Database Management System Extensibility

Abi Kim

Marco Slot*, Dave Andersen, Andy Pavlo

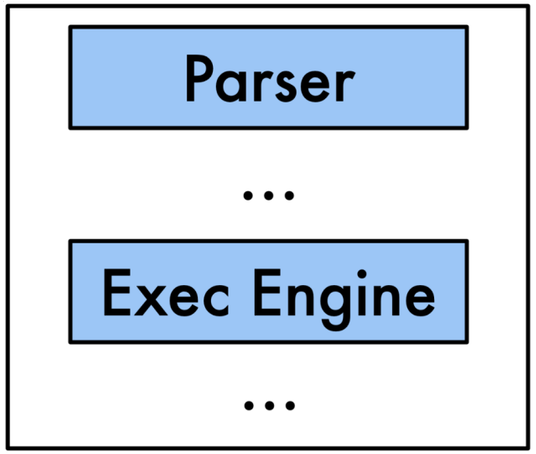
PARALLEL DATA LABORATORY

Carnegie Mellon University, *Crunchy Data

Carnegie Mellon
Parallel Data Laboratory

So you want to find a DBMS for your use case…

Hmm.. existing DBMSs don't work for our special new use case.



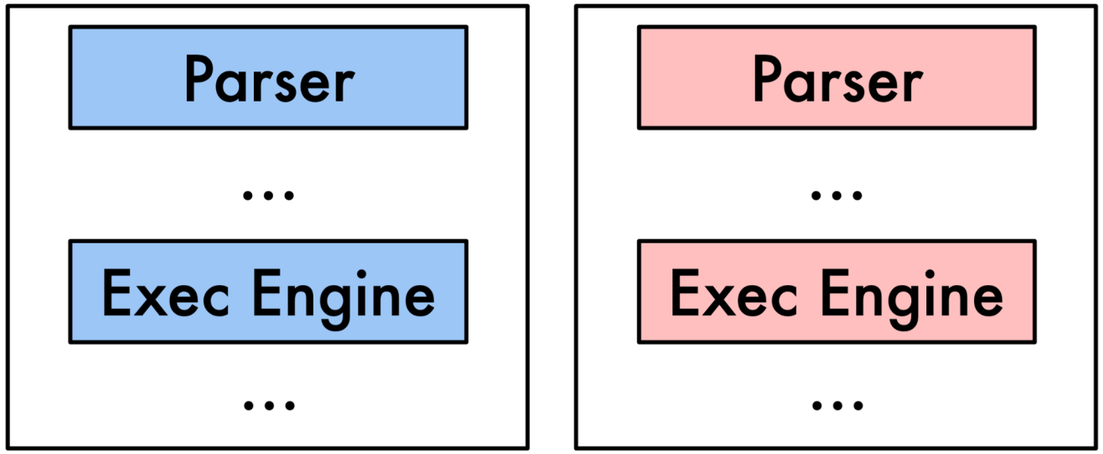
Existing DBMS

(app w/GPT3 backend)

(let's find a vector DBMS!)

POV: You want to find a DBMS for your use case…

Let's rewrite a new DBMS from scratch instead!



Existing DBMS New DBMS

PINECONE

Weaviate

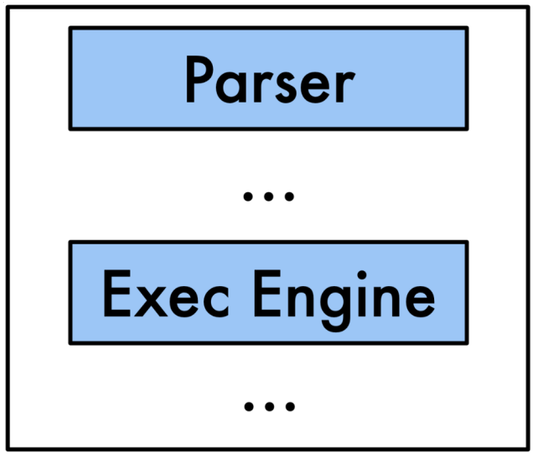
FeatureBase

Milvus

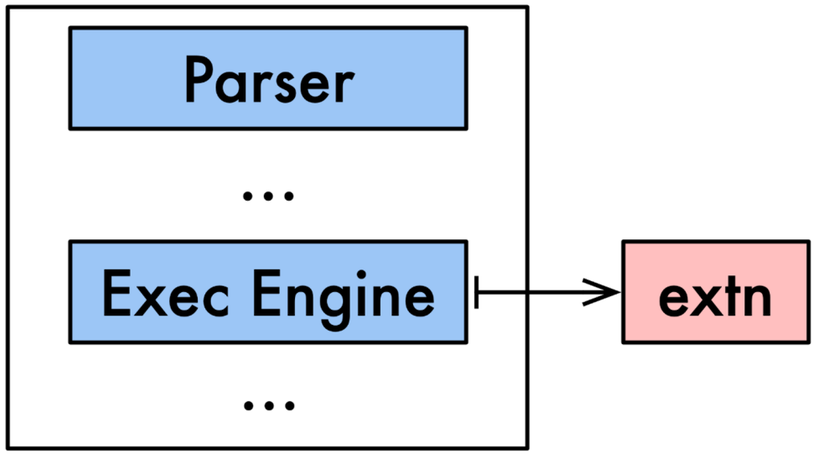
(and many more)

POV: You want to find a DBMS for your use case…

On the other hand, we could write an extension!

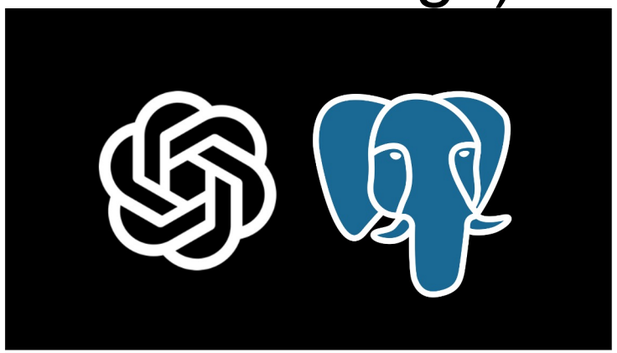


Existing DBMS



Existing DBMS with Extension

pgvector (LLM embeddings)



DBMS Extensibility

- A feature where the DBMS allows developers to expand the system's capabilities with custom software
- Examples:
  - math2 (math function library)
  - pgvector (LLM embedding user-defined type)
  - pg_stat_statements (query statistics collector)

DBMS Extensibility Benefits

Developing a DBMS extension:

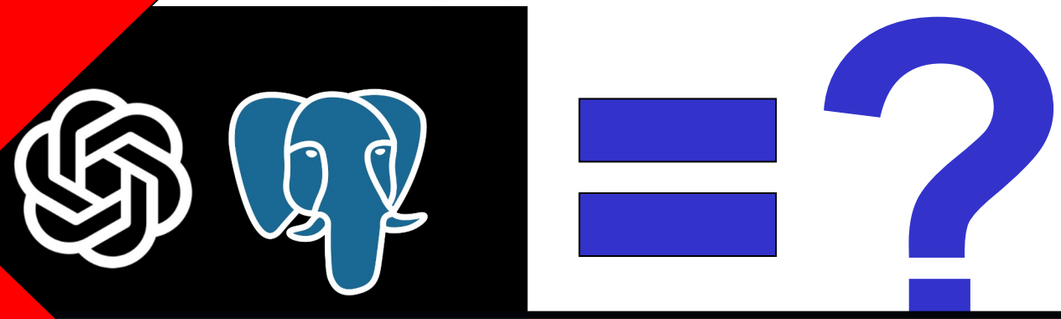
- Streamlined developer efforts 
- Engineers create new features easily 
- Lets users customize DBMS 

Customizability

- Lets users add multiple extensions to one instance

Distributed PostgreSQL
(as an extension)

tor
beddings)



pgvector with citus could r termine pe of parameter $1 #7078

Open GPF199541 opened this issue on J ients

Extensibility Research Overview

Current field of extensibility:

- Not well-studied ✗
- Design problems ✗

My research:

- Taxonomize extensibility ✓
- Find and quantify design problems ✓

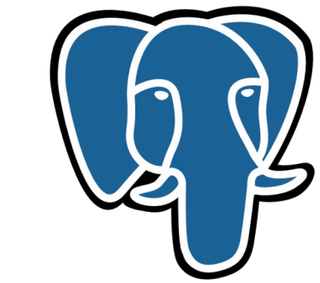
Eventual Goal

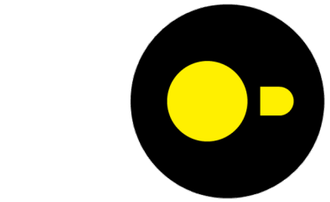
Design extensibility well ➡ People use it to extend DBMSs

Our Extensibility Taxonomy

- Survey existing DBMSs and extensibility literature
- Categorize different extensions and different extensibility design decisions

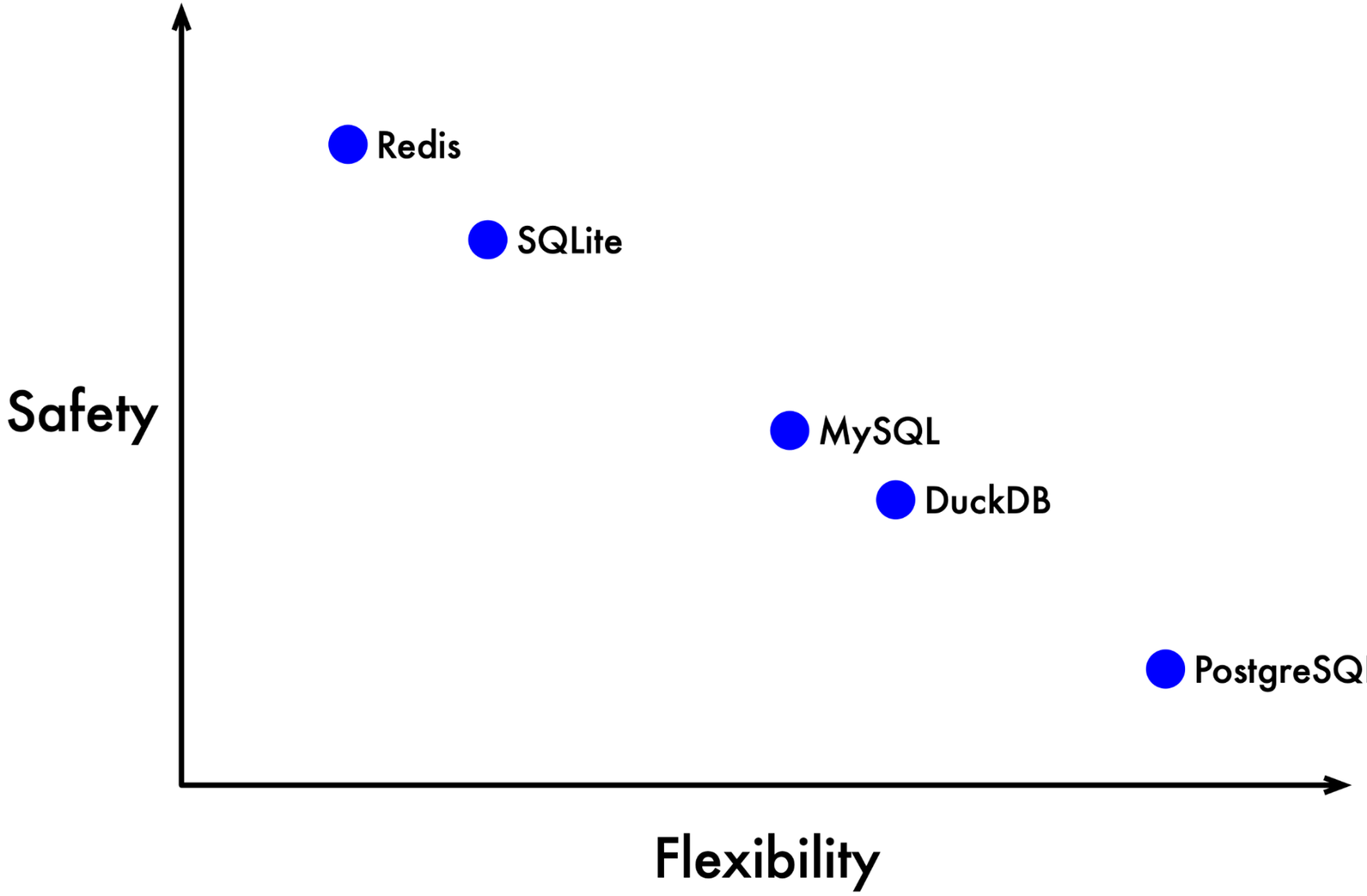
If you're curious about this, see me at my poster!





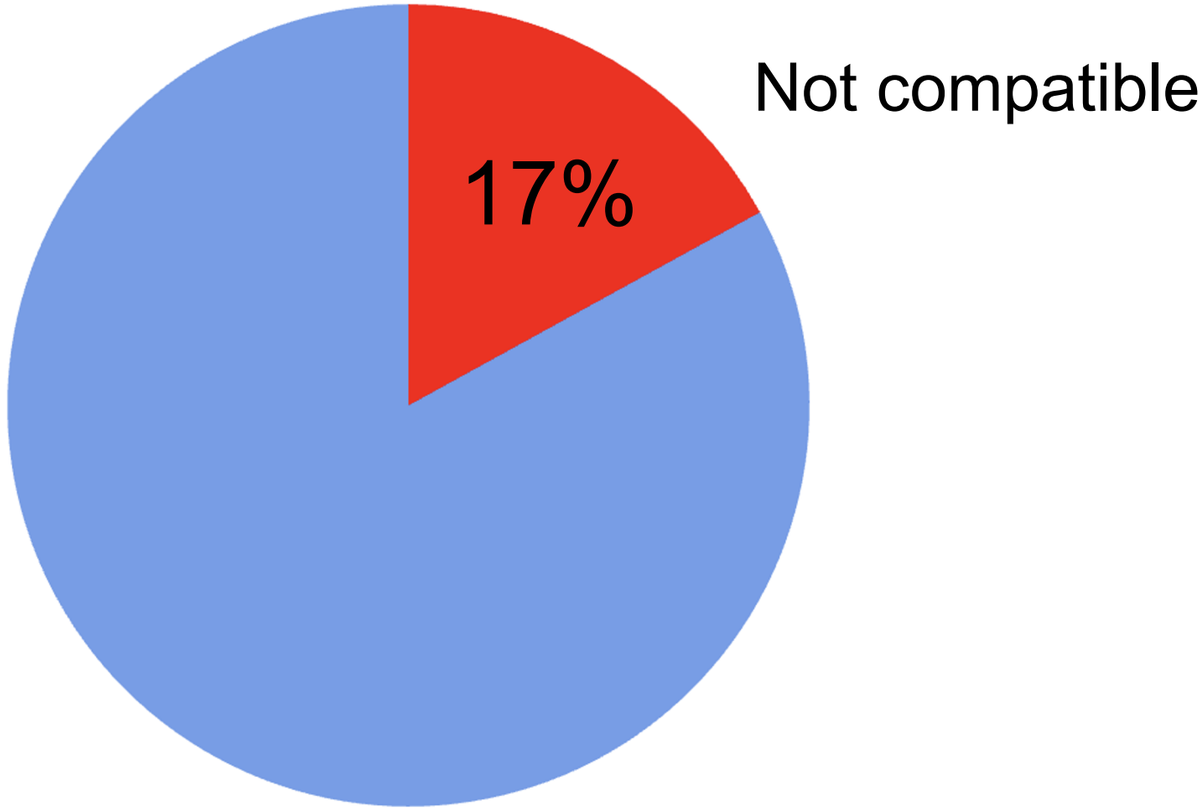

Safety vs. Flexibility Tradeoff

Safety: possibility of causing an extension to break the system
Flexibility: feasibility of developing extension that suits your needs



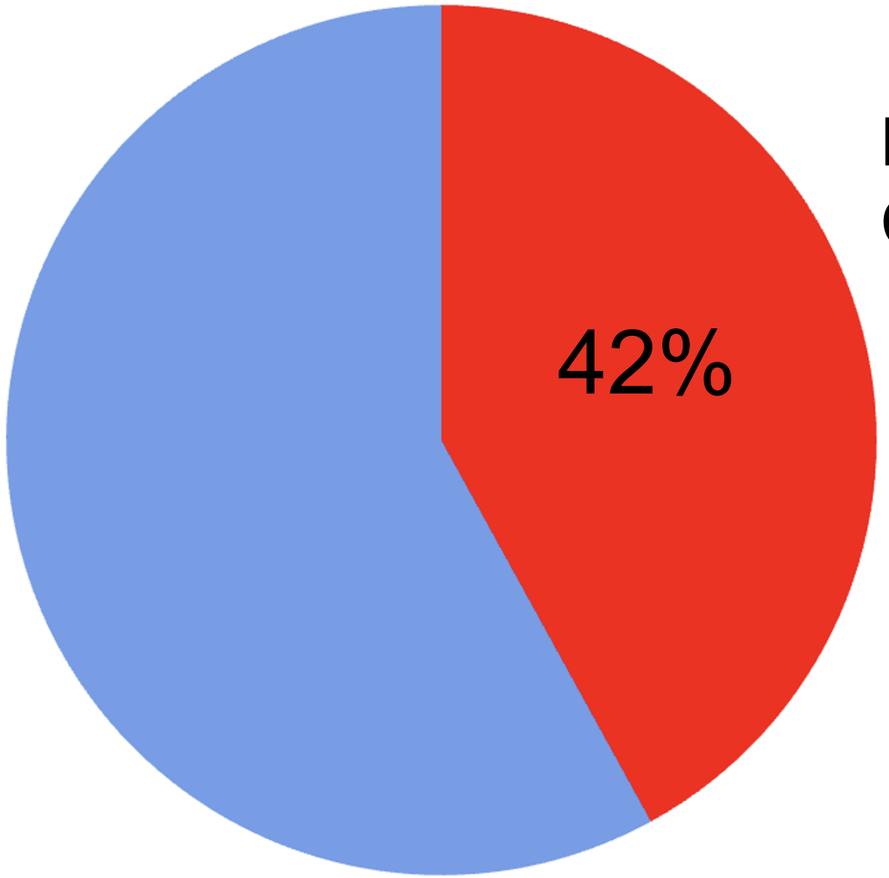
Design Problem: Compatibility

- No guarantees of extension **correctness/safety when installed with other extensions**



Design Problem: Development

- **Duplicated code** and **lack of infrastructure** are problems



Duplicated Code from Core DBMS

Our end goal: No one will ever rewrite a DBMS from scratch.

Our Ideas

- Safe and compatible extensibility
  - Program analysis techniques, verification
- Reduce complexity of writing extensions
- Make extensions portable between different DBMSs and versions
  - 45% of extensions use version macros

Takeaways

- Extensions are powerful, but not widely used
- DBMS extensibility field has design flaws
- Extensibility is the solution to the current DBMS landscape

See my poster for details on my research!