ZOMG!!!!!

HAPPY

THURSDAY!!

WELCOME TO

RubyConf X!

X-TREME

RUBYCONF!!!!!

Aaron Patterson

WWFMD?

Performance

ARel

Rails

Prepared Statement Caching

DEEPER
UNDERSTANDING
REQUIRED

ActiveRecord 5x slower than Rails 2.3.5

http://bit.ly/omgslow

5x Slower?!?

WTF?

Yes, 5x Slower

What could possibly go wrong?

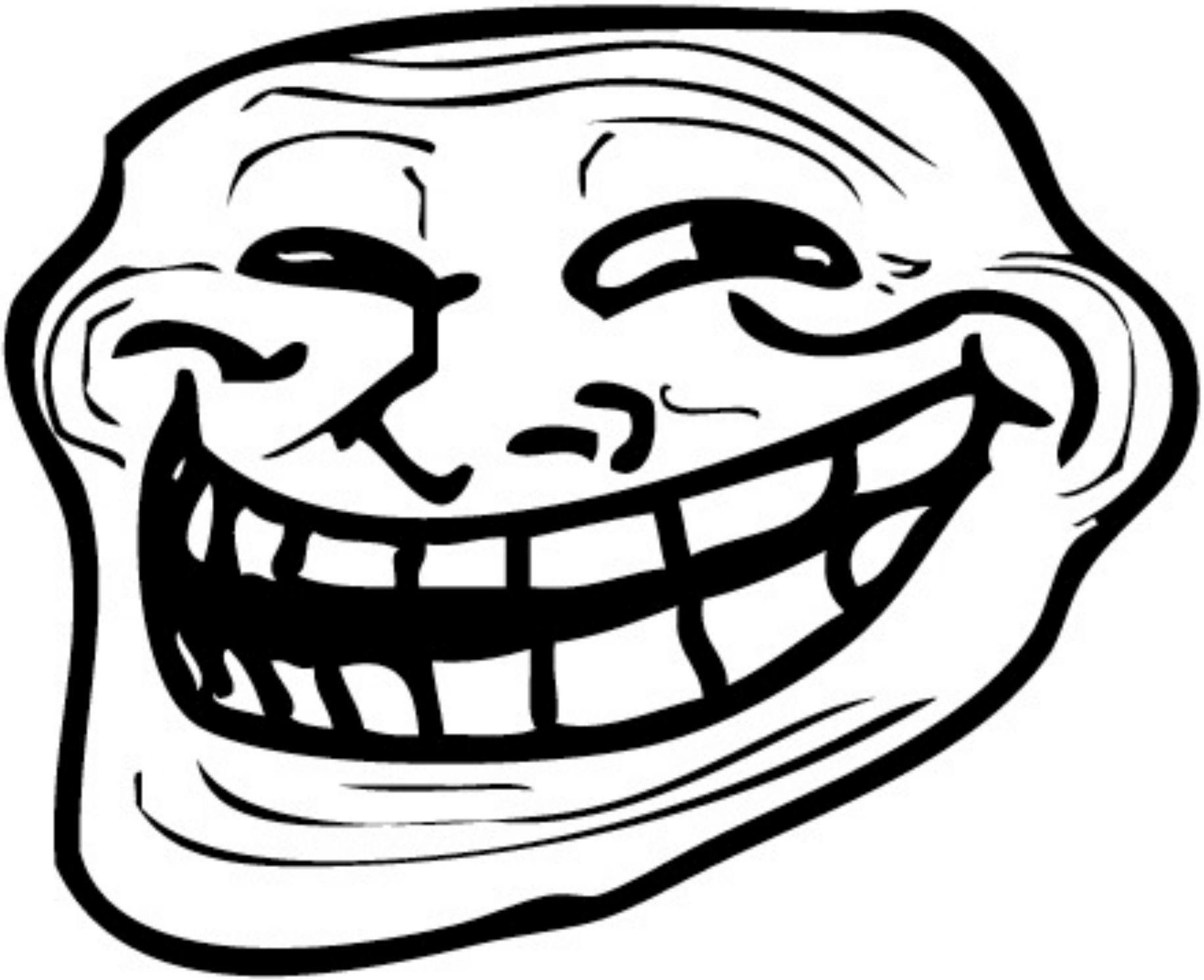
Motivation

Why do you care about speed?

Scaling Ruby

query_methods.rb (~/git/rails/a...lib/active_record/relation) - VIM1

```
        arel.join(join)
      end

      arel.joins(arel)
    end

    def build_arel
      arel = table

      arel = build_joins(arel, @joins_values) unless @joins_values.empty?

      (@where_values - ['']).uniq.each do |where|
        where = Arel.sql(where) if String === where
        arel = arel.where(Arel::Nodes::Grouping.new(where))
      end

      arel = arel.having(*@having_values.uniq.reject{|h| h.blank?}) unless @havi
ng_values.empty?

      arel = arel.take(@limit_value) if @limit_value
      arel = arel.skip(@offset_value) if @offset_value

      arel = arel.group(*@group_values.uniq.reject{|g| g.blank?}) unless @group_
values.empty?

      arel = arel.order(*@order_values.uniq.reject{|o| o.blank?}) unless @order_
values.empty?

      arel = build_select(arel, @select_values.uniq)

      arel = arel.from(@from_value) if @from_value
      arel = arel.lock(@lock_value) if @lock_value

      arel
    end

    private

query_methods.rb [2] 0-1, 195 of 267, (4572) 71% 0x0 (0)
```



When should I make my code faster?

When it isn't fast enough.

What is

"fast enough"?

Do people notice it?

In comparison to?

Finishes in a reasonable amount of time.

Don't believe me.

Think Critically

For Performance:

Things To Reduce

- Method calls
- Branching and looping
- Objects (memory consumption)

For Clean Code:

Things To Reduce

- Method calls
- Branching and looping
- Objects (memory consumption)

∴

(therefore)

Clean Code

≠

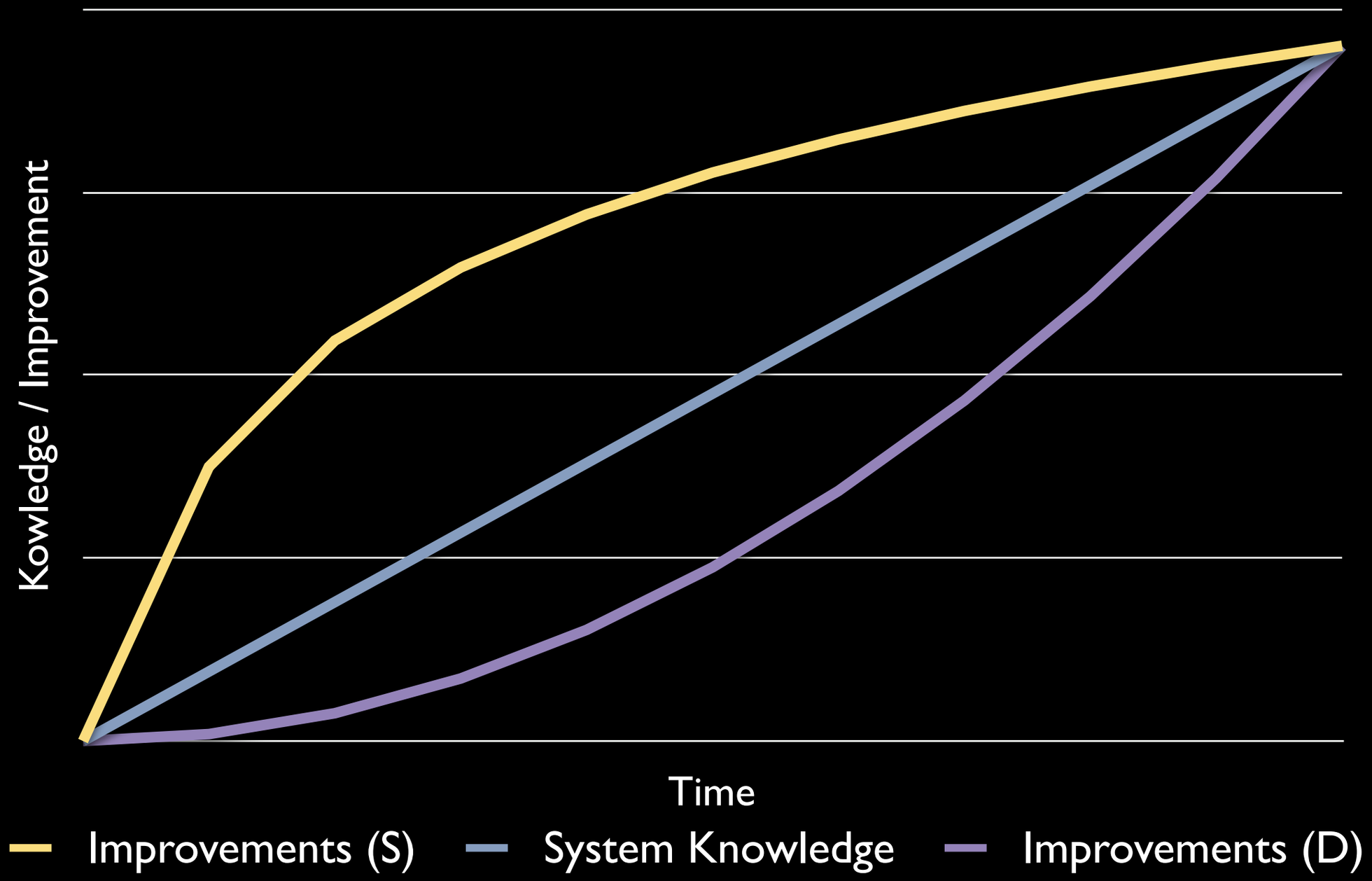
Performant code

Conclusion

AKA:

Things I've Learned

System Impact



When Should I Rewrite?

Rewrite Timeline

Rewrite Timeline

We emphasize

the art

of Code

We should not forget

the Science