TurKit: Human Computation Algorithms on Mechanical Turk
Crash-and-rerun

- What’s the primary value of this model? Simple programming or keeping incremental results if you have an error downstream?
Crash-and-rerun

- Debugging
  - Manas: easy to forget to clear cache
  - Wesley: connection between particular version of code and db contents isn’t obvious. Could be solved by annotating cache entries with commit hash?

Kurtis: just preserve hash of function call (including args)
Bjoern: this only works if you use a linear style of programming (too hard if you have conditionals, etc.)
Crash-and-rerun

- Scalability
  - Nicholas: this model may be impractical depending on how much computation a task entails
  - Authors note this as well:

This means that waiting for a single human takes an order of magnitude longer than running most of our scripts, which suggests that crash-and-rerun programming is suitable for many applications. The slowest script is faster than 99% of our hit-completion times. Note that making the script 10x slower would only be faster than 70% of hit-completion times. For such a slow script, it may be worth investigating options beyond the crash-and-rerun model.
Beth: task design (how to get iterative improvement?)
Bjoern: two types of iteration – 1) buggy code, and 2) posting HITs
Kristal: more like Dryad would be nice
Add a chatroom so users can work together
BH: this experiment was done by Kittur at CMU (translate a poem from Spanish to English)
MM: MT b/c you leverage different competencies
Synergy: brainstorming
KH: understanding instructions
Higher burden of proof with mturk – not just “was it statistically significant?”
Can you do human replication like you can use replication on EC2?
PP: we’re not rational, so makes it harder to replicate
BH: need tool to find our assumptions about human behavior (social psychologists/sociologists)
PP: could have intentional liars too
Adjustable Compensation

- Sally:
  - “First task often has a great impact on the subsequent improvement tasks”
  - “Much of hard work is often completed by the first worker”
- Idea: pay more for tasks earlier on
- Impact on answer throughput/quality?
Interface/Usability

- **Sally**: easy to use
- **Siamak**:
  - “Their main contribution is abstracting the complexity of mechanical turk and providing a programmer-friendly interface”
  - Wants more access to quality control