Generative AI: Data Platform Architecture

IEEE-CNSV
Milpitas, CA – Mar 12, 2024

Presented By:
Prasad Venkatachar- Sr Director Products & Solutions @Pliops
Data Architecture in Gen AI Era

Customer Segments
- Retail
- Insurance
- FSI
- Telecom
- Retail

Technologies & Certifications
- Databases & Analytics
- AI/Gen AI
- Cloud
- Storage
- Product Management Stanford
- ITIL/ITSM

Roles
- Database Admin
- Solution Architect
- Product Management
- Product Marketing

Countries
- US
- India
- Aus & Italy
- UK

Countries
- UK
- India
- Aus & Italy
- US

Companies
- SanDisk
- Hewlett Packard Enterprise
- i2
- Lenovo
- PLiOPS

Education
- Engineering
- Diploma
- MBA

Prasad Venkatachar

ANN Profile - Introduction

Technologies & Certifications
- Databases & Analytics
- AI/Gen AI
- Cloud
- Storage
- Product Management Stanford
- ITIL/ITSM

Roles
- Database Admin
- Solution Architect
- Product Management
- Product Marketing

Countries
- US
- India
- Aus & Italy
- UK

Companies
- SanDisk
- Hewlett Packard Enterprise
- i2
- Lenovo
- PLiOPS

Education
- Engineering
- Diploma
- MBA
Topics

• Data Platform Considerations
• Industry Verticals & Enterprise Functions
• Google AlloyDB Omni Solution
• How do I use it for Business Applications
  • Biz Apps: E-Commerce (Transaction Store)
  • Business Analytics
• Gen AI Intro & Adoption
• Vector Databases & RAG
• RAG Demo
Next Data Platform Consideration

**Seamless Data Movement**
- Edge
- Core
- Cloud

**One Technology**
- Biz Apps: OLTP
- Biz- Real Time Analytics
- Gen AI Apps: Vector Databases

**Robust Support**
- Enterprise Support

**Open Source**
- Flexibility to Switch

“71% of respondents in the Data and AI Trends Report plan to use databases integrated with gen AI capabilities.”

Google 2024 Survey
Traditional Apps to Gen AI Apps Adoption

**Industry Verticals**
- Retail
- Banking & FSI

**Content & Media**

**Enterprise Functions**
- HR & Finance
- Marketing
- Customer Support
One Platform: OLTP/Analytics/Gen AI

E-Commerce Applications
- Transaction Database

Real Time Biz Analytics
- Columnar Engine

Gen AI Chatbots
- Vector Database

AlloyDB Omni

Lenovo Edge Server

XDP Data Accelerator + SSDs

Lenovo Datacenter Server

PLiOPS EXTREME DATA PROCESSOR

Google Cloud
Online Transaction Processing System

# Large number of users
1K – 1M users

- Online Web Users
- Mobile Users

Simple & Short duration Transaction
1 Row Insert: 2 Sec

Transaction Complete: User notified
Order Complete: 5 to 10 Seconds

Failover to Standynode: Primary Server/DB Failure
< Secs - Minutes

Server 1
Database 1

Server 2
Database 2

Total Transaction per Second
4X PostgreSQL Performance: for Transactional Workloads

• Process more Transaction Requests
• 2X Higher Transaction Requests Postgres SQL to AlloyDB Omni
• Upto 4X Transaction Requests from PostgreSQL to AlloyDB Omni with Pliops & Lenovo

• Serve more Web and Mobile users
User Experience:
Average & Tail Latency Reduction
AlloyDB Columnar Engine: Implementation & Benefits

Real-time business insights

Columnar Engine Memory = 36GB

<table>
<thead>
<tr>
<th></th>
<th>customer</th>
<th>lineitem</th>
<th>part</th>
<th>supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>block_count_in_cc</td>
<td>365550</td>
<td>13474829</td>
<td>419205</td>
<td>22643</td>
</tr>
<tr>
<td>Total Block Count</td>
<td>365550</td>
<td>13474829</td>
<td>419205</td>
<td>22643</td>
</tr>
<tr>
<td>Column</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

Row Store = 512GB

<table>
<thead>
<tr>
<th></th>
<th>Block Count cc</th>
<th>Total Block Count cc</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>lineitem</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>part</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>partsupp</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>supplier</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

google_columnar_engine.relations='TPC-H. Customer',
columnar_engine.memory_size_in_mb=36GB

Populating Tables

Recommend Col Memory Sizing
AlloyDB Omni: Row vs Columnar Execution

Google tests show 100X performance improvements, however, these tests are highly stringent with identical memory setup for rows & columns.

Query 4 has 16X Performance Improvement

Query 6 has 26X Performance Improvement

Average 4X Performance improvement per Query level
AI VS Gen AI Life Cycle

**AI Model Life Cycle**
- Data Preparation
- Build & Train Model
- Validate Model
- Deploy Model
- Monitor Model

**Gen AI Model Life Cycle**
- Choose LLM
- Customize LLM
- Consume LLM
- Monitor Model
Gen AI: Journey & Skill Set

- Build & Train Model From Scratch
- Fine Tuning LLM Model
- RAG (Retrieval Augmented Generation)
- Prompt Engineering

20 M Developers
20K ML Engineers
2000 LLM Engineers

Skill Set/Effort & Cost
What do you think it’s going wrong here?

Automobile Deploys Chatbot for Customer Support

The user Ask a Python Script

Chatbot provides Response
Prompt Engineering

Basic – Prompt Structure

What you want to become: Journalist
Write journal
Climate Change
Create Awareness
Boundaries—500 words

Role
Outcome
Context
Intent
Constraint

Advanced – Prompt Structure

Zero-Shot prompting
Few Shot Learning
Chain of Thought
Tree of Thought

“n” Examples
Example

Example

Treat it as you’re bringing up Your Child

Example
Few Shot Example

```
prompt = f"""
Example1: The GPU has a TF32 Tensor core is less than 989 Tera Flops. Classify the GPU as BASIC or ADVANCED? Answer: BASIC
Example2: The GPU has a TF32 Tensor cores is greater than 1979 Tera Flops. Classify the GPU as BASIC or ADVANCED? Answer: ADVANCED
Example3: The GPU has a TF32 Tensor Cores is 800 Tera Flops. Classify the GPU as BASIC or ADVANCED. Answer:
Example4: The GPU has a TF32 Tensor Cores is 2200 Tera Flops. Classify the GPU as BASIC or ADVANCED. Answer:
"""
response = get_completion(prompt)
print(response)

Example3: BASIC
Example4: ADVANCED
```
Chain of Thought

system_message = f""""Answer the customer queries based on the Database product Performance info below.

1. Product: PostgreSQL
   Performance: 200
   Latency: 400

2. Product: AlloyDB Omni
   Performance: 400
   Latency: 200

3. Product: AlloyDB Omni with Lenovo and Pliops
   Performance: 800
   Latency: 100
"""

user_message = f""""by how much is the AlloyDB Omni with Lenovo and Pliops more performance than the PostgreSQL"""

messages = [
    {'role': 'system',
    'content': system_message},
    {'role': 'user',
    'content': f'{user_message}'},
]

response =
get_completion_from_messages(messages)
print(response)

The AlloyDB Omni with Lenovo and Pliops has 4 times more performance than PostgreSQL. This can be calculated by dividing the performance of AlloyDB Omni with Lenovo and Pliops (800) by the performance of PostgreSQL (200), which equals 4.
## LLM Accuracy for Prompt Engineering

<table>
<thead>
<tr>
<th></th>
<th>Claude 3 Opus</th>
<th>Claude 3 Sonnet</th>
<th>Claude 3 Haiku</th>
<th>GPT-4</th>
<th>GPT-3.5</th>
<th>Gemini 1.0 Ultra</th>
<th>Gemini 1.0 Pro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undergraduate level knowledge MMLU</td>
<td>86.8%</td>
<td>79.0%</td>
<td>75.2%</td>
<td>86.4%</td>
<td>70.0%</td>
<td>83.7%</td>
<td>71.8%</td>
</tr>
<tr>
<td>Graduate level reasoning GPTQA, Diamond</td>
<td>50.4%</td>
<td>40.4%</td>
<td>33.3%</td>
<td>35.7%</td>
<td>28.1%</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Grade school math GSM8K</td>
<td>95.0%</td>
<td>92.3%</td>
<td>88.9%</td>
<td>92.0%</td>
<td>57.1%</td>
<td>94.4%</td>
<td>86.5%</td>
</tr>
<tr>
<td>Math problem-solving MATH</td>
<td>60.1%</td>
<td>43.1%</td>
<td>38.9%</td>
<td>52.9%</td>
<td>34.1%</td>
<td>53.2%</td>
<td>32.6%</td>
</tr>
<tr>
<td>Multilingual math MGSM</td>
<td>90.7%</td>
<td>83.5%</td>
<td>75.1%</td>
<td>74.5%</td>
<td>—</td>
<td>79.0%</td>
<td>63.5%</td>
</tr>
<tr>
<td>Code HumanEval</td>
<td>84.9%</td>
<td>73.0%</td>
<td>75.9%</td>
<td>67.0%</td>
<td>48.1%</td>
<td>74.4%</td>
<td>67.7%</td>
</tr>
<tr>
<td>Reasoning over text DROP, Fi score</td>
<td>83.1</td>
<td>78.9</td>
<td>78.4</td>
<td>80.9</td>
<td>64.1</td>
<td>82.4</td>
<td>74.1</td>
</tr>
<tr>
<td>Mixed evaluations IRB-Bench-Hard</td>
<td>86.8%</td>
<td>82.9%</td>
<td>73.7%</td>
<td>83.1%</td>
<td>66.6%</td>
<td>83.6%</td>
<td>75.0%</td>
</tr>
<tr>
<td>Knowledge Q&amp;A ARC-Challenge</td>
<td>96.4%</td>
<td>93.2%</td>
<td>89.2%</td>
<td>96.3%</td>
<td>85.2%</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Common Knowledge HellaSwag</td>
<td>95.4%</td>
<td>89.0%</td>
<td>85.9%</td>
<td>95.3%</td>
<td>85.5%</td>
<td>87.8%</td>
<td>84.7%</td>
</tr>
</tbody>
</table>

Source: https://www.anthropic.com/news/claud... These Ranking keeps Evolving with new model release
### Localization & Domain-Specific Questions

- **where is levis stadium located in california**
  
  I cannot provide location information as I am unable to access real-time location data. For the most up-to-date information on the location of Levi's Stadium, please refer to official stadium websites or reputable news sources.

- **What are the popular items for sale in Walmart in Milpitas**
  
  I cannot access real-time information, therefore I cannot provide a list of popular items for sale in Walmart in Milpitas. For the most up-to-date information on Walmart products and sales, please refer to their official website or other reputable news sources.

- **How is the mortgage rates as of Feb 2024**
  
  I cannot access real-time information, therefore I cannot provide a mortgage rate as of February 2024. For the most up-to-date information on mortgage rates, please refer to reputable financial news sources.
RAG (Retrieval Augmented Generation)

- Provide information Sources to develop Knowledge
  - Read this article
  - Read this book
  - Refer to this set of books
  - Read from this source

- Process to develop Knowledge from external sources
  - Chunking
  - Slicing the data
  - Embedding
  - Vectorize [0.4,0.1]
  - Store & Index
  - Store distance between vectors
  - User Response
  - Serve Users with new Knowledge

- Remember to treat you are bringing up Your Child
  - To provide specific/domain Knowledge
  - LLM knowledgebase is not up to date

- PLiOPS Extreme Data Processor
AlloyDB AI – Vector Database

Input Data Sources
- PDF
- Doc
- XLS
- Json
- URLs

Chunking & Embedding
- Chunk 1
- Chunk 2
- Chunk 3
- Chunk 4

Vector Store
- AlloYDB - AI Vector Store
- [0.3, 0.2, 0.1, 0.4]

Indexing
- Langchain, Lamaindex

Data Ingestion

Data Querying (Retrieval)

Response

Domain Specific
- LLM
Vector Search: Find Most Similar Embeddings

L2_distance(vector1, vector2)

Squared Euclidean (L2 Squared)
\[ \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \]

L1_distance(vector1, vector2)

Manhattan (L1)
\[ \sum_{i=1}^{n} |x_i - y_i| \]

cosine_distance(vector1, vector2)

Cosine Distance
\[ 1 - \frac{A \cdot B}{||A|| \cdot ||B||} \]

Inner_product(vector1, vector2)

Dot Product
\[ A \cdot B = \sum_{i=1}^{n} A_i B_i \]

Source: Deep Learning.ai
Postgres Vector Indexes

IVF (Inverted File Index)

Hierarchical Navigable Small Worlds (HNSW)

- m: Maximum number of connections per layer
- Ef_construction: Size of Dynamic list for construction graph

Number/size of the lists

Search: Number of lists to be verified
RAG Deployment: Decision Factors

Cost of Querying = Dimensions * Queries

Cost of Indexing = Dimensions * Writes

Storage Cost = Record Count * Dimensions

Search Latency = Dimensions * Index Performance

Vector Database
RAG Evolution: Basic/Advanced/Modular

Source: Retrieval-Augmented Generation for Large Language Models
RAG Use cases & Advantages

Chatbots
Searching for similar content (Text, Image, Video)
Personalized recommendation
Detecting Anomalies

Advantages
• Reduce Hallucinations
• Enable a LLM to cite Sources
• Solve Knowledge Intensive Tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>w/o RAG</th>
<th>w/ RAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-4-Turbo</td>
<td>0.700</td>
<td>0.835</td>
</tr>
<tr>
<td>GPT-3.5-Turbo</td>
<td>0.669</td>
<td>0.804</td>
</tr>
<tr>
<td>Mixtral-8×7B</td>
<td>0.583</td>
<td>0.808</td>
</tr>
<tr>
<td>Llama-2-70b</td>
<td>0.609</td>
<td>0.760</td>
</tr>
</tbody>
</table>

Gemini Pro

Source: Google
Vector Database Landscape

- **On-Premise (Self Hosted)**
  - redis
  - Weaviate
  - AlloyDB Omni
  - Postgres
  - pgvector
  - MongoDB

- **Cloud Native (PaSS)**
  - chroma
  - LanceDB
  - Astra DB
  - LanceDB
  - Pinecone
  - AlloyDB

- **In-Memory Data**
  - FAISS
  - MemoryStore

- **Serverless - Cloud**
  - Azure Cosmos DB

List is not Exhaustive
Fine Tuning

Treat it as your bringing up Your Own Child

Obtain Base Model
- Not Tuned

Fetch Curated Dataset
- Human Response

Load LORA
- PEFT

Quantization
- 4/8/16 Bit

PEFT

Serve
- Serve the Fine Tuned Model

Training
- Train the Model

Embedding
- Learning Rate

Load LORA
Build & Train Model from Scratch

Conversational Forums Pipeline

Dataset for Training

Source: *Dolma an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research*
HammerDB Benchmark: AlloyDB Accelerated by Pliops XDP

4X Vs PostgreSQL w/ Pliops XDP

AlloyDB (Postgres) Demo
Thank You
Backup
Steps Required to perform Task

The role of a connected data platform from Edge-Core-Cloud is becoming more crucial as organizations gather an ever-increasing volume of data from IOT devices, customer transactions and third-party sources. Data modernization initiatives can prove to be a game changer for retail enterprises to efficiently store and process the data at the edge, data centers and Cloud. AlloyDB Omni solution from Google, Lenovo, Pliops is designed and developed to serve retail customer data modernization needs.

E-Commerce Acceleration: Make shopping experience faster and smoother.

Completion for prompt 1:
1 - Retail enterprises can benefit from the AlloyDB Omni solution from Google, Lenovo, and Pliops to efficiently store and process data for data modernization initiatives, ultimately improving the shopping experience for customers.
2 - Les entreprises de vente au détail peuvent bénéficier de la solution AlloyDB Omni de Google, Lenovo et Pliops pour stocker et traiter efficacement les données dans le cadre d'initiatives de modernisation des données, améliorant ainsi l'expérience d'achat pour les clients.
3 - Retail customers benefit from faster and smoother shopping experiences due to the efficient storage and processing of data provided by the solution.
Pre-Training Llama 7B LLM Single GPU

- Llama 7B Large Language Model Training
- Single consumer-grade GPU (RTX 4090) 24GB
- Gradient Low-Rank Projection
- Gradient weight matrix as low rank without performance degradation
- 82.5% reduction in memory for storing optimizer states during training.

Memory-Efficient LLM Training by Gradient Low-Rank Projection

Training Large Language Models (LLMs) presents significant memory

9:41 PM • Mar 6, 2024 • 291.7K Views
## Vector Database Comparison

<table>
<thead>
<tr>
<th>Vector Database</th>
<th>Supports Deployment in Current Database</th>
<th>Size of Vector Dimensions</th>
<th>Aggregations</th>
<th>Queries per second ANN Benchmarks</th>
<th>Metadata Filtering</th>
<th>Time Based Metadata Filtering</th>
<th>Time-Series Compression</th>
<th>Hybrid Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>pgvector on PostgreSQL</td>
<td>Yes</td>
<td>16000</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>(Supports Postgres date data types)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AlloyDB/Omni – Vector for Postgres</td>
<td>Yes</td>
<td>16000</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>(Supports Postgres date data types)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Qdrant</td>
<td>No</td>
<td>-</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Somewhat (Need to convert time to an integer)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ChromaDB</td>
<td>No</td>
<td>-</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Somewhat (Need to convert time to an integer)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>KDB.AI</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>(datetime64, timedelta64) (Need to future)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Weaviate</td>
<td>No</td>
<td>65535</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>(Supports ‘date’ data type) (Sparse-Dense Vectors)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Pinecone</td>
<td>No</td>
<td>20000</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Somewhat (Need to convert date/time to integer in Unix time) (Sparse-Dense Vectors)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Milvus</td>
<td>No</td>
<td>34768</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Somewhat (Need to convert date/time to integer in Unix time) (Sparse-Dense Vectors)</td>
<td>No, they use the phrase “Hybrid Search”, but it really means metadata filtering</td>
<td>No</td>
</tr>
</tbody>
</table>
## Open Source Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Llama 2</td>
<td>7B</td>
<td>3.8GB</td>
</tr>
<tr>
<td>Mistral</td>
<td>7B</td>
<td>4.1GB</td>
</tr>
<tr>
<td>Dolphin Phi</td>
<td>2.7B</td>
<td>1.6GB</td>
</tr>
<tr>
<td>Phi-2</td>
<td>2.7B</td>
<td>1.7GB</td>
</tr>
<tr>
<td>Neural Chat</td>
<td>7B</td>
<td>4.1GB</td>
</tr>
<tr>
<td>Starling</td>
<td>7B</td>
<td>4.1GB</td>
</tr>
<tr>
<td>Code Llama</td>
<td>7B</td>
<td>3.8GB</td>
</tr>
<tr>
<td>Llama 2 Uncensored</td>
<td>7B</td>
<td>3.8GB</td>
</tr>
<tr>
<td>Llama 2 13B</td>
<td>13B</td>
<td>7.3GB</td>
</tr>
<tr>
<td>Llama 2 70B</td>
<td>70B</td>
<td>39GB</td>
</tr>
<tr>
<td>Orca Mini</td>
<td>3B</td>
<td>1.9GB</td>
</tr>
<tr>
<td>Vicuna</td>
<td>7B</td>
<td>3.8GB</td>
</tr>
<tr>
<td>LLaVA</td>
<td>7B</td>
<td>4.5GB</td>
</tr>
<tr>
<td>Gemma</td>
<td>2B</td>
<td>1.4GB</td>
</tr>
<tr>
<td>Gemma</td>
<td>7B</td>
<td>4.8GB</td>
</tr>
</tbody>
</table>