

How to Get an Order of Magnitude in Software Productivity (for data-oriented applications)

Mike Stonebraker

Agenda

- Introduction
- Increasing software engineering productivity 10x
- DBOS demo
- Q&A

Founded on MIT-Stanford joint R&D



Mike Stonebraker
CTO
MIT, Turing award winner



Matei Zaharia
Founding Advisor
Databricks CTO



Christos Kozyrakis
Founding Advisor
Stanford Platform Lab



Chuck Bear
Co-founder/Architect
Chief Scientist Vertica



Michael Coden
Co-founder CEO
MD BCG Security Practice



Andy Palmer
Co-founder
Tamr Chairman



Qian Li
Co-founder/Architect
Stanford PhD



Peter Kraft
Co-founder/Architect
Stanford PhD

DBOS Overview

```
class Greetings {

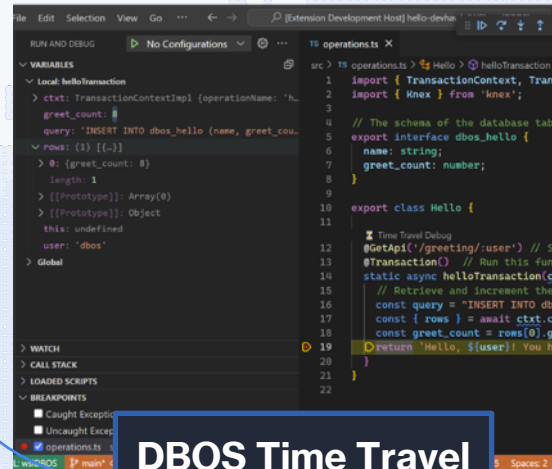
  @Workflow()
  @GetApi("/greeting/:friend")
  static async GreetingWorkflow(ctx: WorkflowContext, friend: string) {
    const noteContent = `Thank you for being awesome, ${friend}!`;
    await ctx.invoke(Greetings).SendGreetingEmail(friend, noteContent);
    await ctx.invoke(Greetings).InsertGreeting(friend, noteContent);
    ctx.logger.info(`Greeting sent to ${friend}!`);
    return noteContent;
  }
}
```

Open source DBOS Transact

- TypeScript framework
- Simplify back-end development
- Reliable & observable by default
- Run anywhere
- Deploy to DBOS Cloud...

DBOS Cloud (on Amazon)

- Host and run DBOS Transact applications, workflows
- Auto-scaling
- Auto restart-resume
- Disaster & cyber-resilient



DBOS Time Travel Debugger

- Troubleshoot & debug
- Replay execution and state of any DBOS code
- Modify code and retry
- Query observability data

How to Get an Order of Magnitude in Software Productivity (for data-oriented applications)

Mike Stonebraker

And it is not LLMs/ML

- Co-pilot is worth 30 - 50% productivity improvement
- Stay tuned for an order of magnitude!

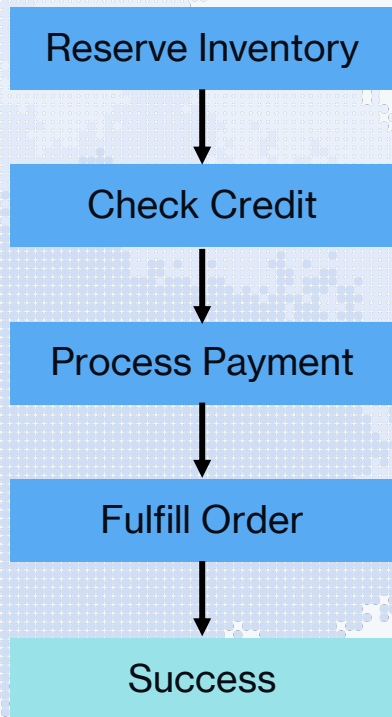
Everybody is moving everything possible to the cloud as quickly as possible

- Cost (unless you cheat)
- Resource elasticity

To fully leverage the cloud:

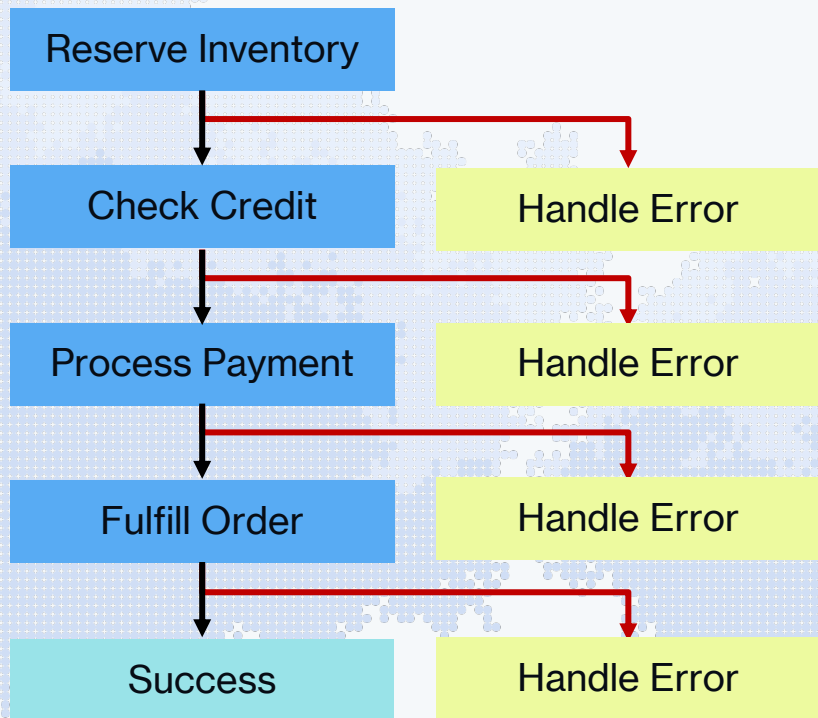
- Structure your application as a directed graph (workflow) of operations
- Passing state between operations through storage (S3) or as parameters to the “next in line”
- You get charged for computing only when you are active – not all the time
- Think AWS Lambda and Step Functions

Example: Workflow for a “Buy Button”



The forward logic seems simple to implement...

Challenge: The Error Branches



The forward logic seems simple to implement...

...but what happens when operations fail?

Bigger Challenge: The “Oops” Logic

- On a restart from a crash, then ???
- On a disaster or ransomware attack, then ???
- On an overload, then ???

In our experience.....

- The forward logic is 10%
- Everything else is 90%

DBOS Workflow Guarantees

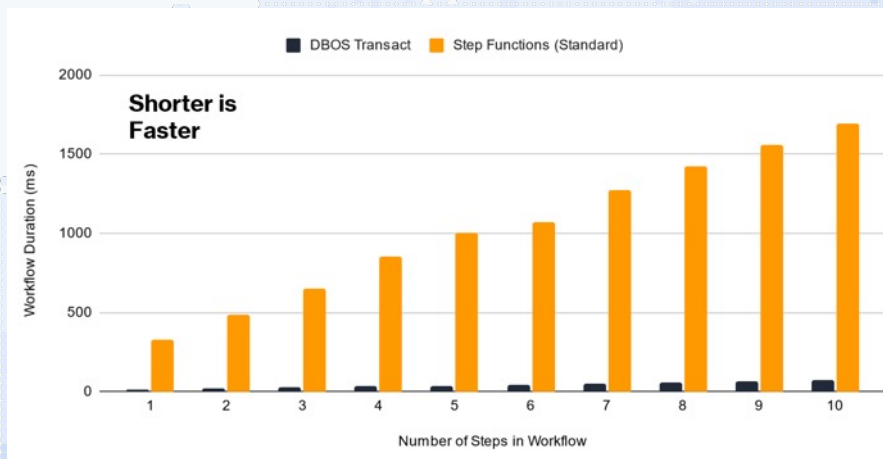
- Once-and-only-once (OAOO)
 - Don't want to pay twice or ship twice
 - One regional bank estimated that 40% of their code base implements OAOO
- “Database sagas”
 - If bad address for fulfillment, want to unwind everything
 - Physical backout (sometimes); often logical backout
 - Compensating transaction for decrementing inventory is incrementing inventory
- Workflow finishes even if disaster strikes
 - When there is a crash or a ransomware attack or

With these guarantees.....

- You write the operations, the platform handles everything else
 - Orchestration (run operations when inputs are available)
 - Careful bookkeeping on errors to support OAOO
 - Sagas to support logical backout
 - “Time travel” to backup everything on disasters and then replay
 - Load balancing and scalability
- Worth an order of magnitude!!!!
 - According to early DBOS programmers
 - “We got done in 2 days what took 2 months in AWS Lambda”

Other goodies

- Time travel debugger
 - Back up your application and then single step and/or change stuff
- Performance
 - Co-locate orchestration, the operations and the DBMS
 - Steps are stored procedures in the DB
 - Coded in the language of the app, not something else (TypeScript vs. PL-SQL)
- Cheap
 - An order of magnitude cheaper/faster than AWS Lambda



And Security

- Much smaller attack surface (Linux/Kubernetes gone)
- All events (in the platform and all applications) are recorded in a data warehouse
 - Selective or full recovery to a point in time

Summary

- X10 productivity improvement on data-oriented applications
 - Because you write the app; the drudge is handled by the platform
- Together with lower cost and better performance
- And increased security/resiliency
- Check out [DBOS.dev](https://dbos.dev) for more information

