Satsy

Input-Output Code Search Engine

Carl Chapman
Cole Groff
Cody Hoover
Trevor Lund
Kaitlin McAbee

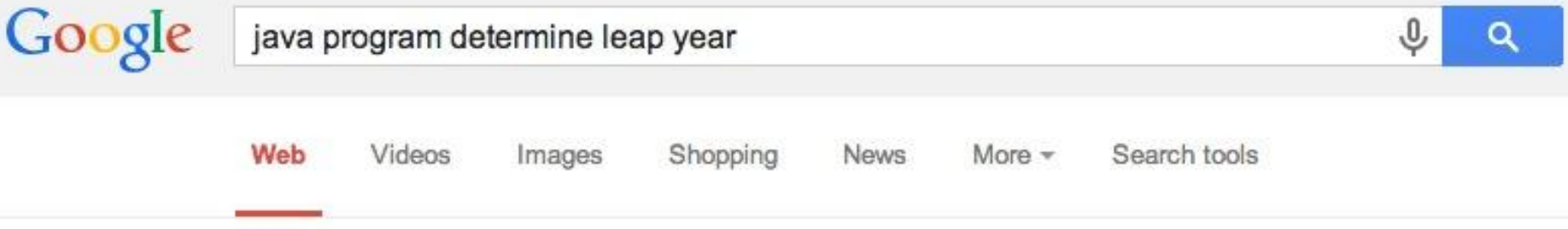
Problem Statement

May14
13

- Lots of open-source code is available for use.
- Instead of rewriting code that already exists, can we search to find the existing code that does what we want?
- Assume we can describe functionality as an Input-Output pair.

Satsy Example

Google Example



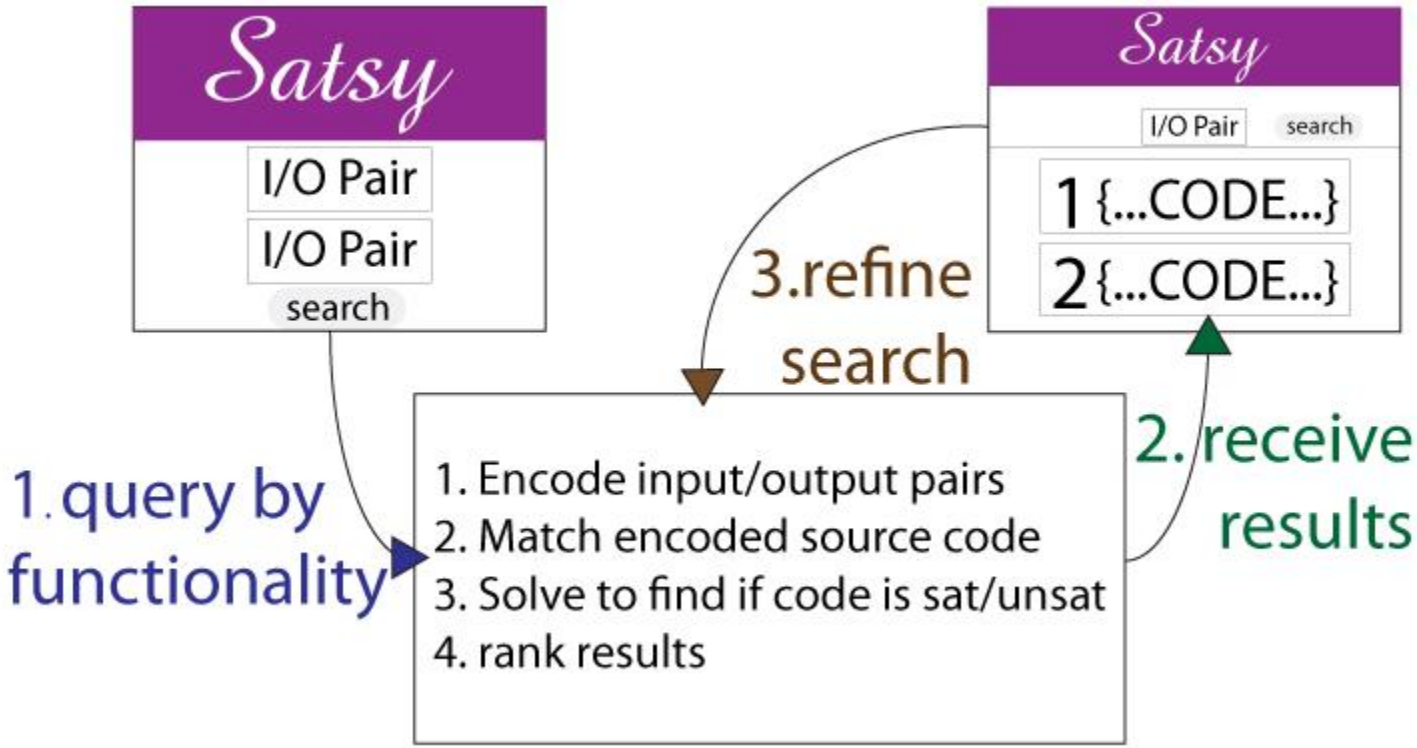
Prototype Market Study

May14
13

Google		Satsy
48.4	Results Returned	20.5
1.5	Satisfactory Results [1]	8.5

Concept Sketch

May14
13

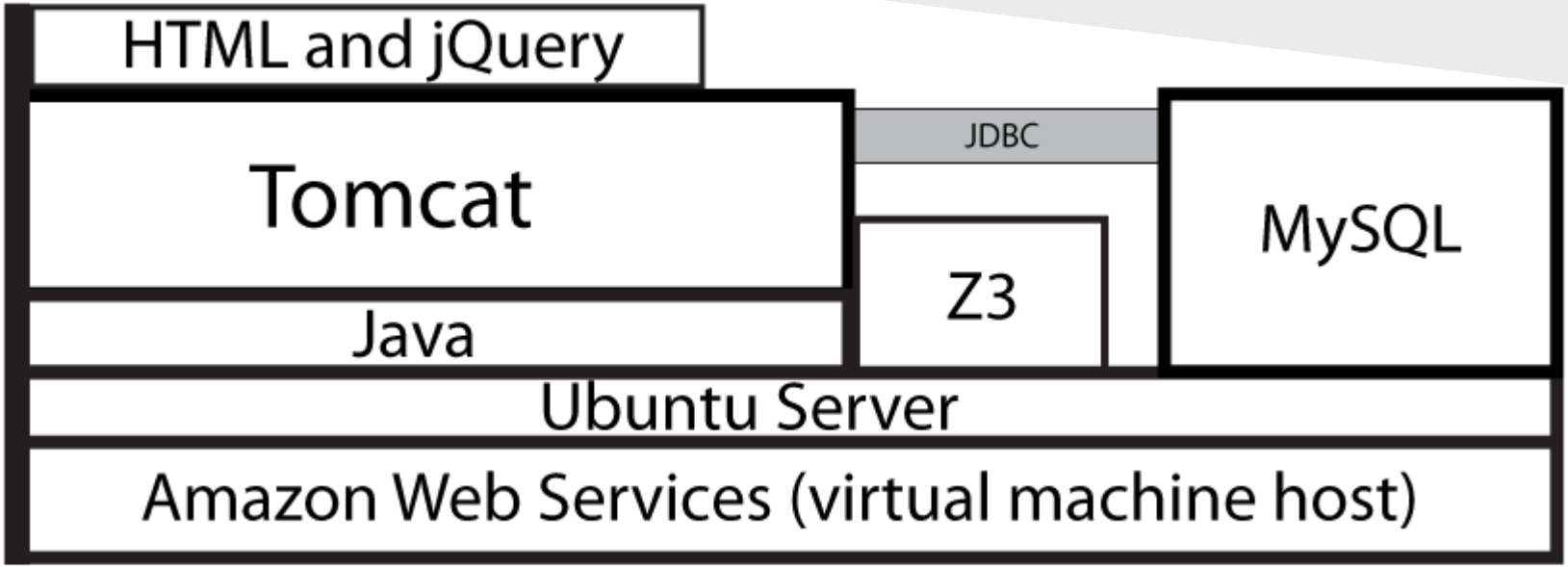


Live Demonstration

May14
13

Challenge 1: Technologies

May14
13

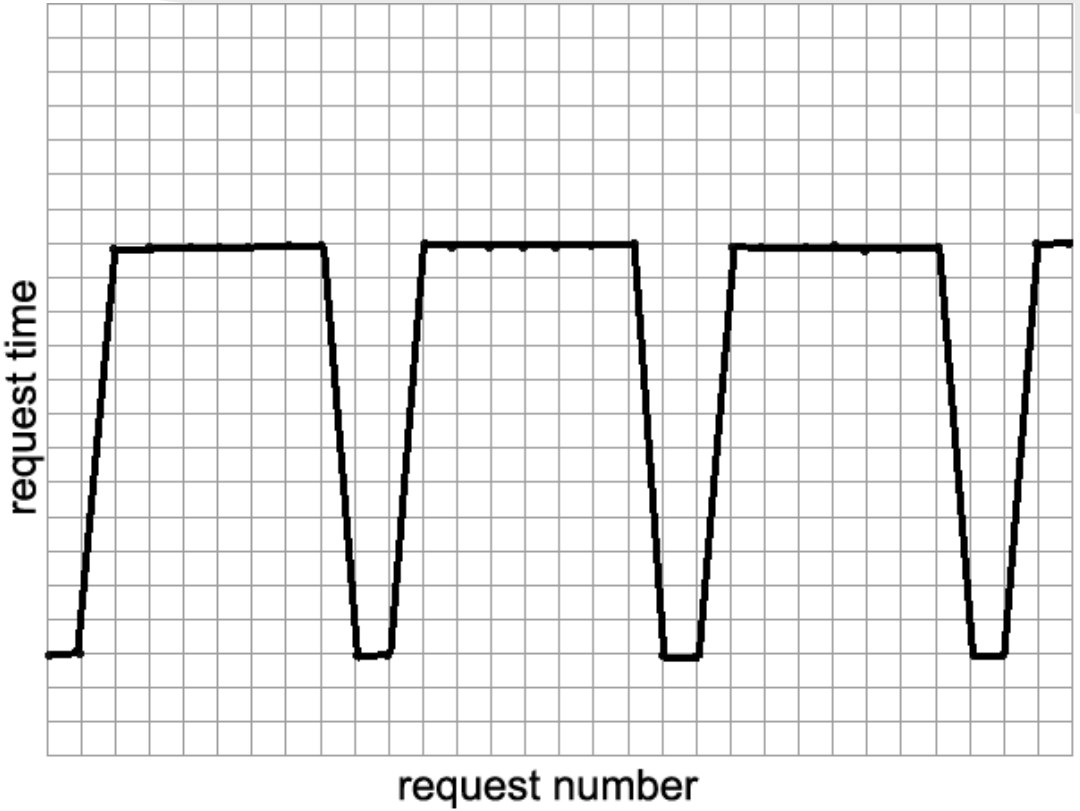


- Amazon Web Services setup
- Running Z3 from Java application
- JDBC connection pool
- Running Java code from HTTP requests
- Asynchronous requests and responses
- Deploying Java web application

Challenge 2: AWS

May14
13

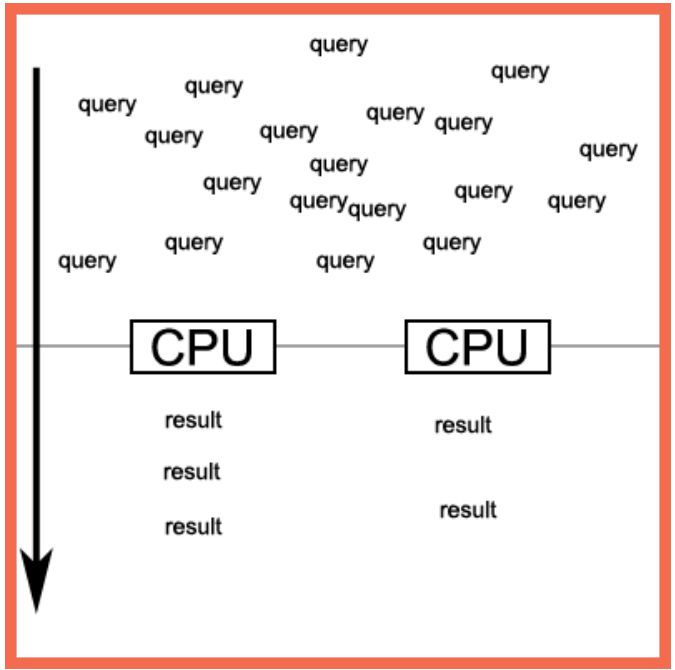
- Why do some search queries seem to stall while others are much more responsive?
- Investigation reveals the AWS EC2 free-tier imposes CPU throttling.

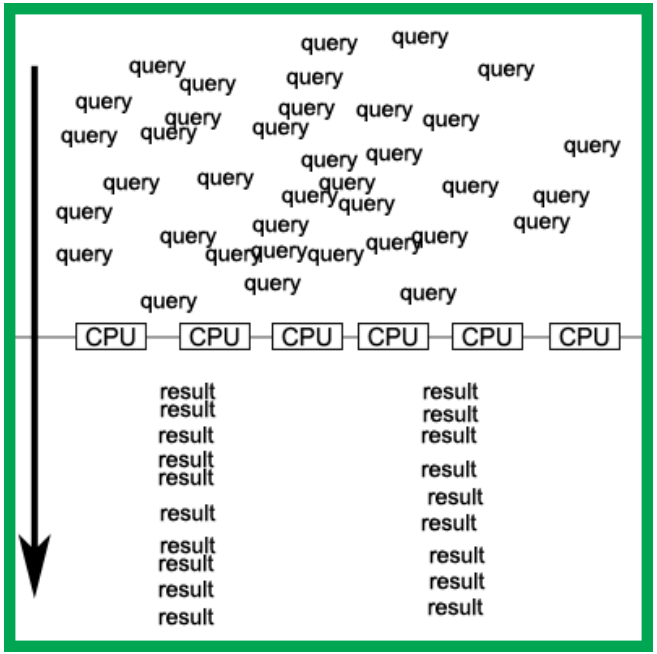


Challenge 3: CPU Bound

May14 13

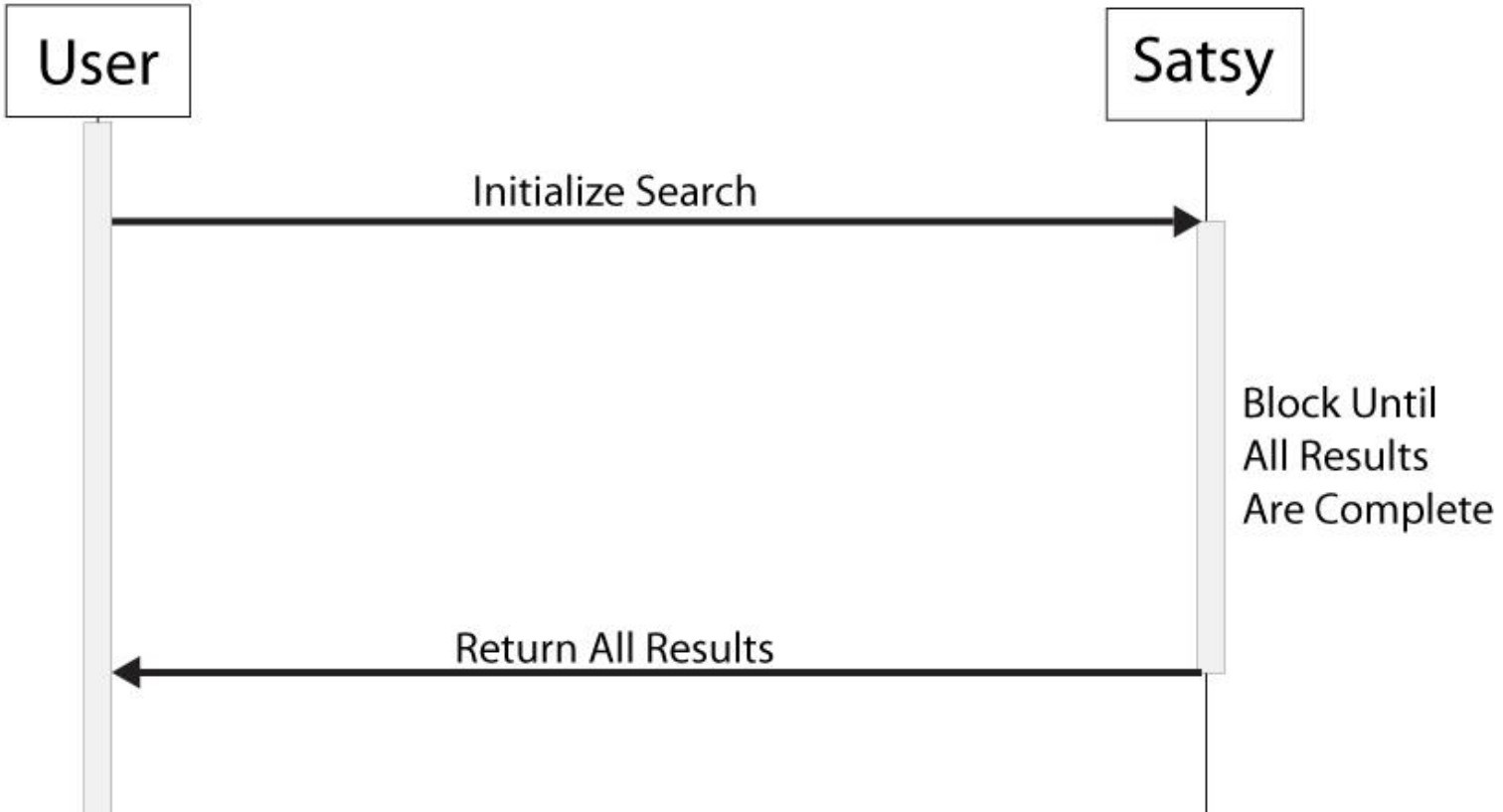
- One Solver takes at least 60 Milliseconds of CPU time
- Searching 1000 database entries takes at least 1 minute





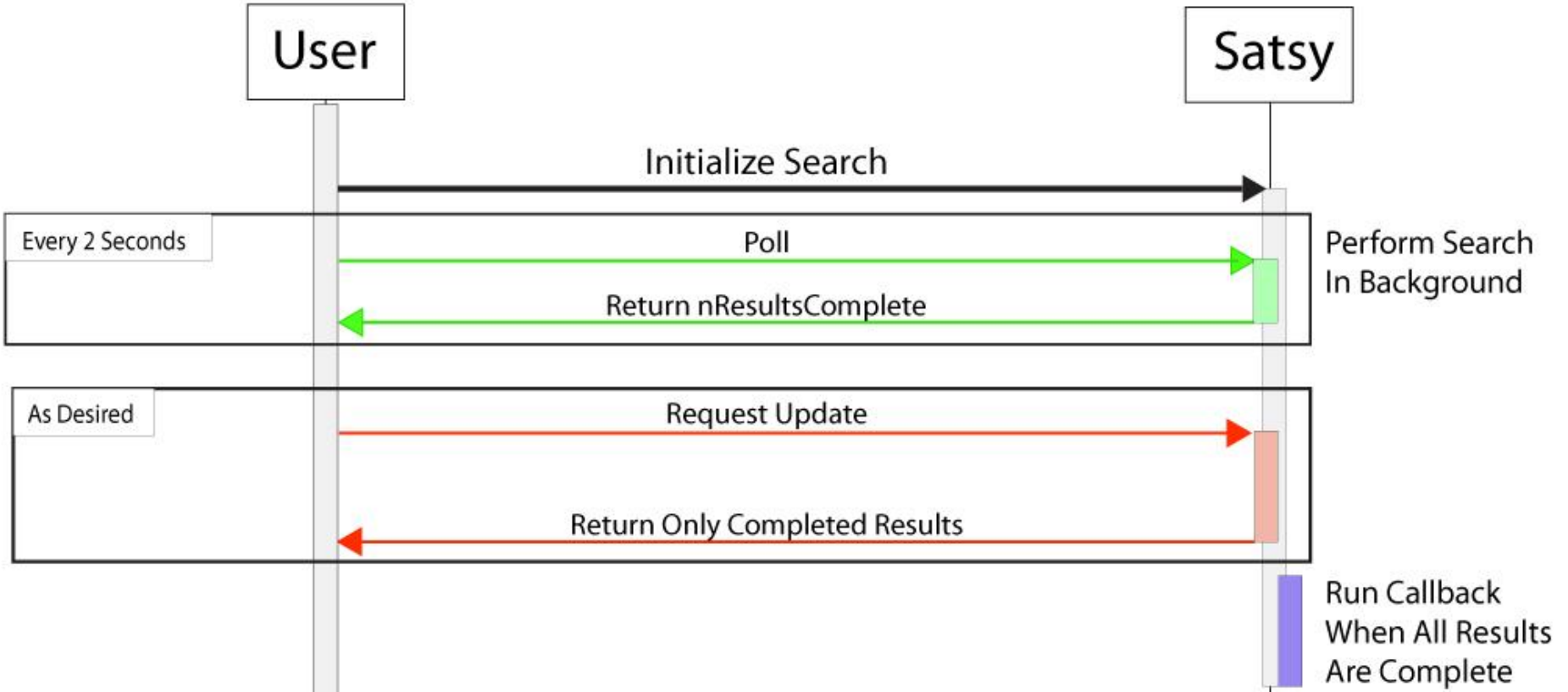
Original Design

May14
13



Improved Design

May14
13



Initialize Search

May14
13

1. end old search
2. prepare new ResultMap and Executor
3. create collection of Solvers
4. launch all the Solvers using the new Executor

Solver Algorithm

May14
13

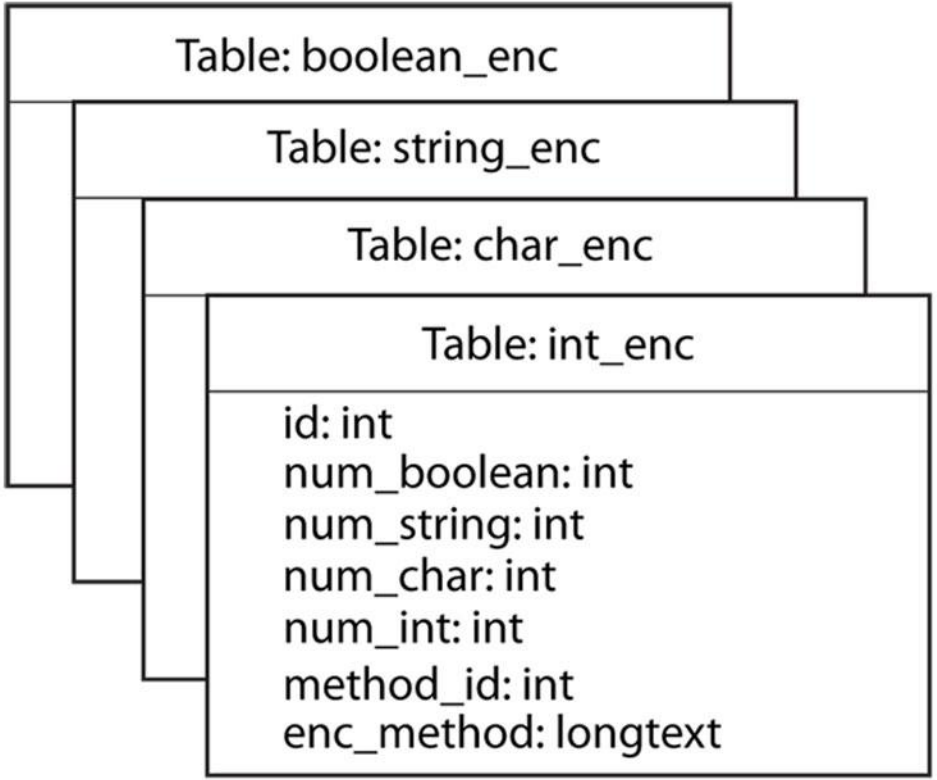
1. check for search timeout
2. create SMT file
3. execute z3 and read result
4. increment result matrix
5. notify SearchCallback

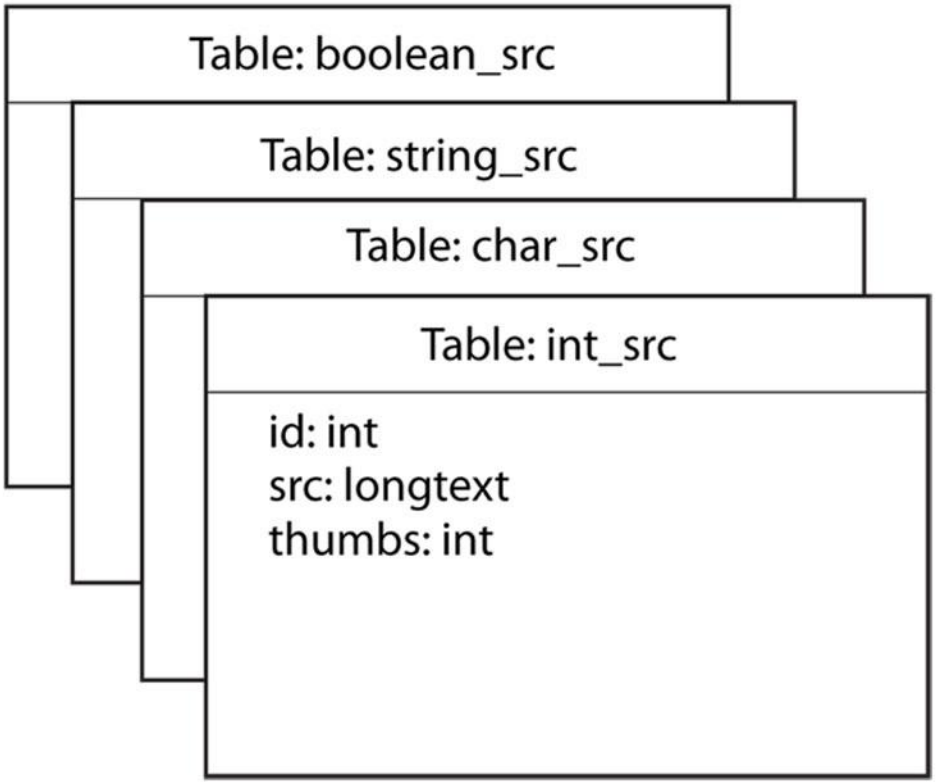
Result Matrix

	IO Pair 1	IO Pair 2
INITIAL	0	0
UNKNOWN	0	0
SAT	0	0
UNSAT	0	0
TIMEOUT	0	0
ERROR	0	0

Database Structure

May14
13





Paths Encoded as SMT

May14
13

id	src
566	

```
public static boolean isLeapYear(int year){
  if (year % 4 == 0) {
    if (year % 1000 == 0 && year % 400 != 0){
      return false;
    }
    return true;
  }
  return false;
}
```

source code for the method isLeapYear

id	nb	nc	ni	ns	m_id	enc_method
679	0	0	1	0	566	
680	0	0	1	0	566	

```
...
(assert (= additive6106 (mod year int6105)))
(assert (= boolean6115 true))
(assert (= relationalEq6113 true))
(assert (= int6100 4))
(assert (= int6102 0))
(assert (= int6107 0))
(assert (= relationalEq6108 (= additive6106 int6107)))
(assert (= relationalEq6108 true))
(assert (= int6112 0))
(assert (= int6110 400))
(assert (= additive6111 (mod year int6110)))
(assert (= relationalEq6103 true))
(assert (= int6105 1000))
(assert (= relationalEq6113 (= additive6111 int6112)))
...
```

SMT code for an execution path through isLeapYear

Ranking

May14
13

- Generic superclass that can be extended to create new algorithms.
- Current algorithms are sorted by number of paths satisfied, percentage of paths satisfied and if all paths are satisfied.

Front End Design

May14
13

- Single Page Web App
- Asynchronous Communication
- Template-based presentation

523 -- 3

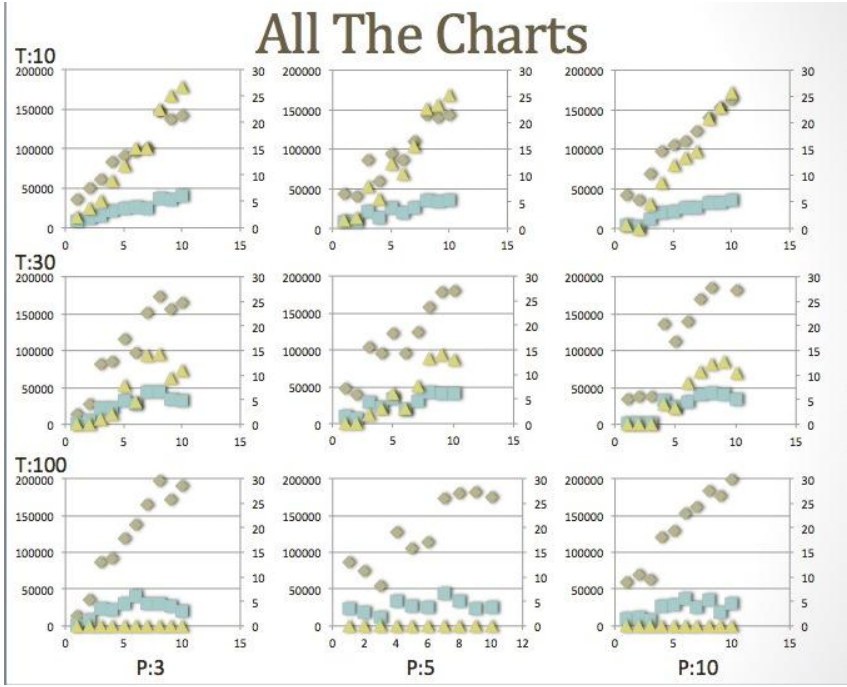
```
public static boolean isgraph(int ch){
  return ('!' <= ch && ch <= '~');
}
```

Show 22 More Results

Testing 1: User Simulations

May14
13

- Strong Engineering Effort put into comprehensive tests
- Relied on Blocking/All-or-nothing nature
- Architectural change -> Out the window



Testing 3: Revised Testing

May14
13

- GUI: Manual “Mock” Flags
- Backend: Unit Tests
- Exceptional Cases: Database Integrity

Testing 2: SearchCallback

May14
13

- Shift to Asynchronous/Incremental DeliveryArchitecture
- Less control over timing, more “events”
- On completion call SearchCallback

Questions

May14
13