to be consistent over all the stages of development, from initial build and test through to production, as this decreases the number of variables we have to support, the potential for issues, and the cost to manage all of our instances.

For some, this could mean using versions that are not signed off for production deployment yet. For others, it could mean sticking with a specific version that is tried and trusted. This is a problem from a troubleshooting perspective when an issue is fixed in a later version of MongoDB that is approved for production but has not been deployed yet. Alternatively, you might forget about that database instance that is “just working” in the background, and miss when you need to implement a patch.

In response to this, you should regularly check if your version is suitable for production using the release notes of each version. For example, MongoDB 5.0

provides the following guidance in its release notes: https://www.mongodb.com/docs/upcoming/release-notes/5.0/

mongodb warning

IDG

The guidance here would be to use MongoDB 5.0.11 as this version has the required updates in place. If you don't update to this version, you will run the risk of losing data.

While it might be tempting to stick with one version, keeping up with upgrades is essential to preventing problems in production. You may want to take advantage of newly added features, but you should put these features through your test process first. You want to see if they pose any problems that might affect your overall performance before moving them into production.

Lastly, you should check the MongoDB Software Lifecycle Schedules and anticipate the upgrades of your clusters before the end of life of each version: https://www.mongodb.com/support-policy/lifecycles

End-of-life versions do not receive patches, bug fixes, or any kind of improvements. This could leave your database instances exposed and vulnerable.

From a performance perspective, getting the right version of MongoDB for your production applications involves being "just right" — not so near the bleeding edge that you will encounter bugs or other problems, but also not so far behind that you will miss out on vital updates.

MongoDB best practice #3: Use MongoDB replication to ensure HA and check the status of your replica often

A replica set is a group of MongoDB processes that maintains the same data on all of the nodes used for an application. It provides redundancy and data availability for your data. When you have multiple copies of your data on

different database servers—or even better, in different data centers around the world—replication provides a high level of fault tolerance in case of a disaster.

MongoDB replica sets work with one writer node (also called the primary server). The best practice recommendation is to always have an odd number of members. Traditionally, replica sets have at least three instances:

Primary (writer node)

Secondary (reader node)

Secondary (reader node)

All of the nodes of the replica set will work together, as the primary node will receive the writes from the app server, and then the data will be copied to the secondaries. If something happens to the primary node, the replica set will elect a secondary as the new primary. To make this process work more efficiently and ensure a smooth failover, it is important for all the nodes of the replica set to have the same hardware configuration. Another advantage of the replica set is that it is possible to send read operations to the secondary servers, increasing the read scalability of the database.

After you deploy a replica set to production, it is important to check the health of the replica and the nodes. MongoDB has two important commands for this purpose:

rs.status() provides information on the current status of the replica set, using data derived from the heartbeat packets sent by the other members of the replica set. It's a very useful tool for checking the status of all the nodes in a replica set.

rs.printSecondaryReplicationInfo() provides a formatted report of the status of the replica set. It's very useful to check if any of the secondaries are behind the primary on data replication, as this would affect your ability to recover all your data in the event of something going wrong. If secondaries are too far behind the primary, then you could end up losing a lot more data than you are comfortable with.

However, note that these commands provide point-in-time information rather than continuous monitoring for the health of your replica set. In a real production environment, or if you have many clusters to check, running these commands could become time-consuming and annoying. Therefore we recommend using a monitoring system like Percona PMM to keep an eye on your clusters.

MongoDB best practice #4: Use $regex queries only when necessary and choose text search instead where you can

Sometimes the simplest way to search for something in a database is to use a regular expression or $regex operation. Many developers choose this option but in fact using regular expressions can harm your search operations at scale. You should avoid the use of $regex queries especially when your database is big.

A $regex query consumes a lot of CPU time and it will normally be extremely slow and inefficient. Creating an index doesn't help much and sometimes the performance is worse with indexes than without them.

For example, let's run a $regex query on a collection of 10 million documents and use .explain(true) to view how many milliseconds the query takes.

Without an index:

```
> db.people.find({"name":{$regex: "Zelmar"}}).explain(true)
- -   Output omitted  - -
"executionStats" : {
"nReturned" : 19851,
"executionTimeMillis" : 4171,
"totalKeysExamined" : 0,
"totalDocsExamined" : 10000000,
- -   Output omitted  - -
```

And if we created an index on “name”:

db.people.find({"name":{$regex: "Zelmar"}}).explain(true)

- - Output omitted - -

"executionStats" : {

"nReturned" : 19851,

"executionTimeMillis" : 4283,

"totalKeysExamined" : 10000000,

"totalDocsExamined" : 19851,

- - Output omitted - -

We can see in this example that the index didn’t help to improve the $regex performance.

It’s common to see a new application using $regex operations for search requests. This is because neither the developers nor the DBAs notice any performance issues in the beginning when the size of the collections is small and the users of the application are very few.

However, when the collections become bigger and the application gathers more users, the $regex operations start to slow down the cluster and become a nightmare for the team. Over time, as your application scales and more users want to carry out search requests, the level of performance can drop significantly.

Rather than using $regex queries, use text indexes to support your text search. Text search is more efficient than $regex but requires you to add text indexes to your data sets in advance. The indexes can include any field whose value is a string or an array of string elements. A collection can have only one text search index, but that index can cover multiple fields.

Using the same collection as the example above, we can test the execution time of the same query using text search:

```
> db.people.find({$text:{$search: "Zelmar"}}).explain(true)
- -   Output omitted  - -
"executionStages" : {
"nReturned" : 19851,
"executionTimeMillisEstimate" : 445,
"works" : 19852,
"advanced" : 19851,
- -   Output omitted  - -
```

In practice, the same query took four seconds less using text search than using $regex. Four seconds in “database time,” let alone online application time, is an eternity.

To conclude, if you can solve the query using text search, do so. Restrict $regex queries to those use cases where they are really necessary.

Article by:

Gayathri. M
20691A0534

Students Editors

REDDY ROHITH ENDULURI - 22691A05I1

SAI SANDEEP R - 22691A05J2

PRAVEEN KUMAR – 22691A05G5

II CSE

ACE

Association of Computer Engineers

Faculty Editors

Dr.R. Kalpana, Professor & HOD

Dr. R. Nidhya, Professor

G. Vasundhara Devi, Assistant Professor

***Contact**: ace_cse@mits.ac.in* ***Visit us**: www.mits.ac.in/cse*