



Sloth: Being Lazy is a Virtue

When Issuing Database Queries

Alvin Cheung

Sam Madden Armando Solar-Lezama

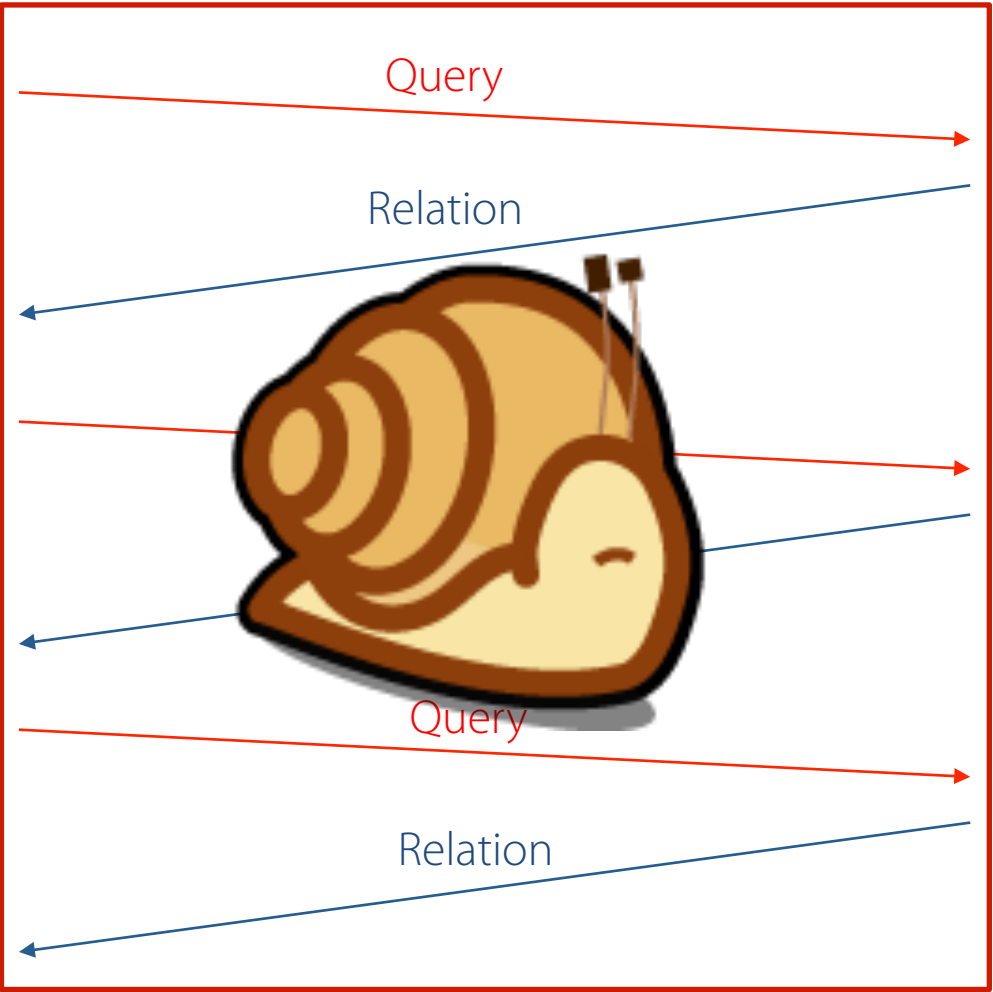
MIT CSAIL

```
void
usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
            name);
    #ifdef LOFF
    printf ("[-g] [-G] ");
    #endif
    printf ("[-p what] [-r]
            [-u file [type]]");
    #ifdef LOFF
    printf (" [-w love] [-w
            ewid] [-z size] ");
    #endif
}
```

```
void
usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
            name);
    #ifdef LOFF
    printf ("[-g] [-G] ");
    #endif
    printf ("[-p what] [-r]
            [-u file [type]]");
    #ifdef LOFF
    printf (" [-w love] [-w
            ewid] [-z size] ");
    #endif
}
```

```
void
usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
            name);
    #ifdef LOFF
    printf ("[-g] [-G] ");
    #endif
    printf ("[-p what] [-r]
            [-u file [type]]");
    #ifdef LOFF
    printf (" [-w love] [-w
            ewid] [-z size] ");
    #endif
}
```

```
void
usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
            name);
    #ifdef LOFF
    printf ("[-g] [-G] ");
    #endif
    printf ("[-p what] [-r]
            [-u file [type]]");
    #ifdef LOFF
    printf (" [-w love] [-w
            ewid] [-z size] ");
    #endif
}
```



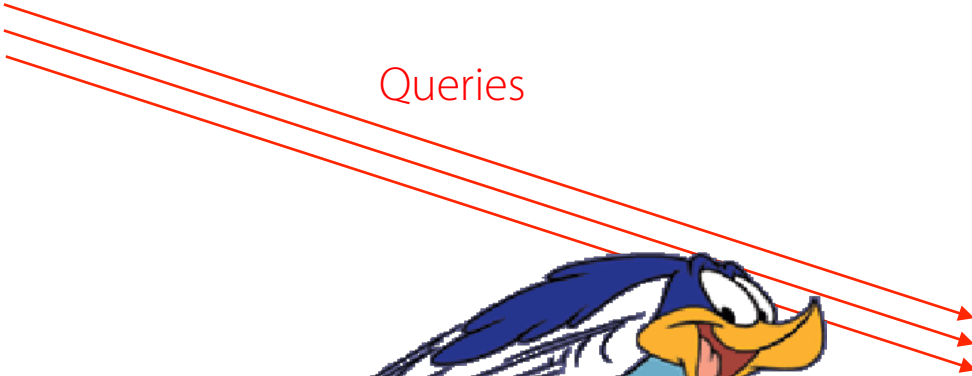
```
void
usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
name);
#ifdef LOFF
    printf ("[-g] [-G] ");
#endif
    printf ("[-p what] [-r]
[-u file [type]]");
#ifdef LOFF
    printf (" [-w love] [-w
ewid] [-z size] ");
#endif
}
```

Queries



Relations

```
void
usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
name);
#ifdef LOFF
    printf ("[-g] [-G] ");
#endif
    printf ("[-p what] [-r]
[-u file [type]]");
#ifdef LOFF
    printf (" [-w love] [-w
ewid] [-z size] ");
#endif
}
```



```
void usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
    name);
    #ifdef LOFI
    printf ("[-g] [-d] ");
    #endif
    printf ("[-p what] [-r]
    [-u file [type]]");
    #ifdef LOFI
    printf (" [-w lowe] [-w
    eweId] [-z size] ");
    #endif
}
```

```
void usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
    name);
    #ifdef LOFI
    printf ("[-g] [-d] ");
    #endif
    printf ("[-p what] [-r]
    [-u file [type]]");
    #ifdef LOFI
    printf (" [-w lowe] [-w
    eweId] [-z size] ");
    #endif
}
```

```
void usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
    name);
    #ifdef LOFI
    printf ("[-g] [-d] ");
    #endif
    printf ("[-p what] [-r]
    [-u file [type]]");
    #ifdef LOFI
    printf (" [-w lowe] [-w
    eweId] [-z size] ");
    #endif
}
```

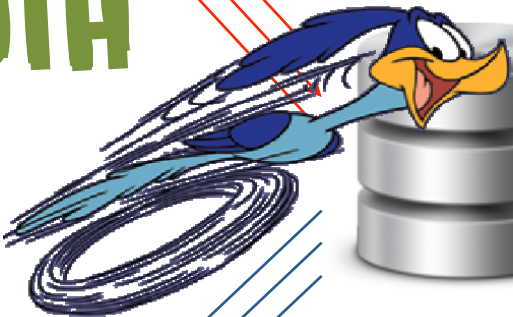
```
void usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
    name);
    #ifdef LOFI
    printf ("[-g] [-d] ");
    #endif
    printf ("[-p what] [-r]
    [-u file [type]]");
    #ifdef LOFI
    printf (" [-w lowe] [-w
    eweId] [-z size] ");
    #endif
}
```

```
void usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
    name);
    #ifdef LOFI
    printf ("[-g] [-d] ");
    #endif
    printf ("[-p what] [-r]
    [-u file [type]]");
    #ifdef LOFI
    printf (" [-w lowe] [-w
    eweId] [-z size] ");
    #endif
}
```

```
void usage (char *name)
{
    printf ("usage:\n");
    printf ("%s -a [-c file",
    name);
    #ifdef LOFI
    printf ("[-g] [-d] ");
    #endif
    printf ("[-p what] [-r]
    [-u file [type]]");
    #ifdef LOFI
    printf (" [-w lowe] [-w
    eweId] [-z size] ");
    #endif
}
```



SLOTH



Sample DB application code

```
handleRequest (name) {  
    p = getPatient(name);  
  
    t = getBodyTemp(p);  
    tc = (t - 32) * 5 / 9;  
  
    visits = getVisits(p);  
    meds = getMeds(p);  
  
    printHTML(tc, visits, meds);  
}
```

4 round trips

 Issues query to DB

Sample DB application code

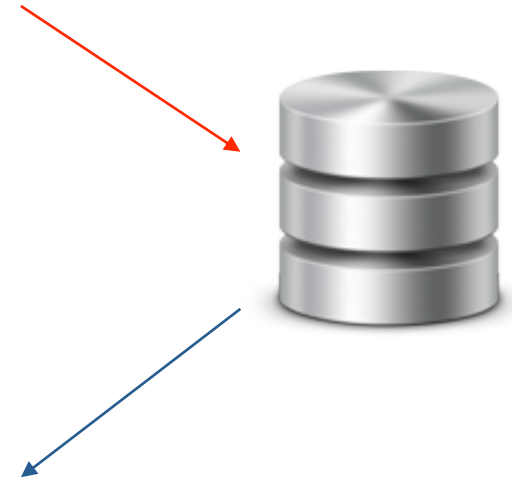
```
handleRequest (name) {  
    p = getPatient(name);  
    t = getBodyTemp(p);  
    tc = (t - 32) * 5 / 9;  
    visits = getVisits(p);  
    meds = getMeds(p);  
  
    printHTML(tc, visits, meds);  
}
```

 Issues query to DB

```
handleRequest (name) {  
    p = getPatient(name);
```

Batch

```
    t = getBodyTemp(p);  
    visits = getVisits(p);  
    meds = getMeds(p);
```



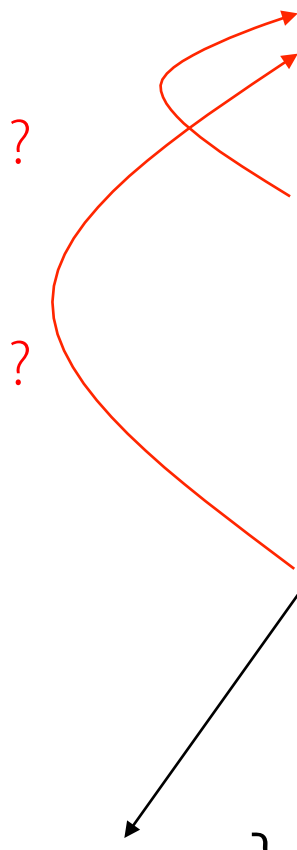
```
    tc = (t - 32) * 5 / 9;  
    printHTML(tc, visits, meds);
```

```
}
```

Ibrahim et al, ECOOP 06

Ramachandra et al, SIGMOD 12

```
handleRequest (name) {  
    p = getPatient(name);  
  
    if (canViewVitals) {  
        t = getBodyTemp(p);  
        tc = (t - 32) * 5 / 9;  
    }  
  
    visits = getVisits(p);  
    meds = getMeds(p);  
    printHTML(tc, visits, meds);  
}
```



```
List getVisits (Patient p) {  
    return executeQuery(p.id);  
}
```

```
// test harness  
List getVisits (Patient p)  
{ return TEST_VISITS; }
```


SLOTH

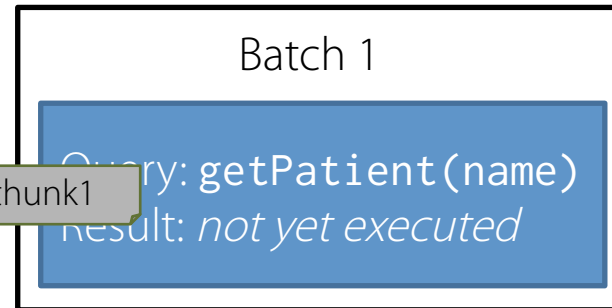


"DO IT ONLY WHEN YOU NEED IT"

Extended Lazy Evaluation

Query Store

```
handleRequest (name) {  
  
  p = get think1 (name);  
  
  if (canViewVitals) {  
    t = getBodyTemp(p);  
    tc = (t - 32) * 5 / 9;  
  }  
  
  visits = getVisits(p);  
  meds = getMeds(p);  
  
  printHTML(tc, visits, meds);  
}
```



 Issues query to DB

Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```

```
  if (canViewVitals) {
```

```
    t = getBodyTemp(p, thunk1);
```

```
    tc = (t - 32) * 5 / 9;
```

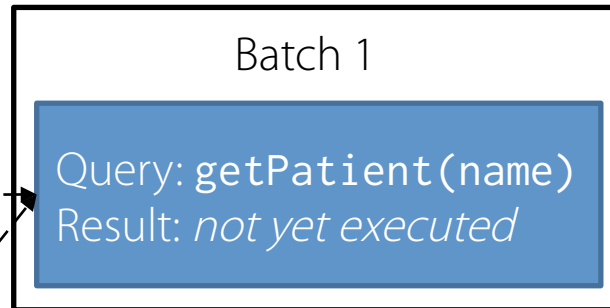
```
  }
```

```
  visits = getVisits(p);
```

```
  meds = getMeds(p);
```

```
  printHTML(tc, visits, meds);
```

```
}
```



?

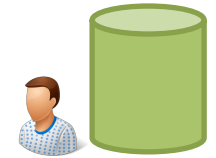
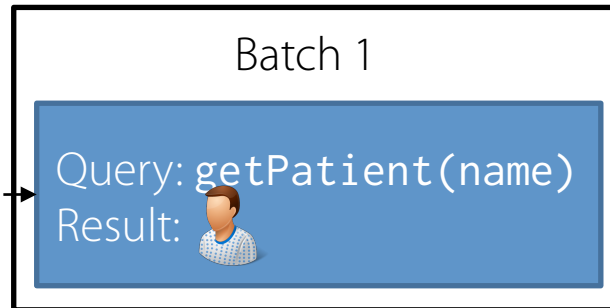
 Issues query to DB

Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```



```
  if (canViewVitals) {
```

```
    t = getBodyTemp( );
```

```
    tc = (t - 32) * 5 / 9;
```

```
  }
```

```
  visits = getVisits(p);
```


```
  meds = getMeds(p);
```

```
  printHTML(tc, visits, meds);
```

```
}
```


 Issues query to DB

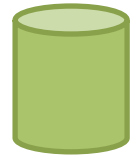
Query Store

```
handleRequest (name) {  
  
  p =  
  
  if (canViewVitals) {  
    t = getBodyTemp();  
    tc = (t - 32) * 5 / 9;  
  }  
  
  visits = getVisits(p);  
  meds = getMeds(p);  
  
  printHTML(tc, visits, meds);  
}
```

thunk1


Batch 1

Query: `getPatient(name)`
Result: 



thunk2

Batch 2

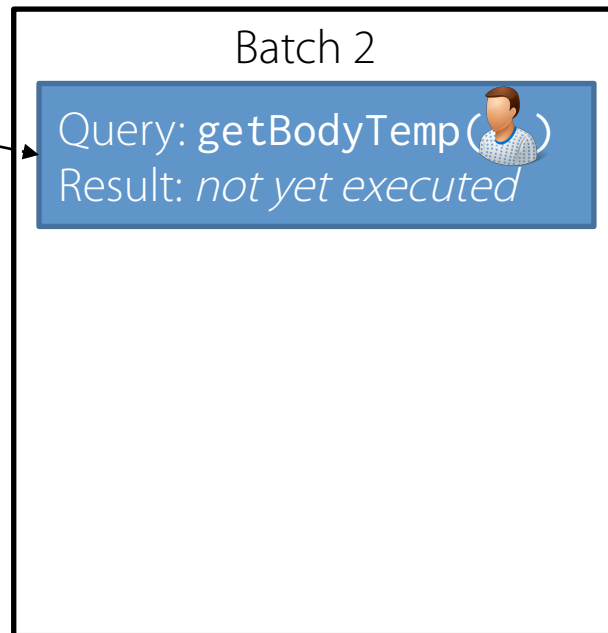
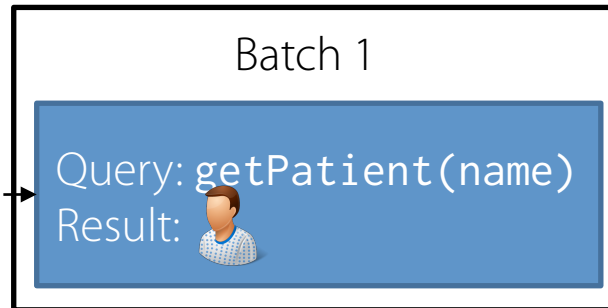
Query: `getBodyTemp()`
Result: *not yet executed*



Issues query to DB

Query Store

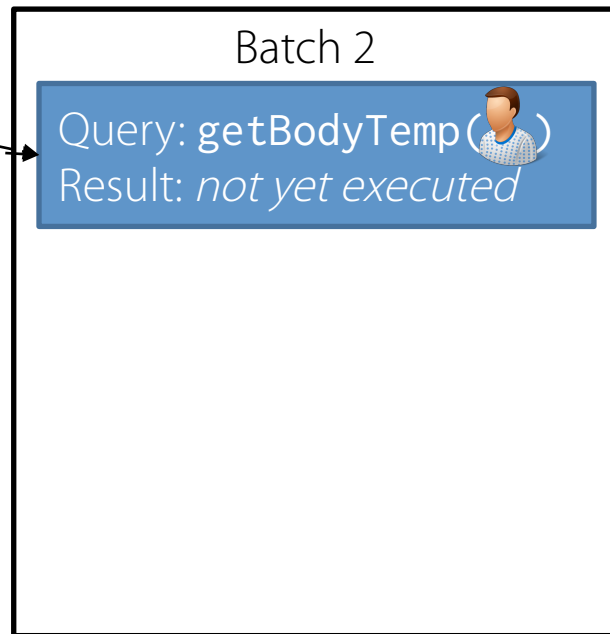
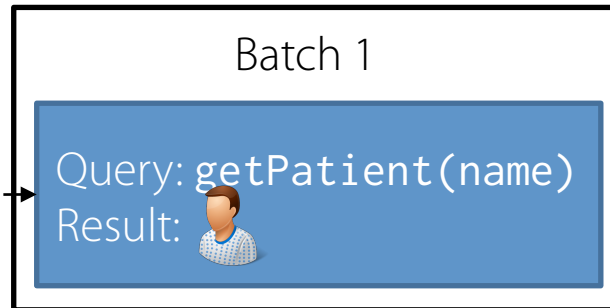
```
handleRequest (name) {  
  
  p = thunk1  
  
  if (canViewVitals) {  
    t = thunk2  
    tc = (t - 32) * 5 / 9;  
  }  
  
  visits = getVisits(p);  
  meds = getMeds(p);  
  
  printHTML(tc, visits, meds);  
}
```



█ Issues query to DB


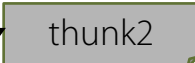
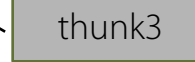
Query Store

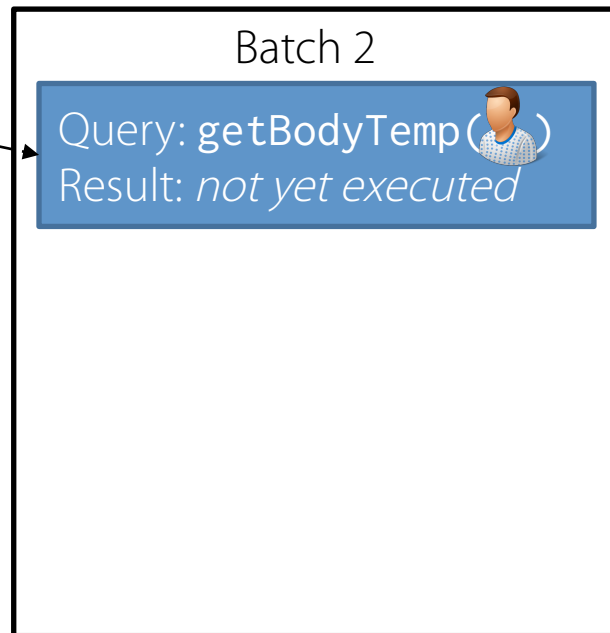
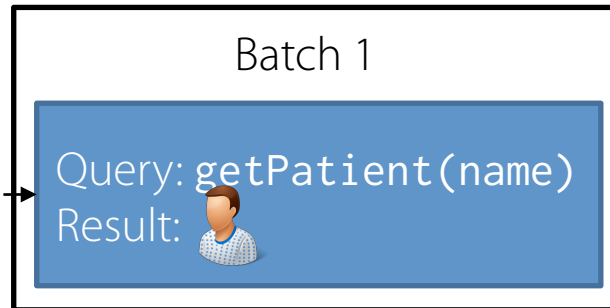
```
handleRequest (name) {  
  
  p = thunk1  
  
  if (canViewVitals) {  
    t = thunk2  
    tc = (thunk2 = 32) * 5 / 9;  
  }  
  
  visits = getVisits(p);  
  meds = getMeds(p);  
  
  printHTML(tc, visits, meds);  
}
```



█ Issues query to DB


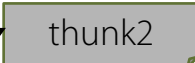
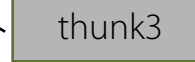
Query Store

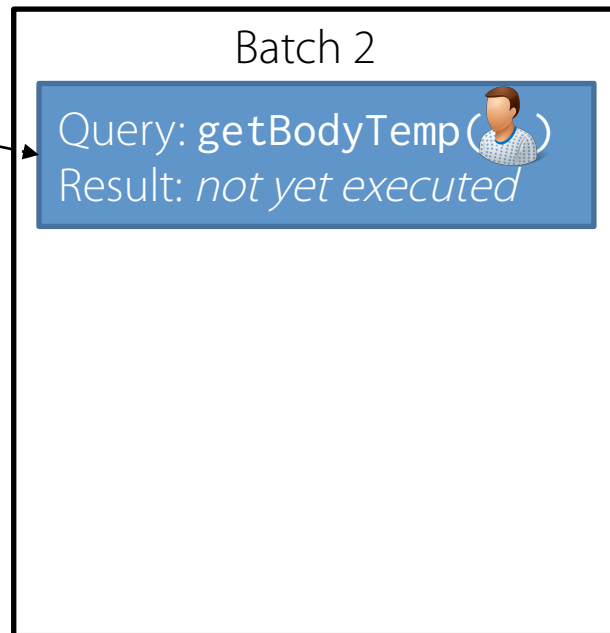
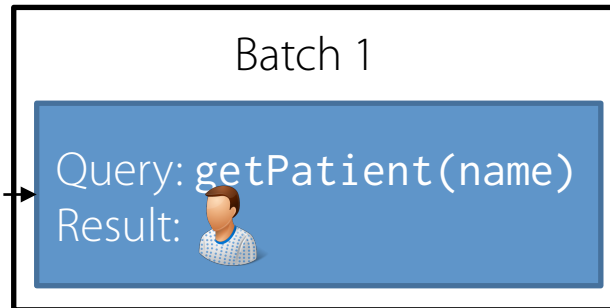
```
handleRequest (name) {  
  
  p =  thunk1  
  
  if (canViewVitals) {  
    t =  thunk2  
    tc =  thunk3  
  }  
  
  visits = getVisits(p);  
  meds = getMeds(p);  
  
  printHTML(tc, visits, meds);  
}
```



 Issues query to DB

Query Store

```
handleRequest (name) {  
  
  p =  thunk1  
  
  if (canViewVitals) {  
    t =  thunk2  
    tc =  thunk3  
  }  
  
  visits = getVisits(p);  
  meds = getMeds(p);  
  
  printHTML(tc, visits, meds);  
}
```



 Issues query to DB

Query Store

```
handleRequest (name) {
```

```
  p = 
```

```
  if (canViewVitals) {
```

```
    t = 
```

```
    tc = 
```

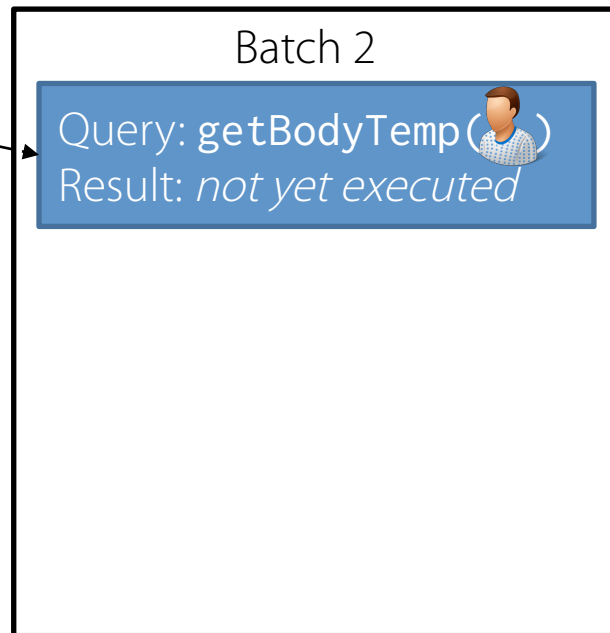
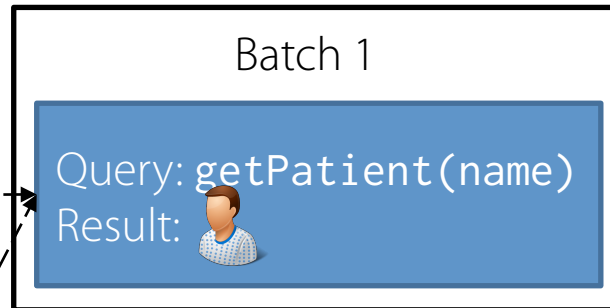
```
  }
```

```
  visits = getVisits();
```

```
  meds = getMeds(p);
```

```
  printHTML(tc, visits, meds);
```

```
}
```

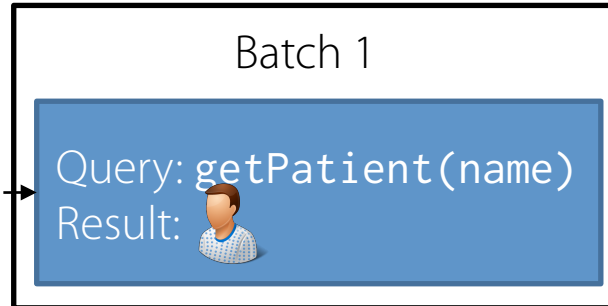


 Issues query to DB

Query Store

```
handleRequest (name) {
```

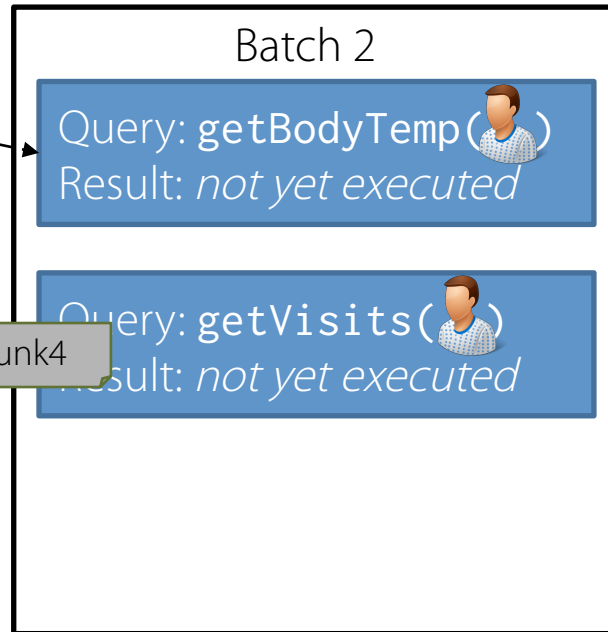
```
  p = 
```



```
  if (canViewVitals) {
```

```
    t = 
```

```
    tc = 
```



```
  visits = getVisits(  );
```


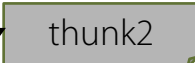
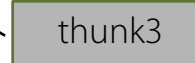
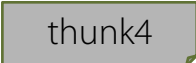
```
  meds = getMeds(p);
```

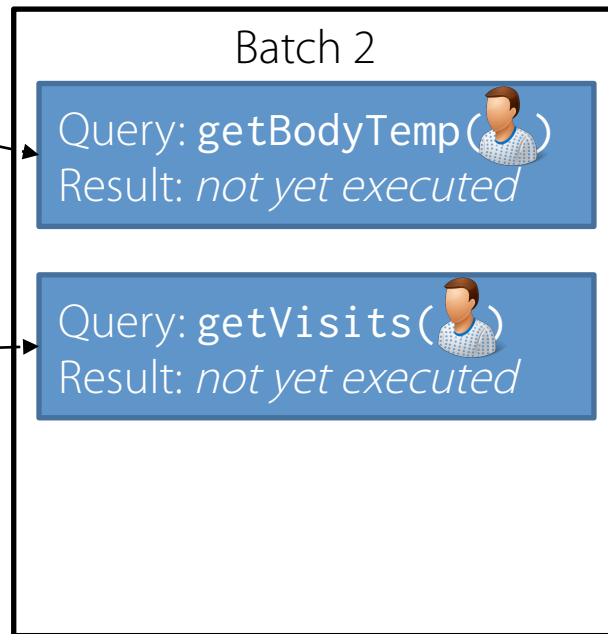
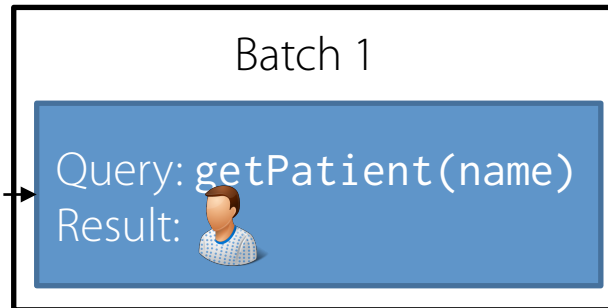
```
  printHTML(tc, visits, meds);
```

```
}
```

 Issues query to DB

Query Store

```
handleRequest (name) {  
  p =    
  
  if (canViewVitals) {  
    t =    
    tc =    
  }  
  
  visits =    
  meds = getMeds(p);  
  
  printHTML(tc, visits, meds);  
}
```

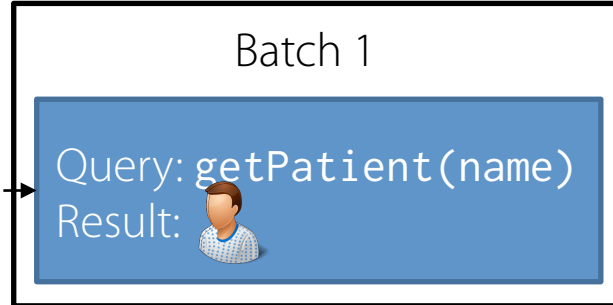


 Issues query to DB

Query Store

```
handleRequest (name) {
```

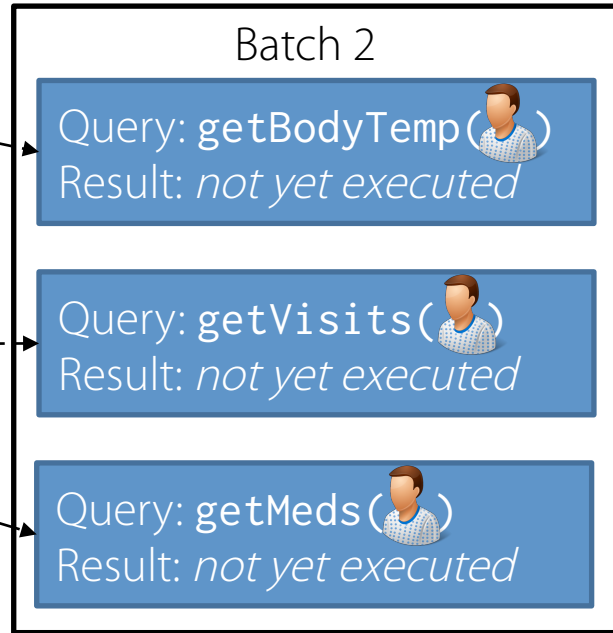
```
  p =  thunk1
```



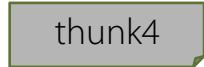
```
  if (canViewVitals) {
```

```
    t =  thunk2
```

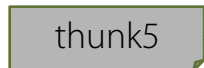
```
    tc =  thunk3
```



```
  visits =
```

```
     thunk4
```

```
  meds =
```

```
     thunk5
```

```
  printHTML(tc, visits, meds);
```

```
}
```

Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```

```
  if (canViewVitals) {
```

```
    t =
```

```
      thunk2
```

```
    tc =
```

```
      thunk3
```

```
  }
```

```
  visits =
```

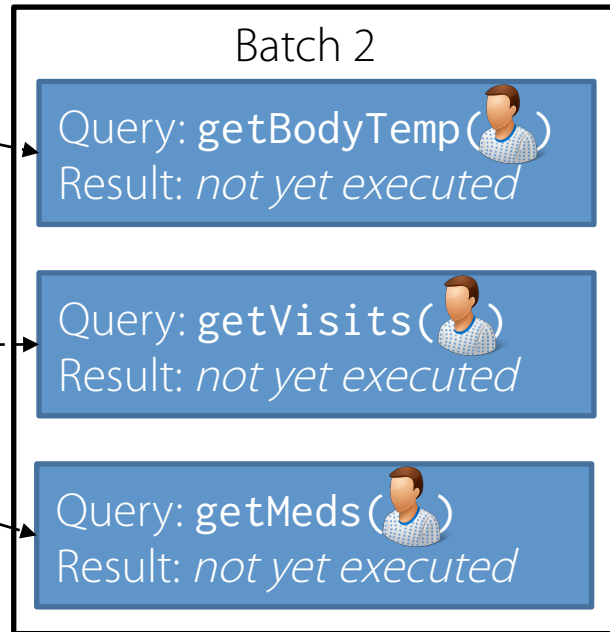
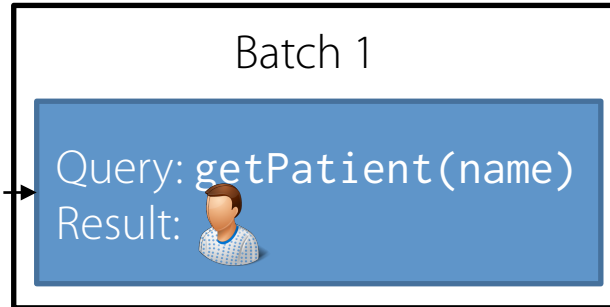
```
    thunk4
```

```
  meds =
```

```
    thunk5
```

```
  printHTML(thunk3, visits,  
            meds);
```

```
}
```

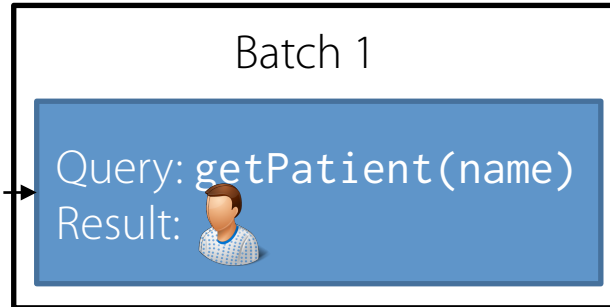


Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```



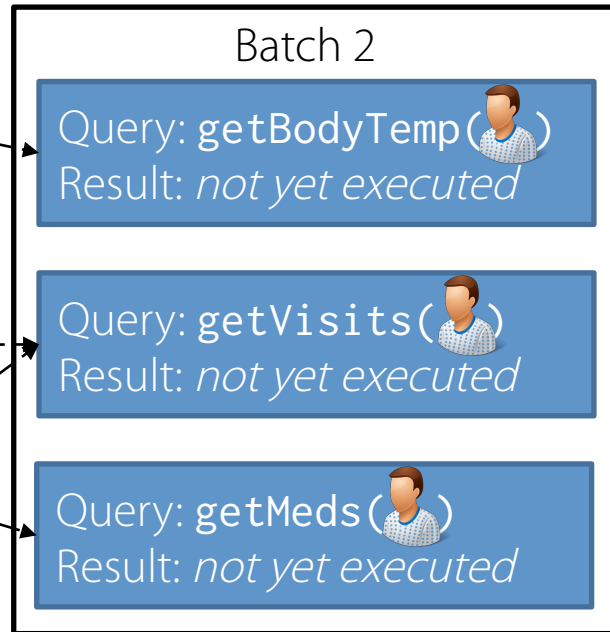
```
  if (canViewVitals) {
```

```
    t =
```

```
      thunk2
```

```
    tc =
```

```
      thunk3
```



```
  visits =
```

```
    thunk4
```

```
  meds =
```

```
    thunk5
```

```
  printHTML(thunk3, thunk4, meds);
```

```
}
```

Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```

```
  if (canViewVitals) {
```

```
    t =
```

```
      thunk2
```

```
    tc =
```

```
      thunk3
```

```
  }
```

```
  visits =
```

```
    thunk4
```

```
  meds =
```

```
    thunk5
```

```
  printHTML (
```

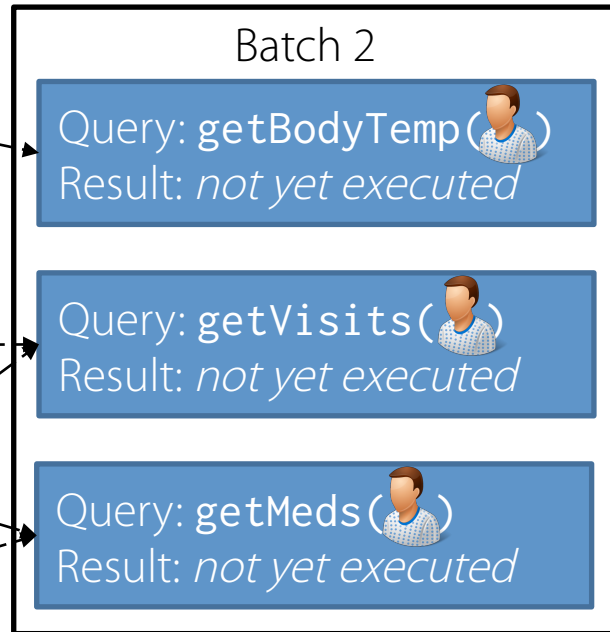
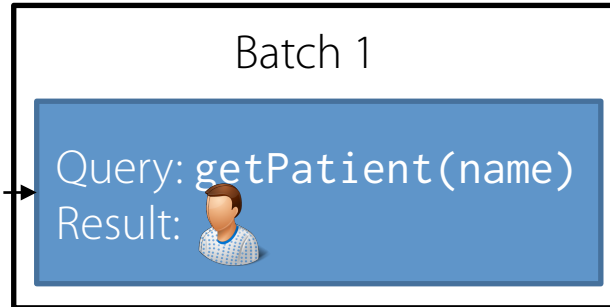
```
    thunk3
```

```
    , thunk4
```

```
    , thunk5
```

```
  );
```

```
}
```



Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```

```
  if (canViewVitals) {
```

```
    t =
```

```
      thunk2
```

```
    tc =
```

```
      thunk3
```

```
  }
```

```
  visits =
```

```
    thunk4
```

```
  meds =
```

```
    thunk5
```

```
  printHTML (
```

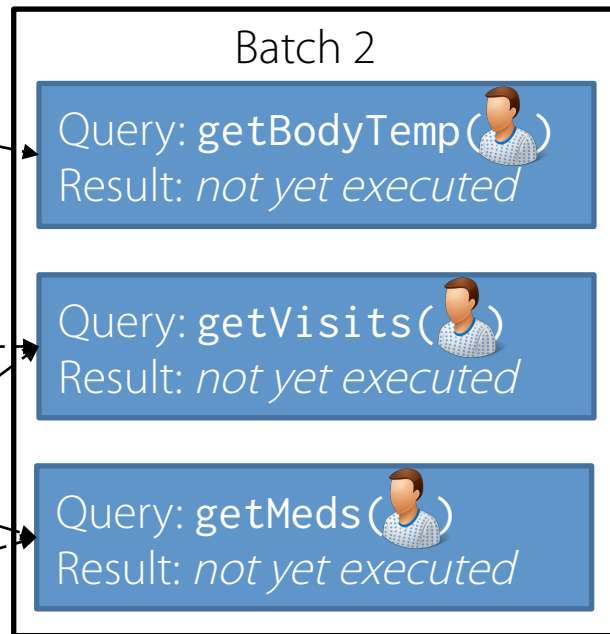
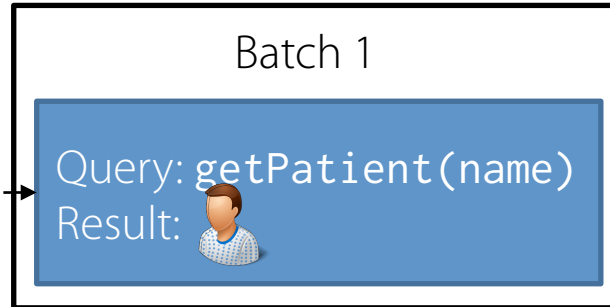
```
    thunk3
```

```
    , thunk4
```

```
    , thunk5
```

```
  );
```

```
}
```

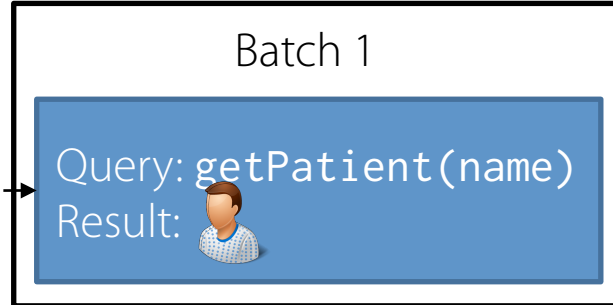


Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```

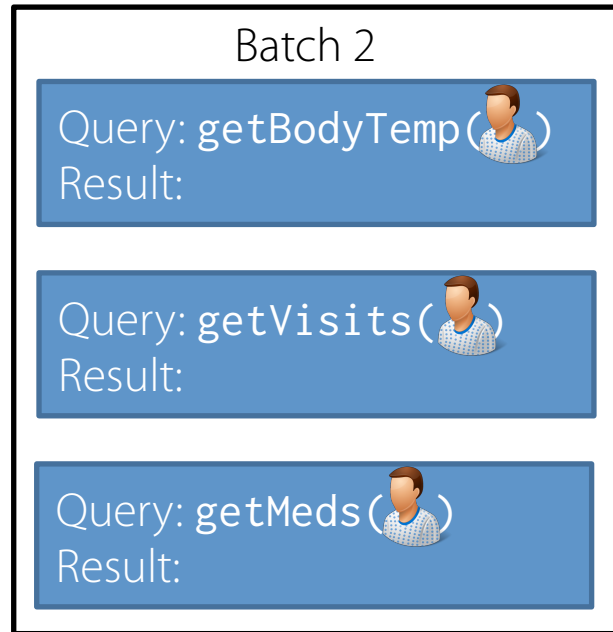


```
  if (canViewVitals) {
```

```
    t =
```

```
    tc =
```

```
  }
```



```
  visits =
```

```
  meds =
```

```
  printHTML(
```

```
    ,
```

```
    ,
```

```
  );
```

```
}
```

Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```

```
  if (canViewVitals) {
```

```
    t =
```

```
    tc =
```

```
  }
```

```
  visits =
```

```
  meds =
```

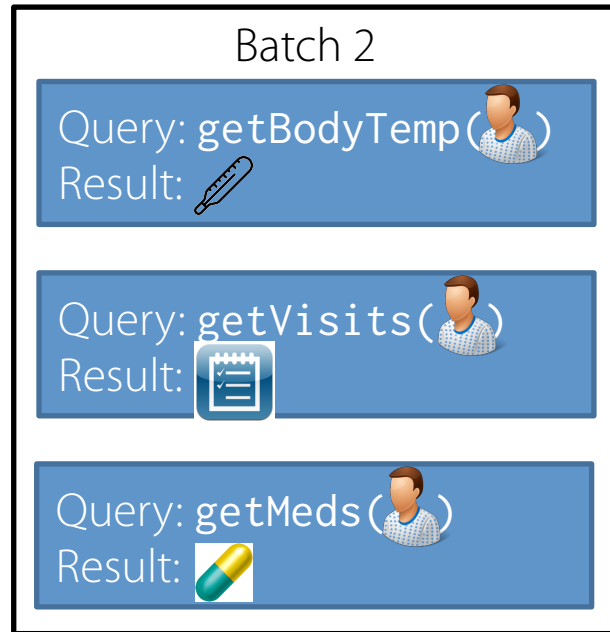
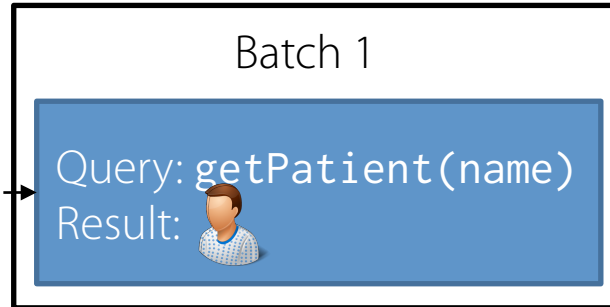
```
  printHTML(
```

```
    ,
```

```
    ,
```

```
  );
```

```
}
```




Query Store

```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```

```
  if (canViewVitals) {
```

```
    t = 
```

```
    tc =
```

```
  }
```

```
  visits =
```

```
  meds =
```

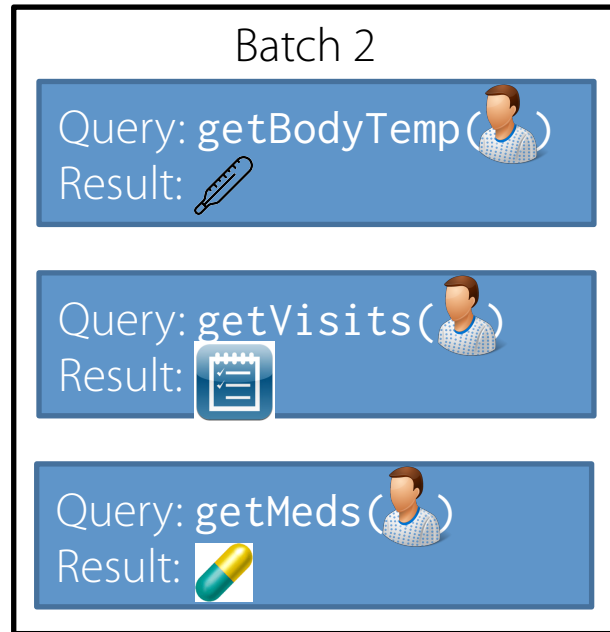
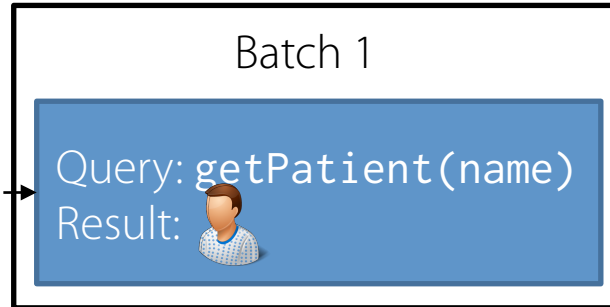
```
  printHTML(
```

```
    ,
```

```
    ,
```

```
  );
```

```
}
```




Query Store


```
handleRequest (name) {
```

```
  p =
```

```
    thunk1
```

```
  if (canViewVitals) {
```

```
    t = 
```

```
    tc = ( - 32) * 5 / 9;
```

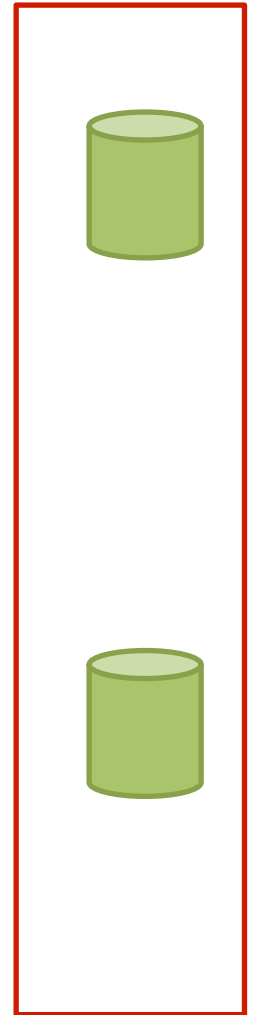
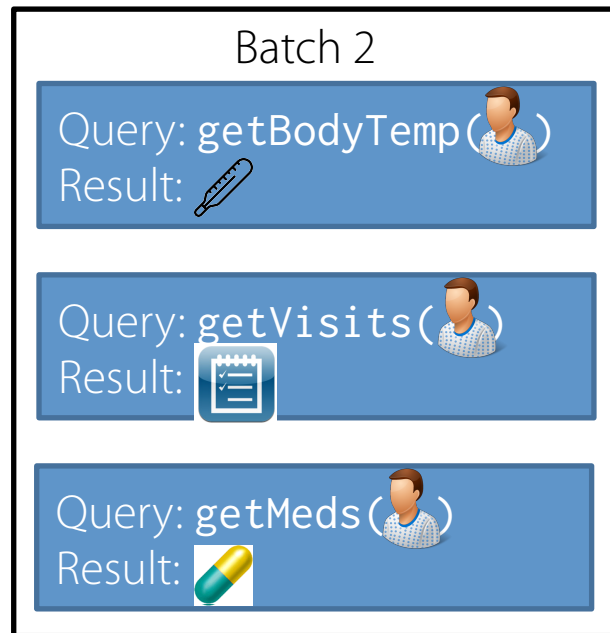
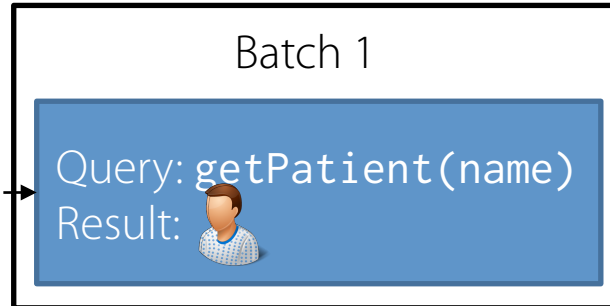
```
  }
```

```
  visits =
```

```
  meds =
```

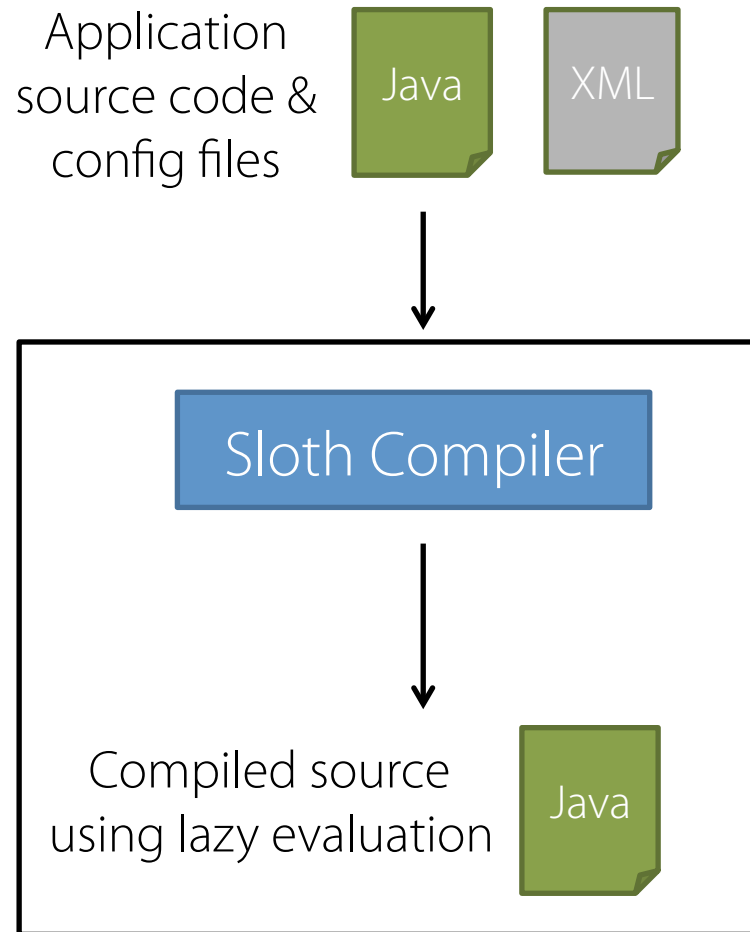
```
  printHTML( tc , ,
```

```
  }
```

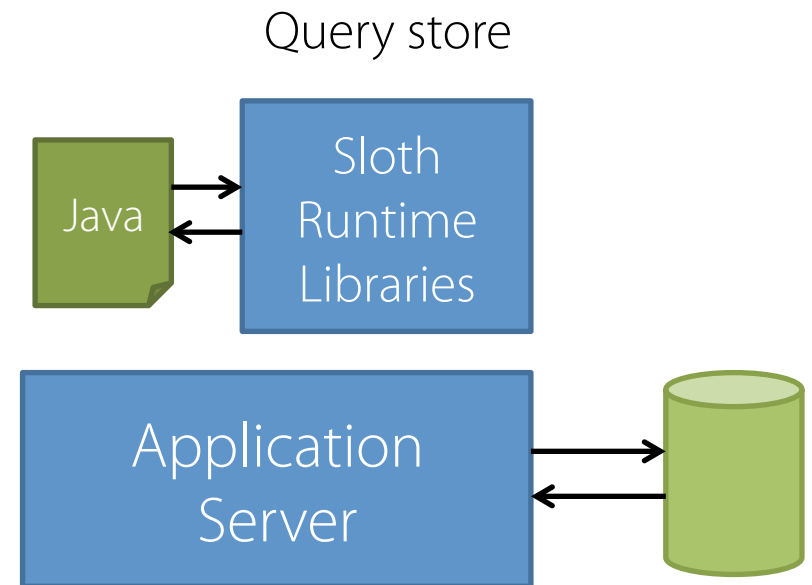


Only 2 round trips

Sloth Toolchain



Compilation



Execution

```
class Thunk<T>
{
    T force()
    {
        if (!evaluated) {
            r = <original program expression>
            evaluated = true;
        }
        return r;
    }
}
```

```
handleRequest (name) {
```

```
  p = getPatient(name);
```

```
  if (canViewVitals) {
```

```
    t = getBodyTemp(p);
```

```
    tc = (t - 32) * 5 / 9;
```

```
  }
```

```
  visits = getVisits(p);
```

```
  meds = getMeds(p);
```

```
  printHTML(tc, visits, meds);
```

```
}
```

```
p = new Thunk()
```

```
{
```

```
  { this.qid =  
    queryStore.reg("SELECT ..."); }  
};
```

```
  force() { return queryStore.get(qid); }  
};
```

```
};
```

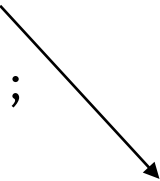
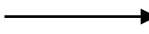
```
t = new Thunk(p)
```

```
{
```

```
  { p_forced = p.force();  
    this.qid = queryStore.reg(  
      "SELECT ...  
      WHERE id = " + p_forced.id);  
  }  
};
```

```
  force() { return queryStore.get(qid); }  
};
```

```
};
```



Issues query to DB


```
handleRequest (name) {
```

```
    p = getPatient(name);
```

```
    if (canViewVitals) {
```

```
        t = getBodyTemp(p);
```

```
        tc = (t - 32) * 5 / 9;
```

```
    }
```

```
    visits = getVisits(p);
```

```
    meds = getMeds(p);
```

```
    printHTML(tc, visits, meds);
```

```
}
```

```
    tc = new Thunk(t)
```

```
    {
```

```
        { this.t = t; }
```

```
    force()
```

```
    { return (t.force() - 32) * 5 / 9; }
```

```
};
```

 Issues query to DB

```
handleRequest (name) {
```

```
    p = getPatient(name);
```

```
    if (canViewVitals) {
```

```
        t = getBodyTemp(p);
```

```
        tc = (t - 32) * 5 / 9;
```

```
    }
```

```
    visits = getVisits(p);
```

```
    meds = getMeds(p);
```

```
    printHTML(tc, visits, meds);
```

```
}
```

```
    tc = new Thunk(t)
```

```
    {
```

```
        { this.t = t; }
```

```
    force()
```

```
    { return (t.force() - 32) * 5 / 9; }
```

```
};
```

 Issues query to DB

When to force

Field writes

`x.f = y` \longrightarrow `x.force().f = y;`

Control flow

`if (c) { ... }` \longrightarrow `if (c.force()) { ... }`

Statements with externally-visible side effects

`print(x);` \longrightarrow `print(x.force());`

Data modification / transaction statements

`execute("INSERT INTO ...")`

\longrightarrow `queryStore.executeBatchedQueries();`
`queryStore.execute("INSERT INTO ...");`

Optimizations

Selective lazification

Coalesce thunks

Defer control flow blocks and
method calls

Defer control flow blocks

```
if (c)
  x = a;
else
  x = b;
```



```
if (c.force())
  x = new Thunk(a) { force() { return a.force(); } };
else
  x = new Thunk(b) { force() { return b.force(); } };
```

Basic



```
x = new Thunk(c, a, b) {
  force() {
    if (c.force())
      return a.force();
    else
      return b.force();
  }
};
```

Optimized

Experiments

Two large-scale open source web apps

Extracted 38 and 112 benchmarks

Hosted on tomcat and mysql

0.5ms round trip ping time

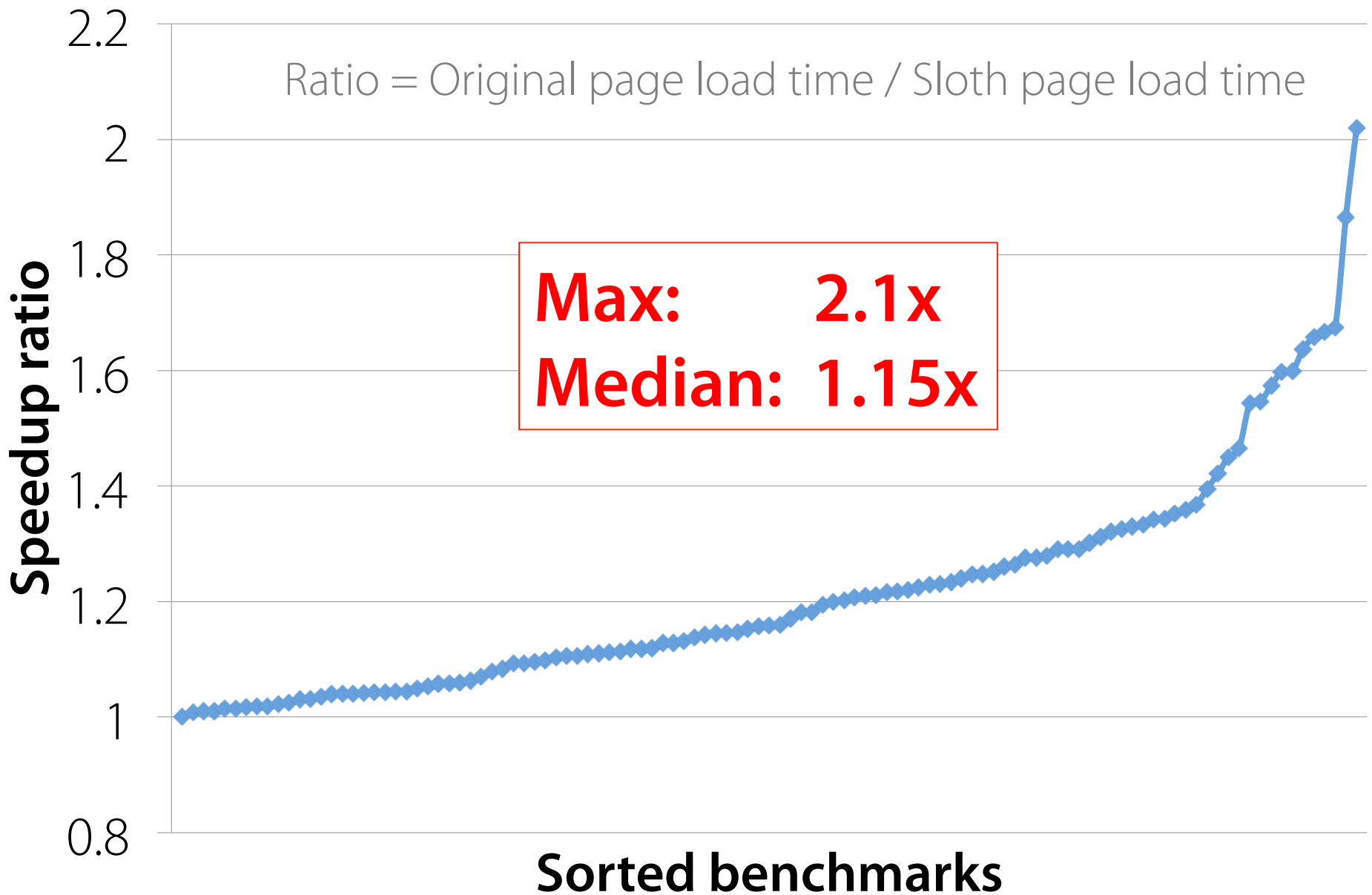
Goals

Page load times

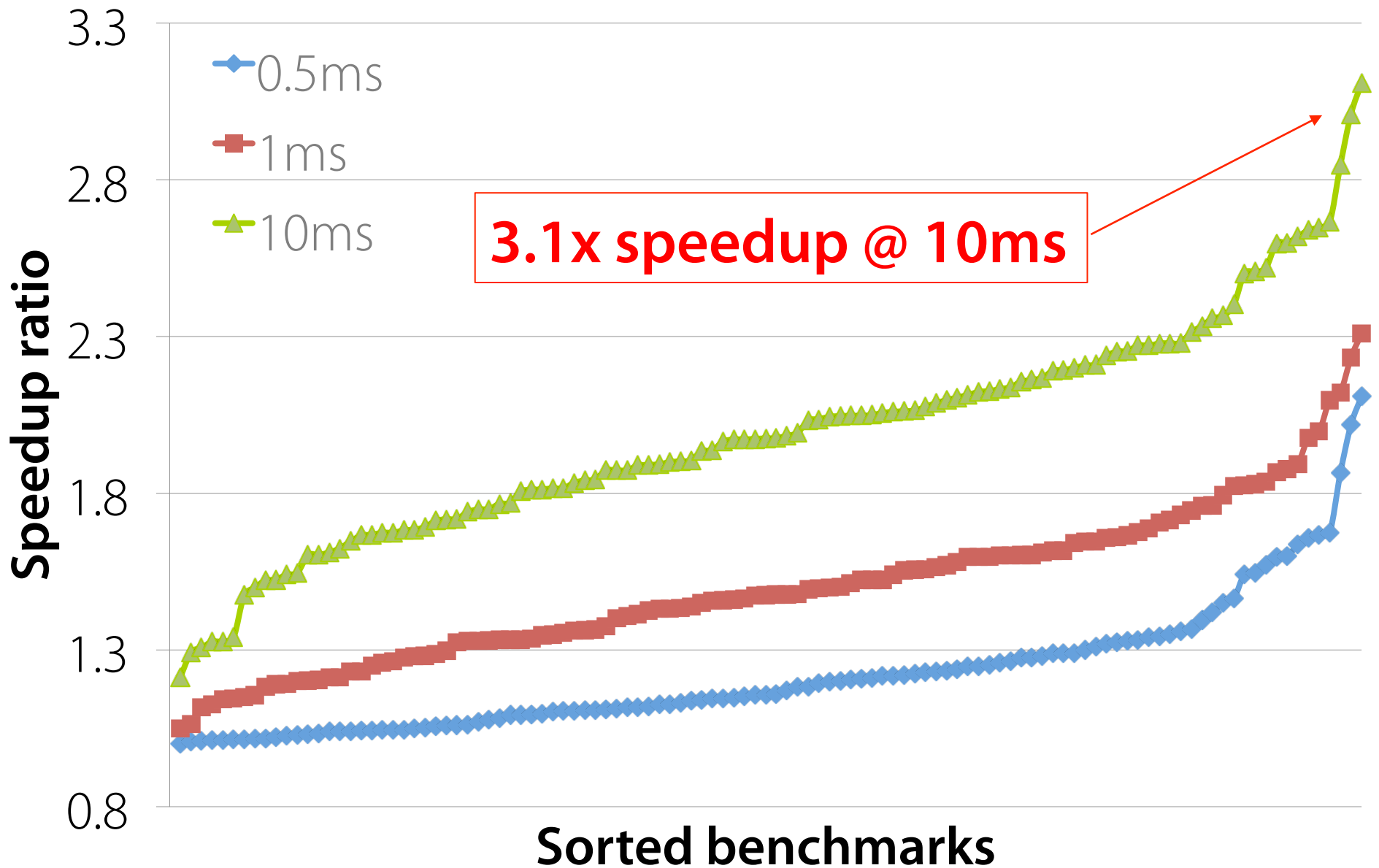
Scaling effects

Runtime overhead

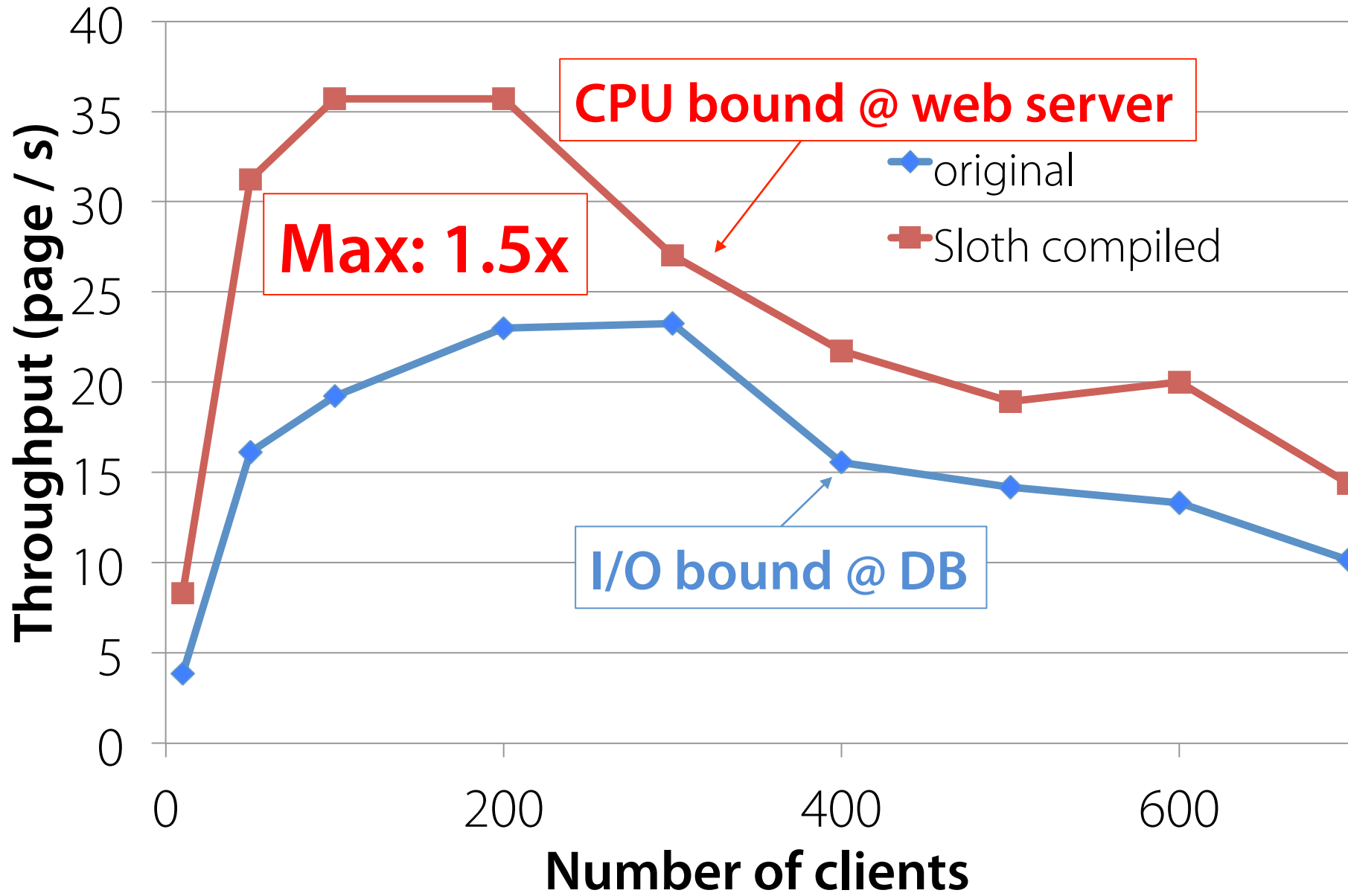
Sloth speeds up most benchmarks



Sloth performs even better when network round trip time increases



Batching queries increases throughput

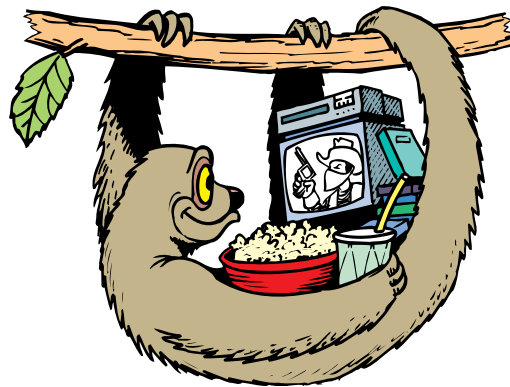


Runtime overhead

Txn type	Original (s)	Sloth (s)	Overhead %
TPC-C			
New order	930	955	15.8%
Order status	752	836	11.2%
Stock level	420	459	9.4%
Payment	789	869	10.2%
Delivery	626	665	6.2%
TPC-W			
Browsing mix	1075	1138	5.9%
Shopping mix	1223	1326	8.5%
Ordering mix	1423	1600	12.4%

6 – 16 %

SLOTH



Extended lazy evaluation

Batch queries issued by client

3x performance speedup

Alvin Cheung

akcheung@mit.edu