

# DP-ECDS@PLDI'2015

---

**Title: Declarative Programming over Eventually Consistent Data Stores**

## Background

---

有些存储系统支持多种数据一致性模型 (Refs: Microsoft, Terry@CACM), 允许用户为每个操作单独指定更细粒度的一致性模型。但是, 用户所关心的应用逻辑 (比如可以表达成不变式) 与存储系统所提供的一致性模型之间存在语义鸿沟。

*Q* : 如何帮助用户为每个操作选择合适的一致性水平 (consistency level)?

## Contributions

---

本文提出了

## Methods

---

### Contract

使用 Contract Language 描述 application-level consistency。Contract 语言以 Op/Eff 为原子变量, 以 *so*, *sameobj* 与 *vis* 为原子 relation。

*vis* relation is dynamic while *so* is static!!!

*so* 仅反映静态的程序文本信息; *vis* 反映了运行时的语义信息, 是 execution 层面的 relation。

## Possible Direction

---

## Automatic Inference of Consistency Contracts:

- Problem:
  - Input: program + invariant
  - Output: consistency level for per (kind) of operations
- Possible Techniques:
  - ☐ synthesize
  - ☐ Exactly as the method in VisRelax@POPL'2019 (enumeration)
  - ☐ PBE (Programming By Examples)
- Related Work:
  - ☐ "Automatic Inference of Memory Fences"@FMCAD10
  - ☐ Cheng Li

## Questions

---

- ☐  $so \subseteq vis \wedge vis$  is transtive 与  $(so \cup vis)^+ \subseteq vis$  是什么关系?
- ☐ "causal cut of updates" 与 "causally consistent snapshot" 是什么关系?
  - ☐ 为 "causal cut" 定义相应的一致性模型?