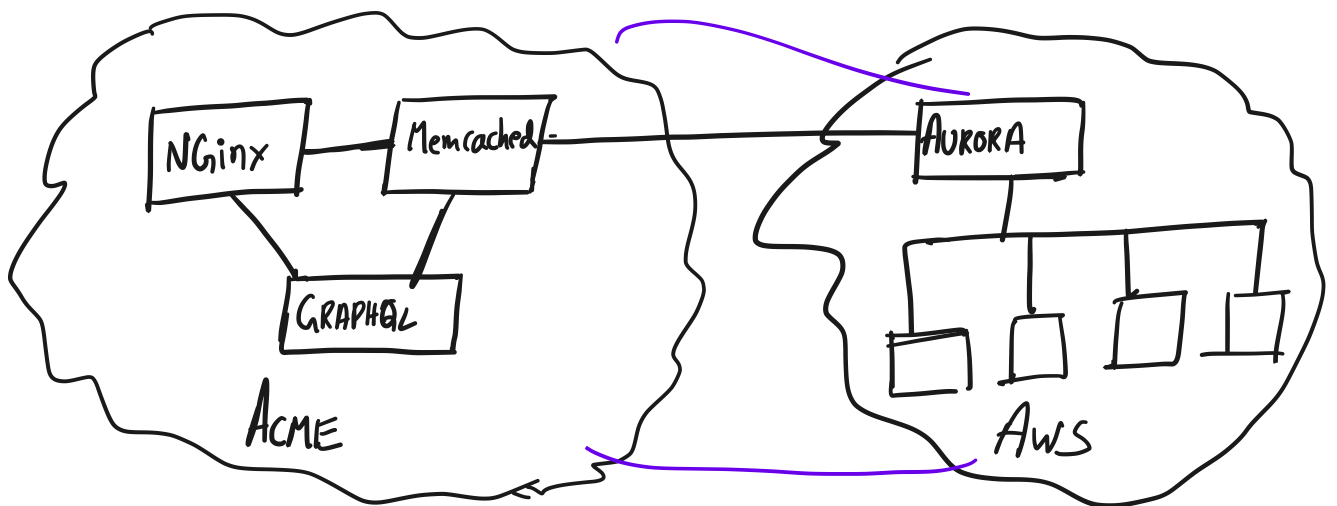


# Lecture 3:

Do over?

## THREE FRAMING QUESTIONS FOR THE SEMESTER.

### Q1. CROSS-DOMAIN TRACING



DESIRABLE

## CHALLENGE

## POSSIBILITY

Q2. Who uses traces & how?

Focus thus far: Humans for debugging.

What about other uses?

Reproducing bugs

What?

How do we ensure right now?

Why challenging right now?

## Security

- What?

- Audit

- Unusual data flow

-

- Why challenging now?

Scheduling, auto scaling, resource allocation

What?

Why challenging?

Q3. Trace sizes: why are they so large?

- Why this matters?

- Fundamental question: What do we need to record?

↳ Depends on use.

→ For reproducing bugs (proxy for human?)

→ For security

→ Other uses?

- Does reducing trace size come at a cost?

## TODAY'S PAPERS

- REDUCE TRACE SIZE BY IDENTIFYING & RECORDING  
WHAT IS NECESSARY.

↳ TWO RELATED (?) APPROACHES

HOW DO THEY FIT INTO THE QUESTIONS:

HOW DO WE DO THIS?

WHAT DO WE DO WITH THE TRACES? (WHO & HOW)

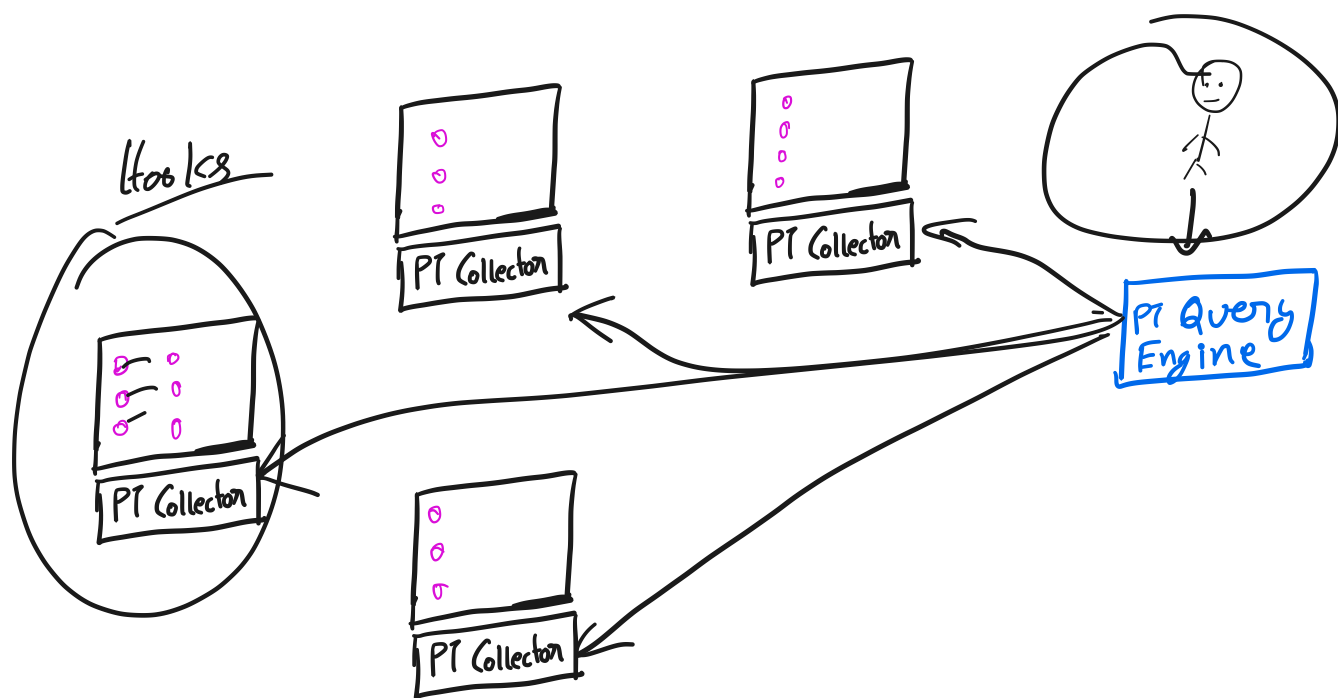


WHAT DO TRACES RECORD? (WHY ARE THEY SO BIG)

UNFORTUNATE REALITY: KNOWING HOW TRACES ARE GOING TO BE USED DOES NOT ALWAYS REDUCE TRACE SIZE.

WHY?

# OVERVIEW (STARTING FROM PIVOT TRACING)

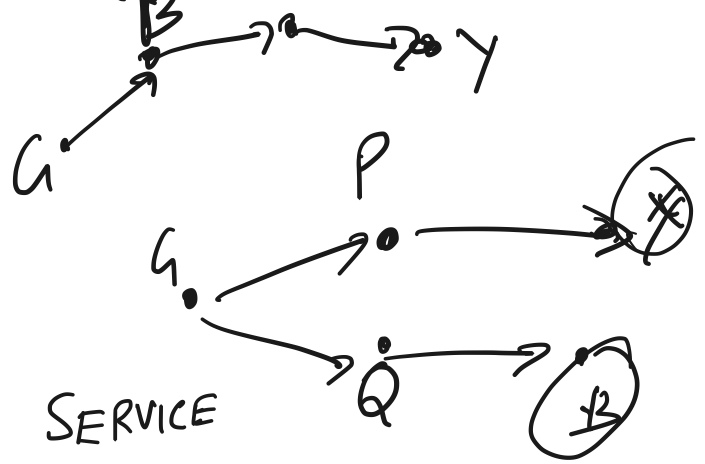


## CHALLENGES

- How to decompose? [AA Design]
- A single process might not have enough information for its part of the query? [X, baggage]
- Limit what a process records. [API, baggage, X]

How to decompose / API design.

FROM X IN Y  
 ↑  
 Hook



NOT RESTRICTED TO ONE SERVICE

FROM X IN Y  
 JOIN A IN B ON A → X  
 . . .

- GIVEN A QUERY CAN IDENTIFY
  - WHERE DATA NEEDS TO BE COLLECTED.
  - USING WHAT HOOKS.

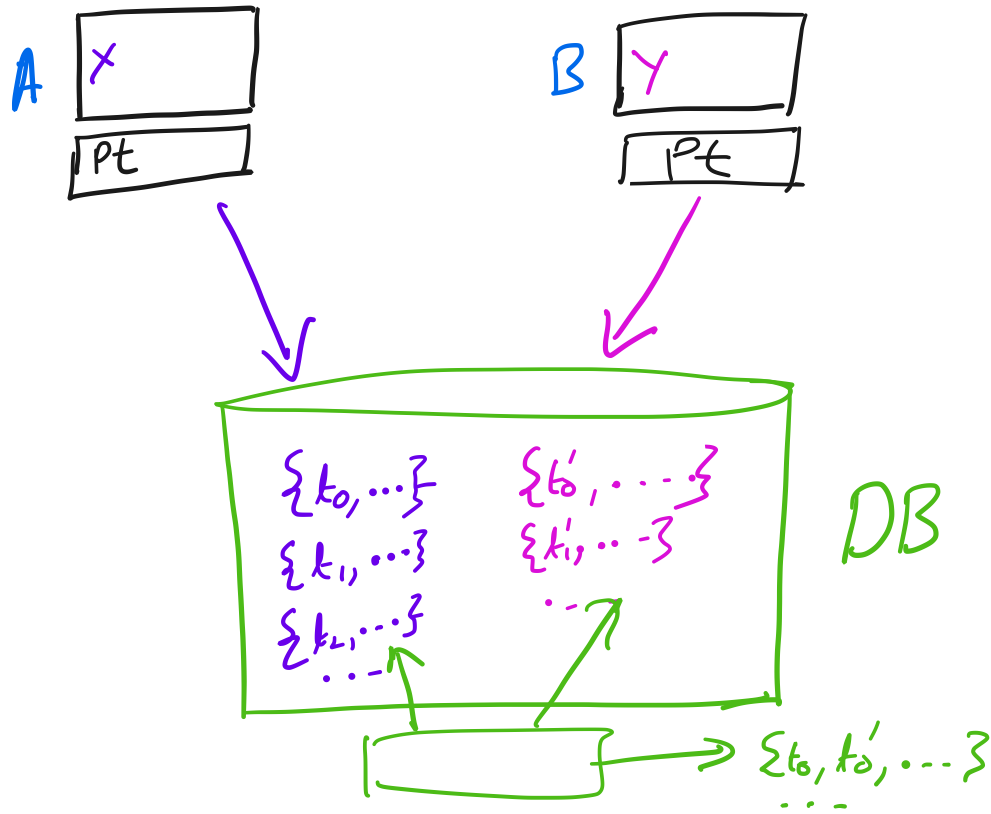
TWO RELATED PROBLEMS

- FROM A IN X  
 JOIN B IN Y ON B → A  
 WHERE B<sub>0i</sub> = . . .
- FROM A IN X  
 JOIN B IN Y ON B → A



SELECT A.i, B.j

How this is classically implemented



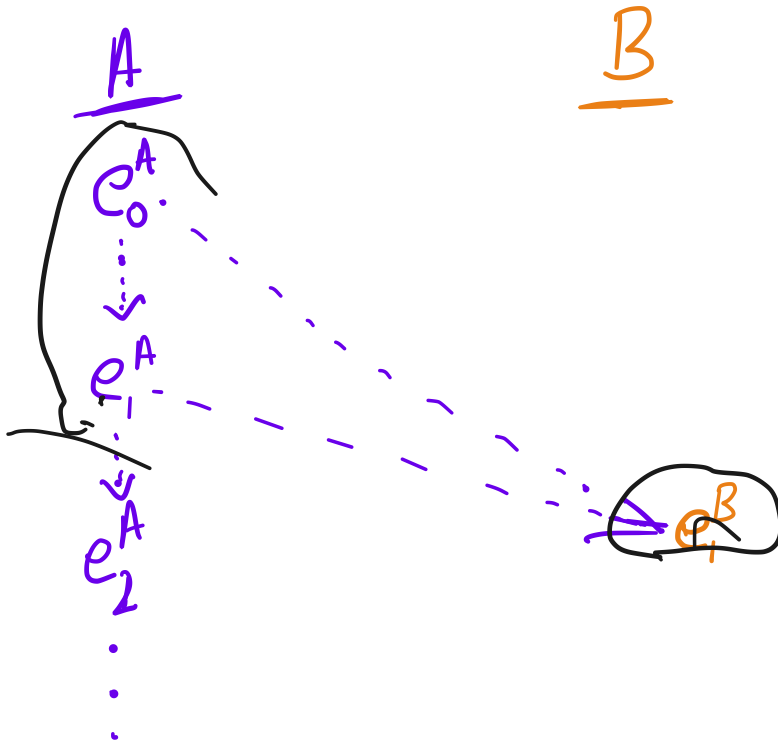
MANY PROBLEMS

FROM A IN X

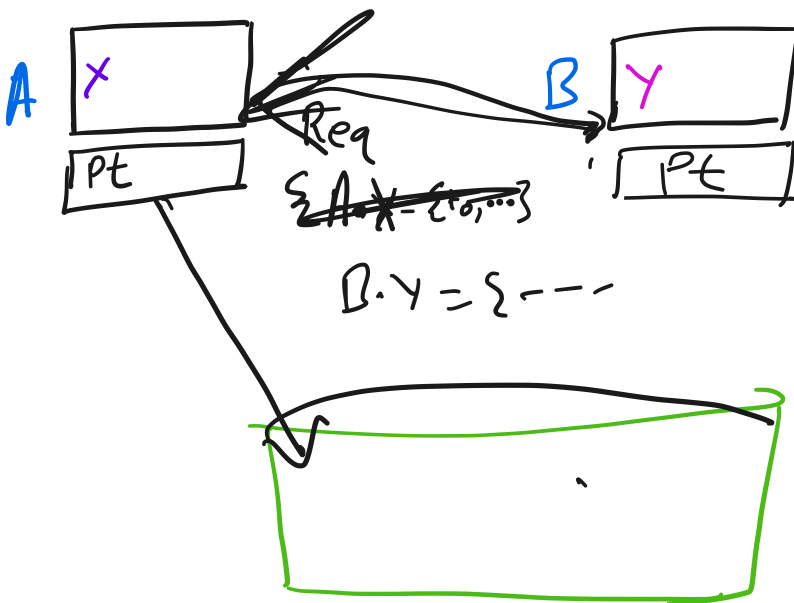
JOIN B IN Y ON  $B \rightarrow A$

WHERE  $B_{0i} = \dots$

MORE GENERALLY



PIVOT TRACING APPROACH



WHAT ABOUT CAUSALITY?

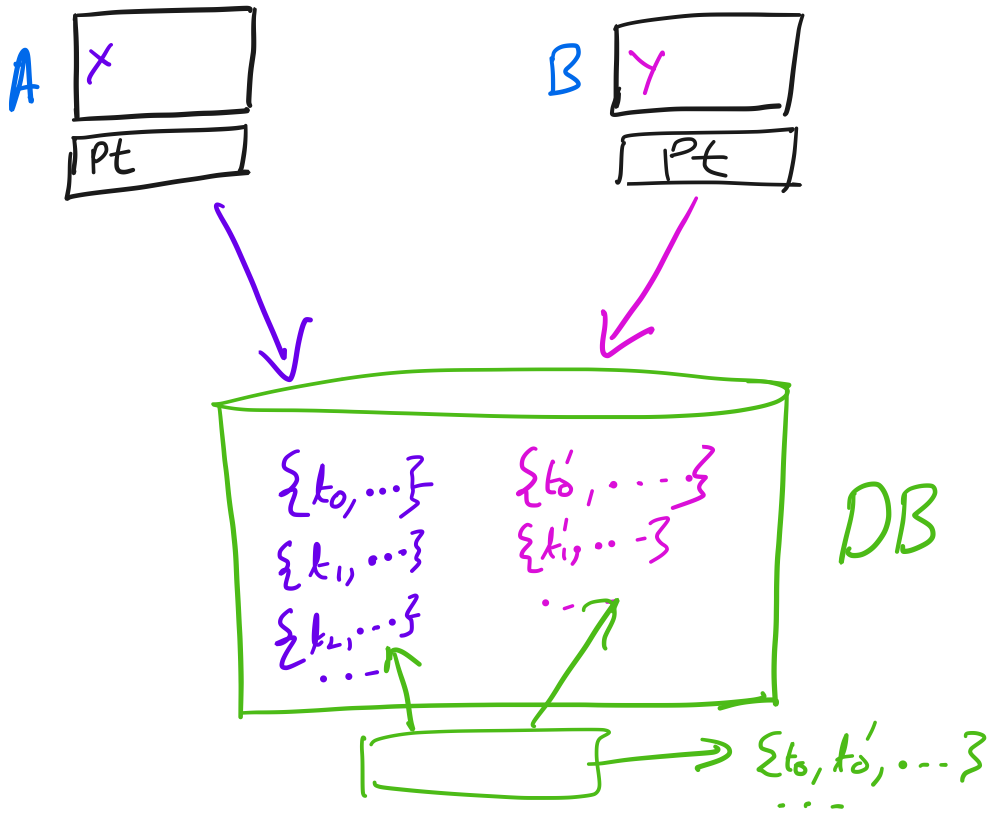
CHALLENGES?



ERGONOMICS: Adding trace points?

SAMPLING? NOT SAMPLING? OVERHEADS?

Where does snicket fit in?



WHERE & WHEN DOES THIS WORK?

