Supporting Our AI Overlords: Redesigning Data Systems to be Agent-First

Shu Liu, Soujanya Ponnapalli, Shreya Shankar, Sepanta Zeighami, Alan Zhu Shubham Agarwal, Ruiqi Chen, Samion Suwito, Shuo Yuan, Ion Stoica, Matei Zaharia Alvin Cheung, Natacha Crooks, Joseph E. Gonzalez, Aditya G. Parameswaran

UC Berkeley

SAA'25 (Systems for Agentic AI)

Reporter: Fanglei Shu

Author : Aditya Parameswaran

https://people.eecs.berkeley.edu/~adityagp/

Projects:

- *LLM-powered data tooling*
- *Data systems both with and for LLMs*
- *Simplifying Data Science at Scale*

Papers:

- [SIGMOD' 26] TWIX: Automatically Reconstructing Structured Data from Templatized Documents.
- [SIGMOD' 26] Cut Costs, Not Accuracy: LLM-Powered Data Processing with Guarantees
- [VLDB' 25] DocETL: Agentic Query Rewriting and Evaluation for Complex Document Processing
- [ICLR' 25] Towards Accurate and Efficient Document Analytics with Large Language Models NUDGE: Lightweight Non-Parametric Fine-Tuning of Embeddings for Retrieval

Outline

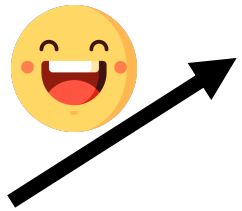
- *Background*
- *Design*
- *Conclusion*

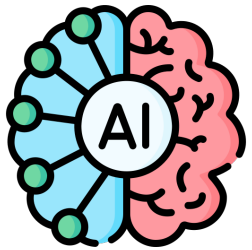
Outline

- **Background**
- Design
- Conclusion

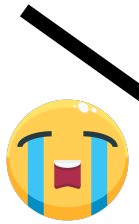
Background

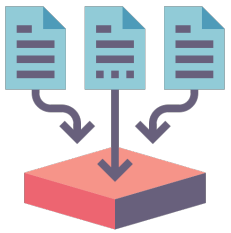
Agentic System





能够匹配人类推理能力

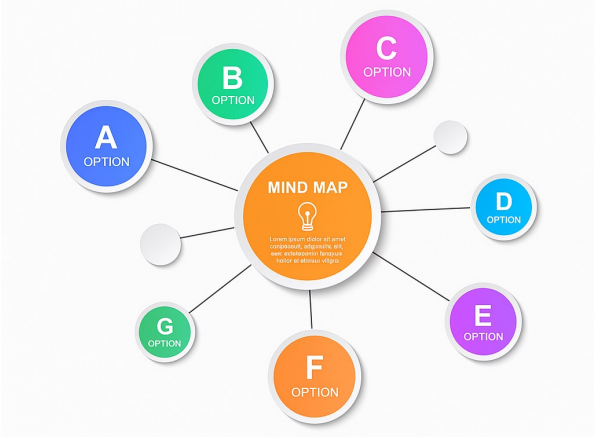




缺乏对底层数据以及底层数据系统的感知

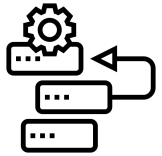
Agentic Speculation（智能体推测）：

- 不断探索、尝试各种可能的解决方案来弥补这种缺陷
- 通过高吞吐量的探索性查询来寻找最佳行动方案



Background

智能体推测与传统数据系统工作负载的区别：



Traditional workloads

- 间歇性：用户或用户操作工具间歇的发起请求
- 针对性：数据请求都有确定的目标

Agentic Speculation

- 持续性：为了完成任务需求，智能体需要持续频繁的访问数据系统
- 探索性：向数据系统发起的海量查询中有许多查询很可能是浪费的，只是帮助智能体建立“落地感”。

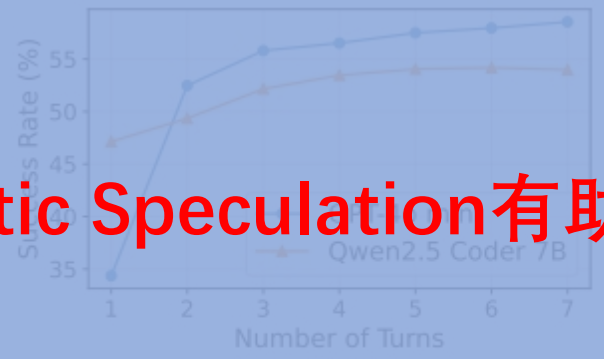
问题：

现有的数据系统大都是针对传统的工作负载，能否设计一种数据系统能够更好地支持 Agentic工作负载？

Background

平均成功率与 LLM 尝试次数的关系

(a) Success @ K



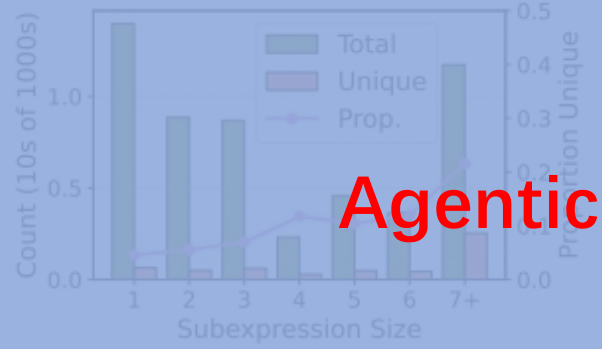
(b) Success vs. Turns

（a）一个“主控”LLM 智能体的指挥多个“前线”智能体分别独立尝试完成任务，最后由主控智能体在这些候[illegible]选择一个

（b）让单个 LLM 智能体不断发起查询，直到其对结果满意为止

Agentic Speculation有助于提高准确率

请求之间的共享程度



(a) versus subexpression size.

(b) versus root operation.

让 50 个独立智能体同时去解决同一个任务，于是得到 50 份彼此独立的查询计划，统计子表达式的统计结果

[illegible]表达式总数量、包含唯一子表达式的表达式的数量、比例

（b）针对根操作类型子表达式的统计

Agentic Speculation 在请求之间存在多个冗余

Background

智能体在执行任务过程中的活动分布

四种活动：

- 探索元数据与样本数据（如查询 schema ）
- 尝试部分查询
- 尝试完整查询

Agentic Speculation在信息需求上表现出异质性。

落地性的提示对智能体解决方案求解步数的影响

Table 1: Mean activity counts per agent trace, averaged across all traces, with and without human expert-provided hints.

Activity	Avg (No Hints)	Avg (w/ Hints)	Reduction (%)
exploring tables	3.53	2.95	-16.4
exploring specific columns	3.56	2.57	-27.7
attempting part of the query	4.28	2.71	-36.6
attempting entire query	1.26	1.05	-16.6
all SQL queries	12.67	10.38	-18.1

- 没有人工提示时的平均次数
- 有人工提示时的平均次数
- 减少率

Agentic Speculation可以通过落地性提示进行引导

Background

特征与机遇：

Agentic Speculation

1.**高吞吐性**：包含大量顺序或并行发往后端数据系统的请求，这保证任务完成的准确性

2.**冗余性**：许多请求可能访问相似的数据或执行重叠的操作

3.**异质性**：它涵盖了粗粒度的数据与元数据探索、部分与完整的解决方案制定及结果验证

4.**可引导性**：推测本质上是探索性的，落地性的提示可以帮助引导智能体推测请求

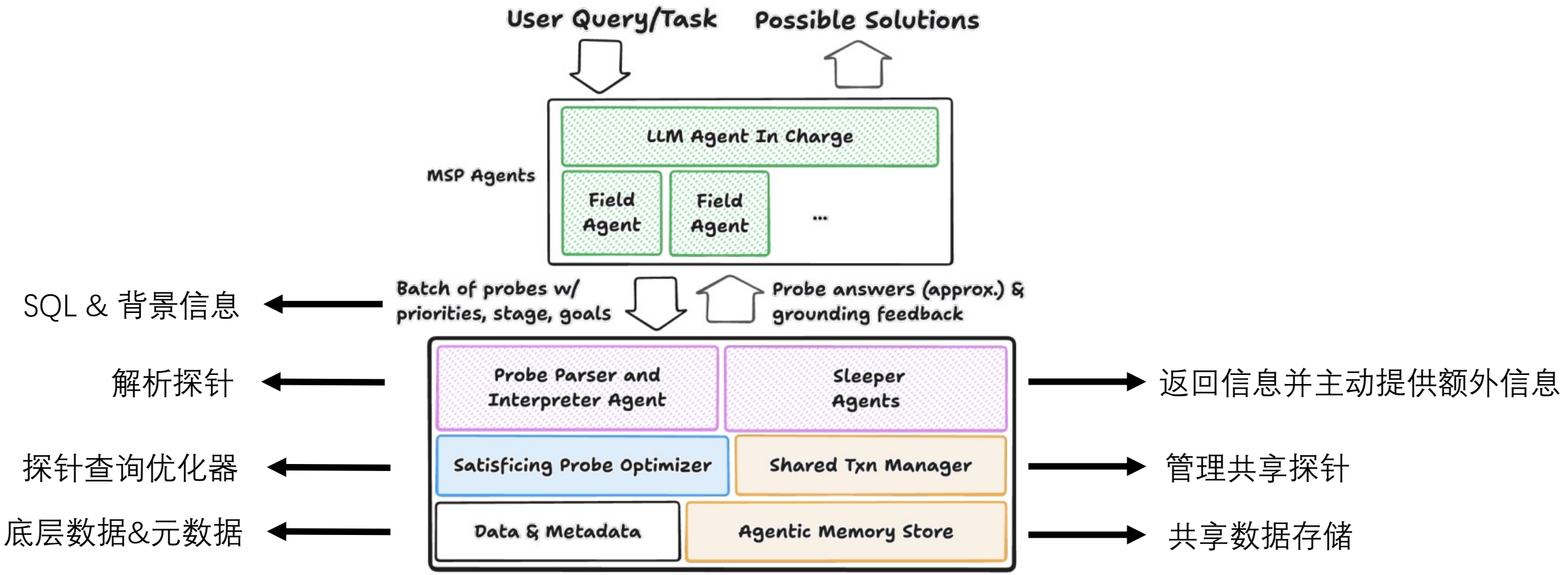
充分利用Agentic Speculation所具备的高吞吐性、冗余性、异质性与可引导性围绕智能体重新设计数据系统

Outline

- Background
- **Design**
- Conclusion

Design

SYSTEM ARCHITECTURE



Design

Probe && Probe parser

Probe
- SQL Query
- Brief
  - 探针的目标和意图
  - 当前所处的阶段（元数据探索 / 解决方案制定）
  - 近似需求与探针之间的优先级
  - 以及其他开放式信息

Probe parser and Interpreter Agent →
- 是否执行这些查询以及执行顺序（数据系统决定，以最大化效率）
- 根据目标、阶段以及开放信息来调整查询的精度（或近似度）
- 生成一些近似表达式（比如说生成评价函数，用于判断是否提前终止探测）

Probe Optimizer

协调自然语言查询与 SQL 查询

识别销售趋势上升的门店

1. 向搜索智能体提出问题，查找如何获取表的模式
2. 对数据库元数据表的简单元数据SQL的查询
3. 通过搜索智能体查找一些“趋势上升”的查询示例
4. 将返回的信息输入查询生成智能体，以生成查询SQL
5. 最后去执行查询销售趋势上升的门店的SQL

探针优化器：根据查询的阶段对其进行优先级排序（优先执行解决查询）

Design

Probe Optimizer

探针优化器还应该能够考虑效率，决定执行哪些探测以及以何种精度执行

同一批次中探针之间的优化：

- 数据系统使用语义理解与数据查询来检查查询是否符合用户意图，剔除不符合语义的查询。
- 数据系统会比较同一批次中的探针，通过成本估算和信息增益比较，决定哪些探针更有用或成本更低
- 数据系统会考虑智能体所处的阶段，在探索阶段返回粗略近似结果，而在方案制定阶段提供更准确的答案
- 数据系统还能够检查数据库的其他内部状态，以判断是否应继续执行当前查询，或进入下一回合
- 数据系统增量评估查询，优先提高最有用的探针的精度，然后再提高其他探针的精度。

Probe Optimizer

不同批次之间的探针间优化

数据系统利用与智能体的连续交互，进一步优化它决定执行的查询以及查询执行的方式。

- 判断本次查询是否提供相对于过去查询的更多的信息来决定是否要执行该查询
- 决定优先执行哪些查询，以最小化未来可能的执行探针数量。

数据系统可通过观察查询历史并考虑智能体意图，决定物化并缓存答案。例如，基于查询历史与智能体意图，数据系统可预测未来探测将继续涉及某些表的连接操作，从而可提前物化该连接结果。

Design

Sleep Agent

引导上层智能体提出更好的探测　　被动性 ⟶ 主动性

（1）提供数据系统认为相关的辅助数据信息作为旁路渠道，帮助上层智能体探测

Table 1: Mean activity counts per agent trace, averaged across all traces, with and without human expert-provided hints.

Activity	Avg (No Hints)	Avg (w/ Hints)	Reduction (%)
exploring tables	3.44	2.95	-14.2
exploring specific columns	3.56	2.57	-27.7
attempting part of the query	4.28	2.71	-36.6
attempting entire query	1.26	1.05	-16.6
all SQL queries	12.67	10.38	-18.1

（2）提供关于效率和成本的反馈，帮助智能体设计其探针

- Sleep Agent提供执行成本，建议智能体修改后续探针或者近似度
- 提供一系列任务的执行信息，建议智能体将后续探针进行批处理，以降低成本
- 考虑相关的缓存答案，或者是否刚刚为其他智能体回答过类似查询

Agentic Memory Store

存储内容

- 存储先前探针的结果和部分解，让智能体可以重用已知的数据和元数据，从而使相似探针更高效。
- 存储关于数据和元数据的信息，与表本身关联。

记忆存储更新

- 来自新探针的执行，为记忆存储提供补充或覆盖已有信息
- 来自底层数据或元数据的变更，需更新智能体记忆中相关信息

安全挑战

多用户访问控制：不同用户的智能体可能提出类似问题。跨智能体共享答案可提高效率，但会带来隐私风险，尤其是在汇总情况下。

Shared Transaction Manager

探针之间合作共享会导致数据的一致性、隔离性问题

分支隔离

- 多个智能体可能同时创建探索分支，这些分支最终必须相互协调，不仅与主线一致，还需彼此一致。
- 这要求建立多智能体、多版本隔离的新模型。大多数分支相似，但逻辑上要求它们的执行效果在保持独立。

回滚

- 加强版的 MVCC（多版本并发控制）：可能要同时分叉成多个几乎相同的快照，并最终只保留一个，其余全部回滚。
- 与传统数据系统不同，回滚在这里非常频繁，因此需要超快速回滚（对失败分支进行快速中止）。

Outline

- Background
- Design
- **Conclusion**

Conclusion

Agent-First Data System

- *Probe*
- *Probe parser and Interpreter Agent*
- *Probe Optimizer*
- *Sleep Agent*
- *Agentic Memory Store*
- *Shared Transaction Manager*

主要内容：描绘了一种面向智能体推测负载的数据系统愿景。充分利用Agentic Speculation所具备的高吞吐性、冗余性、异质性与可引导性围绕智能体重新设计数据系统。探讨了其中涌现的研究挑战。

Conclusion

- 这个paper有什么问题，基于这个paper还能做什么？
  - 这篇文章主要是一篇workshop，有一些地方只是提出了问题，并没有给出解决方案。
- 这个paper提到的idea，能不能用在自己的方向/project上面？
  - 核心观点是Agent决策时无法完全感知底层数据和数据系统状态。上层的Agent无法感知到底部各个容器的运行状态，导致决策次优。设计底层容器探针，探测底层系统，容器的状态。

Conclusion

- 这个paper能不能泛化？
  - LLM内置到数据系统内部，整个数据系统能够去优化查询——智能数据系统，这种思想可能也能覆盖到其他系统中去，不通过prompt去给上层大模型传递底层信息，而是与底层系统进行抽象的逻辑交互。

東南大學
SOUTHEAST UNIVERSITY

恳请各位老师与同学批评指正！

汇报人：束方磊

2025.9.20