Behind the Report:

Cost and Performance Advantages of Running AI On-Premises

Summary

This methodology report documents the procedures for conducting a production AI process as a workload for performance testing across both cloud and on-premises infrastructure configurations. The testing framework compares processing capabilities between a Microsoft Azure® cloud NC40ads H100 v5 instance and an on-premises Dell™ PowerEdge™ R7615 server (see Table 1 for details), both of which ran the software suite developed by Prowess Consulting for its proprietary Prowess LEAP™ process. Prowess Consulting used this workload as an example of a real-world production workload that uses both CPUs and GPUs for a blended analytical workload. Due to the proprietary nature of the software and source file content, those details have been excluded from this methodology report.

Table 1 | Select environmental details and differences

	On-Premises Dell™ PowerEdge™ R7615 Server	Microsoft Azure® NC40ads H100 v5 Instance
CPU	AMD EPYC™ 9334	AMD EPYC™ 9V84
CPU core count	32	40
CPU thread count	32/64 (AMD Simultaneous Multithreading [SMT] on)	40 (SMT off)
Memory	384 GB	320 GB
GPU	NVIDIA® L40S	NVIDIA® H100 NVL

Test Summary

The workload tested in this study is based on a real production pipeline used by Prowess Consulting to support marketing teams with AI-augmented content analysis. This process, known as the Prowess LEAP service, reflects the practical needs of enterprise marketing teams to generate deeper insights that inform content strategy.

The Prowess LEAP pipeline is divided into three primary phases, each targeting a distinct aspect of the AI workflow:

- **Phase 1 (CPU-centric)**: The first phase involves using traditional machine learning (ML) techniques to perform sentiment analysis and extract key topics from diverse marketing collateral. These operations rely heavily on CPU compute, and they are optimized for multithreaded performance and memory access.
- **Phase 2 (GPU-accelerated)**: The second phase harnesses AI to provide contextual analysis of the extracted topics. This stage is compute-intensive and benefits from high-throughput GPU acceleration.
- **Phase 3 (CPU-accelerated)**: The third phase is the actual analysis of the reults. This stage makes use of the CPU.

Together, these three phases provide a full-featured enterprise pipeline used to inform and guide marketing decisions. The hybrid nature of the workload—blending classical ML with AI—offers a representative benchmark for organizations seeking to evaluate their infrastructures' readiness for modern AI operations.

Test Setup

In addition to its primary compute resources, each environment had attached storage from which data was read, and it also ran a PostgreSQL® server, where the embedded data was stored.

In each environment, we ran two variations of the Python® scripts, one focusing on utilizing the CPU for tensor calculations and one using the GPU.

- **Test suite**:
  a. Python scripts:
     1. Topic modeling
     2. Sentiment analysis
     3. Prompt embedding
     4. Data embedding
     5. File-analysis subset
  b. Initial database tables:
     - sentiment_topic_input
     - selected_files_mix (a 2,500-file set and a 100,000-file set)
     - prompt_table

The on-premises hardware was tested with SMT both enabled and disabled.

Hardware Setup

This section contains the steps used to set up the hardware for both the on-premises and cloud testing environments.

On-Premises Hardware Setup

This section contains instructions for connecting to an on-premises hosted server via the Integrated Dell™ Remote Access Controller (iDRAC) interface and installing Windows Server® 2025:

1. Log in to the iDRAC console.
2. Navigate to the **Virtual Console** window to interact with the host.
3. To load the operating system installation media:
   a. Click **Virtual Media**.
   b. In the resulting window, click **Connect Virtual Media**.
   c. In the **Map CD/DVD** section, click **Browse**.
   d. Select the Windows Server 2025 installation ISO, and then click **Open**.
   e. Click **Map Device**.
   f. Click **Close**.
4. To boot the system to the operating system (OS) installation media:
   a. Click **Boot**.
   b. Select **Virtual CD/DVD/ISO**.
   c. Click **Yes** to confirm the boot option.
   d. At the top of the page, click the **Power** icon.
   e. Click **Power On System**.
   f. Click **Yes** to confirm the action.
5. To install Windows Server 2025:
   a. When prompted, press any key to boot into the ISO image.
   b. Press **Enter** to select the Windows® setup option.
   c. Leave the default selection of **English**, and then click **Next**.
   d. Leave the default selection of **US**, and then click **Next**.
   e. Leave the default selection of **Install Windows Server**.
   f. Select to confirm everything will be deleted, including files, apps, and settings.
   g. Click **Next**.
   h. Select the **Windows Server 2025 Datacenter (Desktop Experience)** image.
   i. Click **Next**.
   j. Click **Accept** to accept the license agreement.

   k. Select the blank disk on which Windows Server 2025 will be installed.
   l. Click **Next**.
   m. Click **Install**.
   n. Enter a password in the **Password** and **Reenter Password** fields.
   o. Click **Finish**, and then wait as the installation completes.
6. To log in to the desktop environment:
   a. Click the **Console controls** button.
   b. Press **Ctrl+Alt+Del** to access the login prompt.
   c. Enter the previously created password in order to log in.
7. To configure storage:
   a. Open the **Disk Management** control panel.
   b. To create the volume for file storage:
      i. Right-click the first unallocated volume, and then select **New Simple Volume**.
      ii. Click **Next**.
      iii. Leave the default size selection, and then click **Next**.
      iv. Assign a drive letter, and then click **Next**.
      v. Click **Next** to format the volume.
      vi. Click **Finish**.
   c. To create the volume for the PostgreSQL data storage:
      i. Right-click the first unallocated volume, and then select **New Simple Volume**.
      ii. Click **Next**.
      iii. Leave the default size selection, and then click **Next**.
      iv. Assign a drive letter, and then click **Next**.
      v. Click **Next** to format the volume.
      vi. Click **Finish**.
8. To confirm or change the SMT status on the CPU:
   a. Log in to the iDRAC interface.
   b. From the top menu, select **Configuration**.
   c. Select **BIOS Settings**.
   d. Click the dropdown for **Processor Settings**.
   e. From the **Logical Processors** dropdown, select **Enabled** or **Disabled**, as needed.
   f. Click **Apply**.
   g. Click **OK**.
   h. Click **Apply and Reboot**.
   i. Click **OK**.

Azure Hardware Setup

This section outlines the steps used to set up the hardware on the Azure environment.

1. Log in to Azure.
2. From the **Azure Services** section, select **Virtual machines**.
3. To start configuring the virtual machine (VM), click **Create**.
   a. Confirm the **Subscription** dropdown reflects the correct subscription.
   b. From the **Resource Group** dropdown, verify the resource group for the VM: **$RESOURCE_GROUP_NAME**
   c. Specify a **Name** for the VM.
   d. Select the **Region** for the VM.
   e. From the **Availability options** dropdown, select **No infrastructure redundancy required**.
   f. From the **Security Type** dropdown, leave the default selection of **Trusted launch virtual machines**.
   g. From the **Image** dropdown, specify **Windows Server 2025 Datacenter Azure Edition x64 Gen2**.
   h. Leave the default VM architecture selection of **x64**.
   i. From the **Size** dropdown, select **see all sizes**.
   j. Select the **N-Series expansion arrow**.
   k. Select the **NC40ads H100 v5 VM** size.
   l. Click **Select**.
   m. Specify the **Username** for the administrator account.

   n. Set the **Password** for the administrator account.
   o. Confirm the **Password** for the administrator account.
   p. Leave the default selection to allow Remote Desktop Protocol (RDP) for the **Inbound Port Rules**.
4. To configure the disks:
   a. Click **Next : Disks >**.
   b. Leave the default selections for the OS disk.
   c. To configure the data or PostgreSQL volumes:
      i. In the **Data disks for $RESOURCE_GROUP_NAME** section, click **Create**, and then attach a new disk.
      ii. Select the volume as a **Premium SSD 1024 GiB**.
      iii. Enter the **Name** as **files_volume** or **postgres_volume**.
      iv. Enter the **Source type** as **None**.
      v. Select **Delete disk with VM**.
      vi. Click **OK**.
   d. Repeat the prior step for the second volume.
   e. Click **Review + Create**.
   f. Click **Create**.
5. To view the VM details, click **Go To Resource**.
6. Note down the **Public IP address**.
7. Open a **Remote Desktop Connection** to the noted IP address with the specified username and password.
8. To configure the storage configuration:
   a. Open the **Disk Management** control panel.
   b. To create the volume for file storage:
      i. Right-click the first unallocated volume, and then select **New Simple Volume**.
      ii. Click **Next**.
      iii. Leave the default size selection, and then click **Next**.
      iv. Assign a drive letter, and then click **Next**.
      v. Click **Next** to format the volume.
      vi. Click **Finish**.
   c. To create the volume for the PostgreSQL data storage:
      i. Right-click the first unallocated volume, and then select **New Simple Volume**.
      ii. Click **Next**.
      iii. Leave the default size selection, and then click **Next**.
      iv. Assign a drive letter, and then click **Next**.
      v. Click **Next** to format the volume.
      vi. Click **Finish**.

Software Setup

This section outlines the steps taken to configure the software prerequisites for running the test suites. These steps apply to both the on-premises and Azure systems.

1. To install PostgreSQL:
   a. Download the Windows version of the PostgreSQL 17 installer from www.enterprisedb.com/downloads/postgres-postgresql-downloads.
   b. Open the resulting file.
   c. Click **Next**.
   d. Leave the default installation directory of **C:\Program Files\PostgreSQL\17**, and then click **Next**.
   e. Leave the default selections for components, and then click **Next**.
   f. Set the **Data Directory** to a folder in the PostgreSQL volume created earlier, and then click **Next**.
   g. Enter and confirm the password (hereafter referred to as **$DATABASE_PASSWORD**), and then click **Next**.
   h. Leave the default port selection, and then click **Next**.
   i. Leave the default local selection, and then click **Next**.
   j. Review the summary, and then click **Next**.
   k. Click **Next** again and wait as the installation proceeds.
   l. Click **Finish**.

2. To install Microsoft® Visual Studio® Code:
   a. Download the Visual Studio Code installer from https://code.visualstudio.com/docs/?dv=win64user.
   b. Open the resulting file.
   c. Click **OK**.
   d. Accept the license agreement, and then click **Next**.
   e. Leave the default installation path, and then click **Next**.
   f. To create a shortcut and continue, click **Next**.
   g. Leave **Add to path** selected, and then click **Next**.
   h. Click **Install**.
   i. Click **Finish**.
3. To install GPU drivers:
   a. Download the **572.83** driver from www.nvidia.com/en-us/drivers/details/242558/.
   b. Select the folder for the installation, and then click **OK**.
   c. Click **Agree and continue**.
   d. Click **Next**.
   e. Click **Close**.
4. To install C++ Visual Studio tools:
   a. From the "Thank You for Downloading Visual Studio Community Edition" web page, download Visual Studio.
   b. Open the resulting file.
   c. Click **Continue**.
   d. Select C++ desktop development workloads.
   e. Click **Install**, and then wait while the installation completes.
5. To install the CUDA toolkit:
   a. From "CUDA Toolkit 12.8 Downloads" web page, download the Windows Server 2025 local installer.
   b. Select the folder for the installation, and then click **OK**.
   c. Click **Agree and continue**.
   d. Click **Next**.
   e. Click **Next**.
   f. Click **Close**.
6. To install the Llama3:8b model:
   a. Download Ollama installer from https://ollama.com/download.
   b. Open the resulting file.
   c. Click **Install**.
   d. Open a PowerShell® terminal and run the following command:

```
Ollama pull llama3:8b
```

7. To install Python 3.12:
   a. Download **Windows Installer (64-bit) Python 3.12** from www.python.org/downloads/release/python-3129/.
   b. Open the resulting file.
   c. Select **Add Python to PATH**.
   d. Click **Install Now**.
   e. Click **Close**.
8. Reboot and log back in to the Windows Server environment to complete the software installations.

Test Configuration

This section outlines the steps needed to configure the files used with this workload. This section applies to both the on-premises and Azure configurations.

1. Populate the needed database backup files, raw data, and Python scripts, and then extract them into the **$SOURCE_FILES_PATH** folder on the system. Due to the proprietary nature of these elements, they are not included in this document.
2. To configure the Python environment:
   a. Open a PowerShell terminal.
   b. To change into the source files path directory, run the following command

```
cd $SOURCE_FILES_PATH
```

c. To create a Python virtual environment, run the following command:

```
python.exe -m venv venv
```

d. To activate the virtual environment, run the following command:

```
./venv/Scripts/activate.ps1
```

e. To install PyTorch®, run the following command:

```
pip install torch --index-url https://download.pytorch.org/whl/cu128
```

f. Create a text document, **requirements.txt**, with the following content:

```
accelerate==1.6.0
einops==0.8.1
emoji==2.14.1
flair==0.15.0
gensim==4.3.3
ninja==1.11.1.4
nltk==3.9.1
ollama==0.4.7
openpyxl==3.1.5
pandas==2.1.1
protobuf==6.30.2
psycopg2==2.9.10
PySocks==1.7.1
python-dotenv==0.21.0
rich==14.0.0
sentencepiece==0.2.0
shellingham==1.5.4
spacy>=3.7.0
transformers==4.38.0
```

g. To install Python dependencies, run the following command:

```
pip install -r requirements.txt
```

h. Create a text file for the Natural Language Toolkit (NLTK) requirements, **nltk_resources.py**, with the following content:

```
pythonimport nltk
nltk.download('vader_lexicon')
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt_tab')
import spacy
spacy.cli.download("en_core_web_sm")
```

i. To install, run the following command:

```
python .\nltk_resources.py
```

3. To configure the initial state of the PostgreSQL database:

a. Open a PowerShell terminal.

b. To change into the source files path directory, run the following command:

```
cd $SOURCE_FILES_PATH
```

c. To set the path and password environment variables, run the following command:

```
$pgPath = "C:\Program Files\PostgreSQL\17\bin\" $env:PATH += ";$pgPath" $env:PGPASSWORD = "$DATABASE_
PASSWORD"
```

d. To initialize the database, run the following command:

```
createdb.exe -U $DB_USERNAME -E UTF8 $DB_NAME
```

e. To create the prompt table, run the following command:

```
psql.exe -U $DB_USERNAME -d $DB_NAME -c "CREATE TABLE prompt_table ( id INTEGER PRIMARY KEY, prompt_name
TEXT, embeddings TEXT ); INSERT INTO prompt_table (id, prompt_name, embeddings) VALUES (1, 'Financial
education for youth', NULL);"
```

4. To create the .env values needed by the Python scripts, in both **$PATH_TO_CPU_SCRIPTS** and **$PATH_TO_GPU_SCRIPTS**, create a **text.env** file with the following content:

```
DB_HOST=localhost
DB_NAME=$DB_NAME
DB_USER=$DB_USERNAME
DB_PASSWORD=$DATABASE_PASSWORD
```

Running the Workloads

This section covers the running of both the CPU and GPU suites. These steps apply to both the on-premises and Azure environments.

These commands will:

1. Restore the database backups to a clean state, as the scripts update the tables.
2. Update the file location in the database backups to the local location.
3. Wait five minutes for the system resources to settle.
4. Initiate the running of the workload scripts.
5. Save the output and timing information into a time-stamped directory.

Update or set the following variables to their relevant local values:

- **$PATH_TO_{CPU|GPU}_SCRIPTS**: Directory containing CPU or GPU test suite scripts
- **$PATH_TO_FILES_TO_PARSE**: Directory containing source files for processing
- **$DB_USERNAME**: Database username
- **$DB_NAME**: Database name
- **$DATABASE_PASSWORD**: Database user password

1. To reset the databases and start a timed run of all five scripts, run the following commands in a PowerShell terminal:

```
$pgPath = "C:\Program Files\PostgreSQL\17\bin\"
$env:PATH += ";$pgPath"
$env:PGPASSWORD = "$DATABASE_PASSWORD"
$baseDir="$PATH_TO_{CPU|GPU}_SCRIPTS"
$startDir=Get-Location
$timestamp=Get-Date -Format "yyyyMMdd-HHmmss"
$runDir=Join-Path $startDir "run-$timestamp"
New-Item -ItemType Directory -Path $runDir | Out-Null
Set-Location $runDir
$combinedSql = "DROP TABLE IF EXISTS sentiment_topic_input;`nDROP TABLE IF EXISTS selected_files_mix;`n"
+ (Get-Content -Raw "$baseDir\sentiment_topic_input.sql") + "`n" + (Get-Content -Raw "$baseDir\selected_
files_mix.sql")
Set-Content -Path "combined_setup.sql" -Value $combinedSql
psql -U $DB_USERNAME -d $DB_NAME -f "combined_setup.sql"
psql -U $DB_USERNAME -d $DB_NAME -c "UPDATE selected_files_mix SET raw_file_path = '$PATH_TO_FILES_TO_
PARSE' || RIGHT(raw_file_path, POSITION('/' IN REVERSE(raw_file_path)) - 1);"
Write-Host "[$(Get-Date)] Database setup complete. Waiting 5 minutes before starting Python scripts..."
-ForegroundColor Yellow
Start-Sleep -Seconds 300
Write-Host "[$(Get-Date)] 5-minute wait complete. Starting script execution..." -ForegroundColor Yellow
$totalStopwatch=[System.Diagnostics.Stopwatch]::StartNew()
$scripts=@("$baseDir\Scripts\1-Topic_Modelling.py","$baseDir\Scripts\2-Sentimental_Analysis-cpu.
py","$baseDir\Scripts\3-Prompt_Embedding.py","$baseDir\4-Data_Embedding.py","$baseDir\Scripts\5-File_
Analysis.py")
$timings=@()
foreach ($script in $scripts) {
    $scriptFile=Get-Item $script
    $scriptName=$scriptFile.Name
    Write-Host "`n[$(Get-Date)] Starting: $scriptName" -ForegroundColor Green
    $sw=[System.Diagnostics.Stopwatch]::StartNew()
```

```
    $output=@()
    $process=Start-Process -FilePath python -ArgumentList $scriptFile.FullName -NoNewWindow -PassThru
-RedirectStandardOutput "$($scriptFile.BaseName)_output_temp.txt"
    $output=Get-Content "$($scriptFile.BaseName)_output_temp.txt"
    Set-Content -Path "$($scriptFile.BaseName)_output.txt" -Value $output
    Remove-Item "$($scriptFile.BaseName)_output_temp.txt" -ErrorAction SilentlyContinue
    $sw.Stop()
    $elapsedTime=$sw.Elapsed.TotalSeconds
    Write-Host "[$(Get-Date)] Completed: $scriptName in $elapsedTime seconds" -ForegroundColor Green
    $timings+=[PSCustomObject]@{Script=$scriptName; TimeInSeconds=$elapsedTime}
}
$totalStopwatch.Stop()
$totalTime=$totalStopwatch.Elapsed.TotalSeconds
$timings | ForEach-Object { "$($_.Script): $($_.TimeInSeconds) seconds" } | Add-Content -Path ".\
execution_times.txt"
Add-Content -Path ".\execution_times.txt" -Value "Total execution time: $totalTime seconds"
Set-Location $startDir
Write-Host "`n[$(Get-Date)] All scripts completed in $totalTime seconds" -ForegroundColor Yellow
Write-Host "Results saved in: $runDir" -ForegroundColor Yellow
```

Appendix: Variable Summary

The following variables are used throughout this methodology document:

Variable	Description	Example Value
$OS_ADMIN_USERNAME	Windows Server® administrator username	Admin
$OS_ADMIN_PASSWORD	Windows Server administrator password	*****
$DB_USERNAME	Database username	PostgreSQL
$DATABASE_PASSWORD	Database administrator password	*****
$PATH_TO_FILES	Directory on the files volume that will house the scripts, database backups, and raw files	F:\testing_assets\
$PATH_TO_FILES_TO_PARSE	Directory containing source files for processing in the data embedding script	F:\testing_assets\source_files\
$PATH_TO_CPU_SCRIPTS	Directory containing CPU test suite scripts	F:\testing_assets\cpu_suite\
$PATH_TO_GPU_SCRIPTS	Directory containing GPU test suite scripts	F:\testing_assets\gpu_suite\

0725/250013