

The Ghosts of Action View Cache

Hartley McGuire - RailsConf 2025

About me

@skipkayhil

- Rails Issues Team
- KateSQL @ Shopify (former)
- Rails Infra @ Shopify (current)



THE GHOSTS OF
ACTION VIEW CACHE



```
class ProductController < ApplicationController
  def index
    Rails.cache.fetch @product do
      Product.really_complex_scope
    end
  end
end
```

```
class ProductController < ApplicationController
  def index
    Rails.cache.fetch @product do
      Product.really_complex_scope.load
    end
  end
end
```

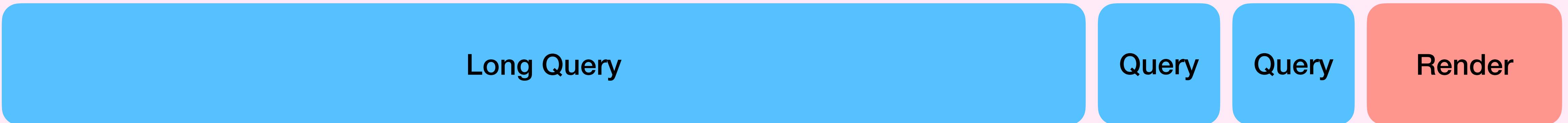
THE GHOST OF
ACTION VIEH
PAST

```
# app/views/products/index.html.erb
<% cache @product do %>
  <%= render @product %>
<% end %>
```

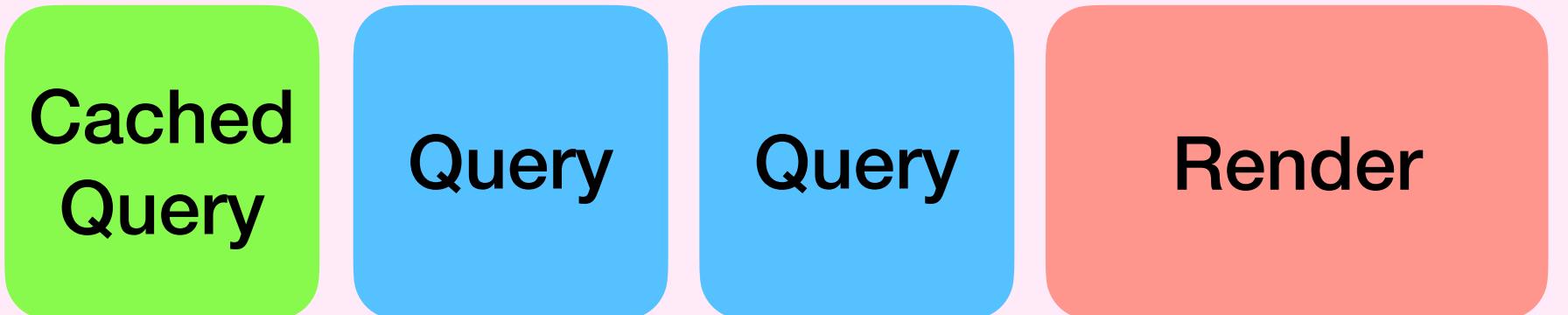
```
# app/views/products/_product.html.erb
<% expensive = Product.really_complex_scope %>
```

Request Time

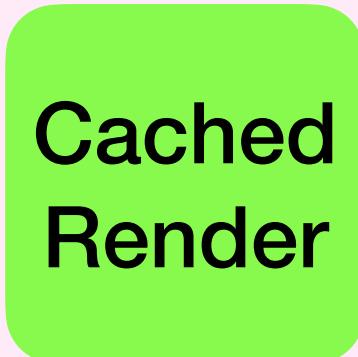
Uncached



Rails.cache



Fragment Cache



```
# app/views/products/index.html.erb
<% cache @product do %>
  <%= render @product %>
<% end %>
```

```
# app/views/products/_product.html.erb
<% expensive = Product.really_complex_scope %>
```

ActionView::Digester

ActionView::Digester

```
# app/views/products/show.html.erb
```

```
<%= render @product %>
```

```
<%= render @product.variants %>
```

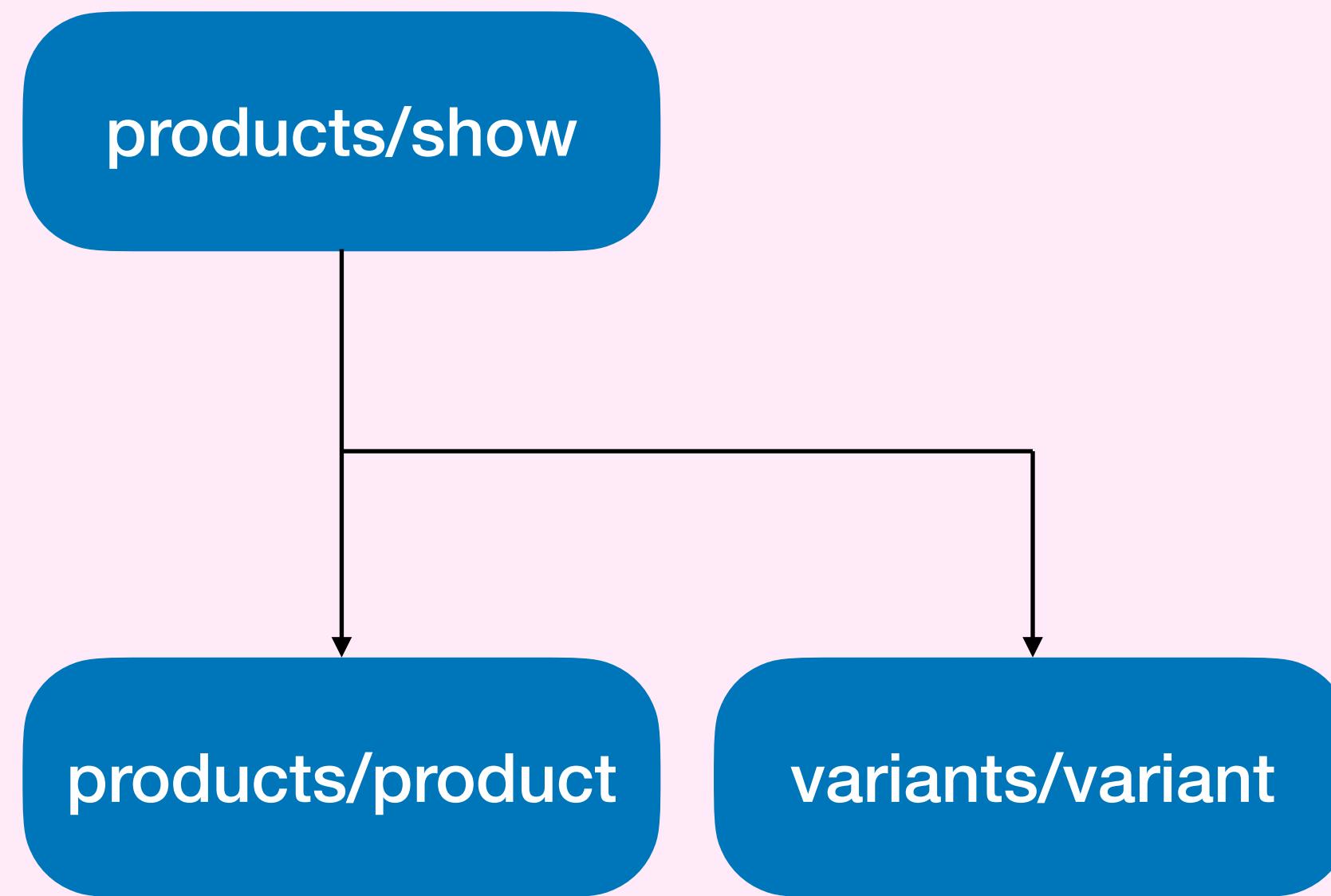
```
# app/views/products/_product.html.erb
```

```
<%= product.name %>
```

```
# app/views/variants/_variants.html.erb
```

```
<%= variant.name %>
```

ActionView::Digester



```
# app/views/products/show.html.erb
```

```
<%= render @product %>
```

```
<%= render @product.variants %>
```

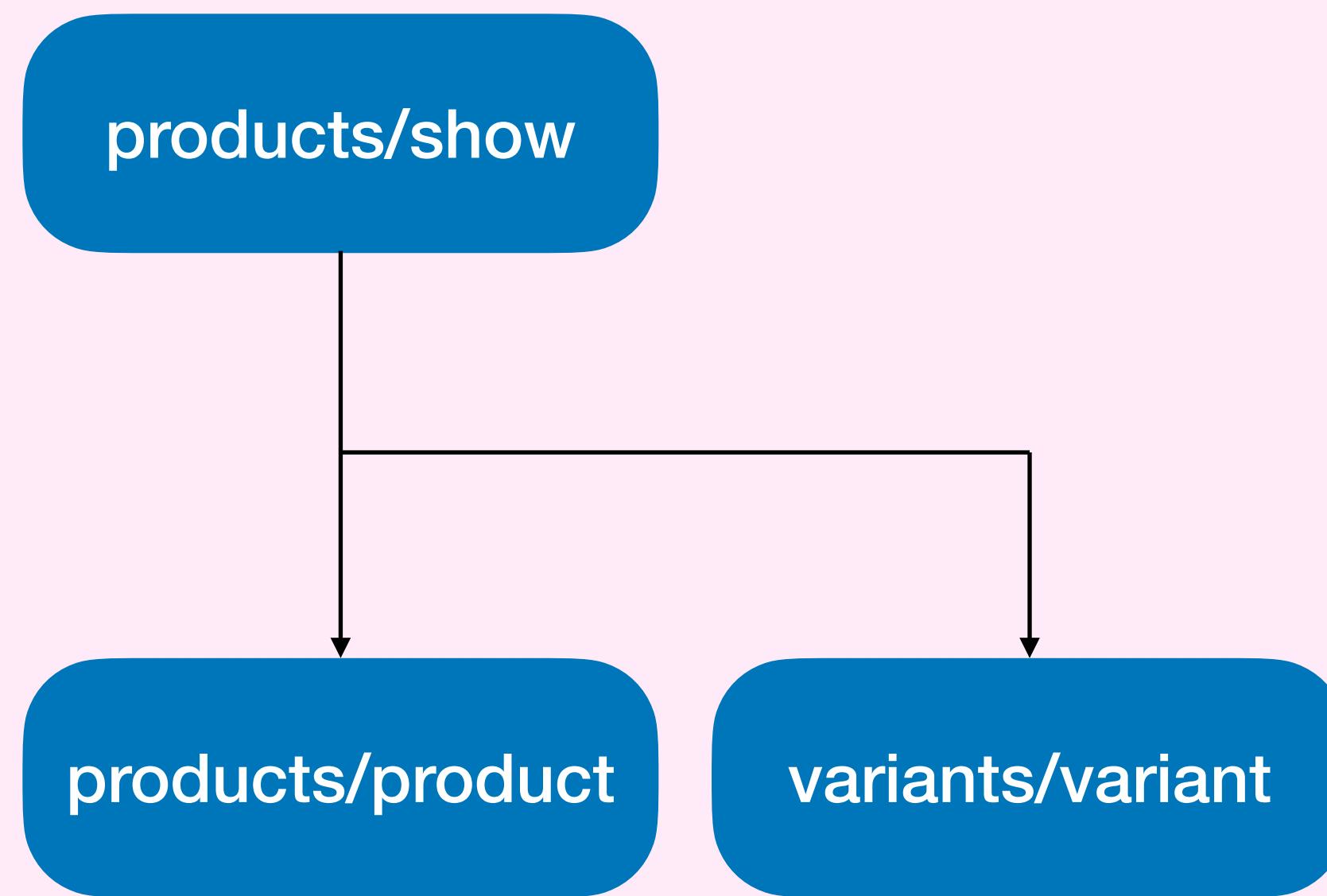
```
# app/views/products/_product.html.erb
```

```
<%= product.name %>
```

```
# app/views/variants/_variants.html.erb
```

```
<%= variant.name %>
```

ActionView::Digester



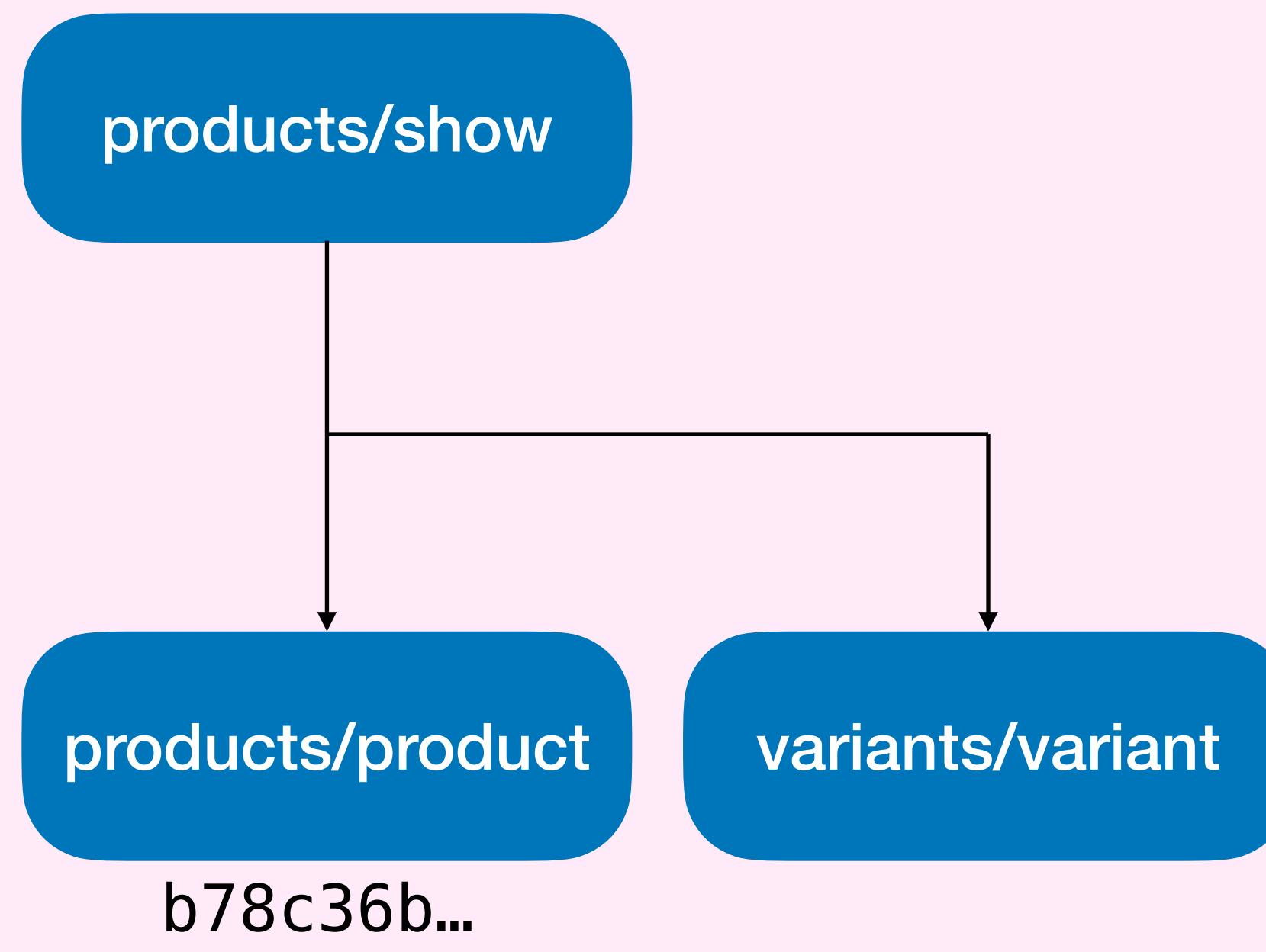
```
# app/views/products/show.html.erb
<%= render @product %>
<%= render @product.variants %>

# app/views/products/_product.html.erb
<%= product.name %>

# app/views/variants/_variants.html.erb
<%= variant.name %>
```

```
digest = hexdigest(template.source)
# => "b78c36bb799a8d8db87ee4b7fa836976"
```

ActionView::Digester



```
digest = hexdigest(template.source)
```

```
# => "418277a37d00530aa5230d6f0d0f9829"
```

```
# app/views/products/show.html.erb
```

```
<%= render @product %>
```

```
<%= render @product.variants %>
```

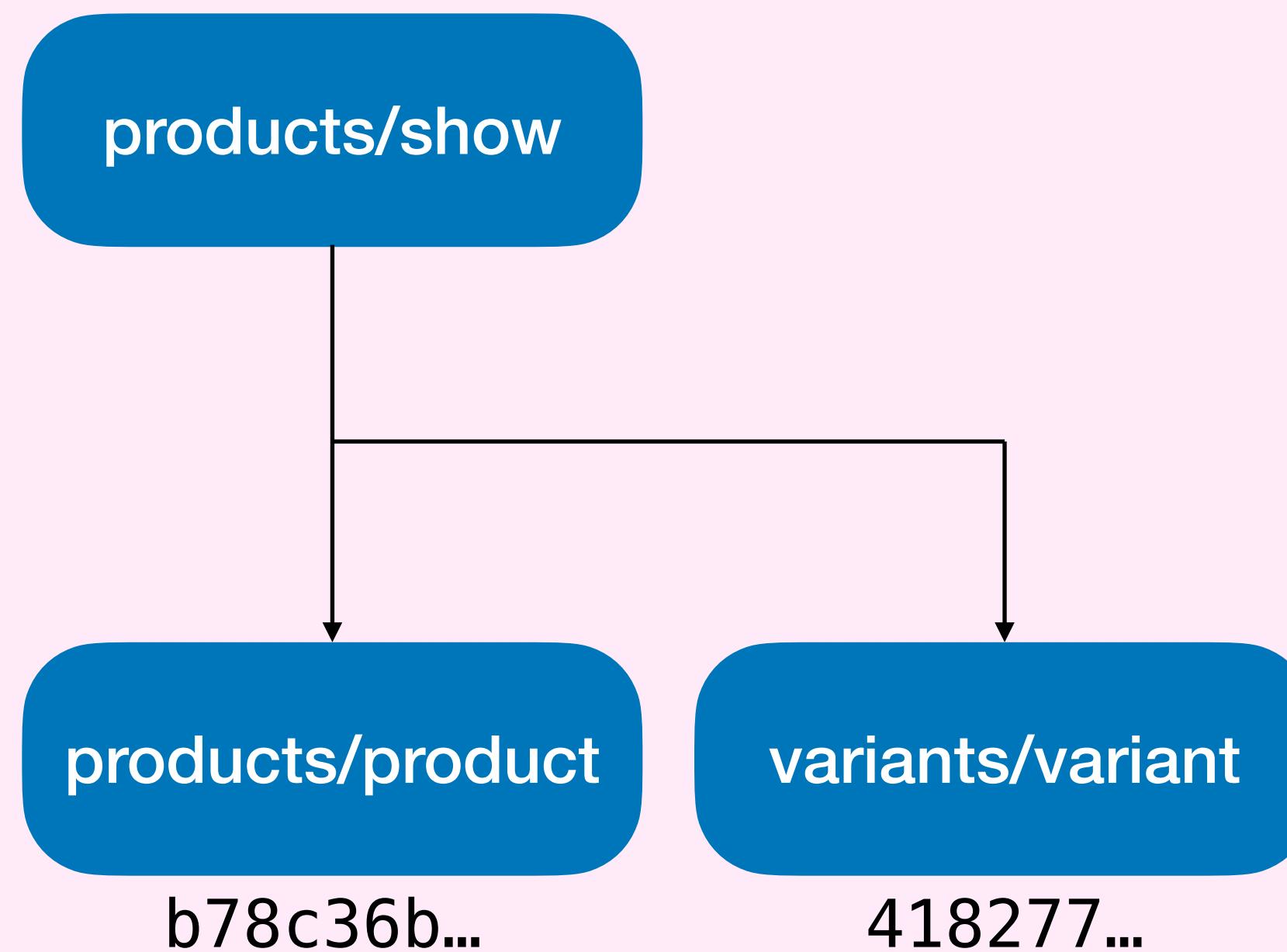
```
# app/views/products/_product.html.erb
```

```
<%= product.name %>
```

```
# app/views/variants/_variants.html.erb
```

```
<%= variant.name %>
```

ActionView::Digester



```
digest = hexdigest(  
  template.source +  
  children.map { it.digest }.join("_")  
)
```

```
# => "b279b5aef400f9f2e0a417ab9d9fb4dc"
```

```
# app/views/products/show.html.erb  
<%= render @product %>  
<%= render @product.variants %>  
  
# app/views/products/_product.html.erb  
<%= product.name %>  
  
# app/views/variants/_variants.html.erb  
<%= variant.name %>
```

ActionView::DependencyTracker::ERBTracker

```

class ERBTracker # :nodoc:
  EXPLICIT_DEPENDENCY = /* Template Dependency: (\S+)/

  # A valid ruby identifier - suitable for class, method and specially variable
  IDENTIFIER = /
    [[:alpha:]] # at least one uppercase letter, lowercase letter underscored
    [[:word:]]* # followed by optional letters, numbers or underscores
  /x

  # Any kind of variable name. e.g. @instance, @@class, $global or local.
  # Possibly following a method call chain
  VARIABLE_OR_METHOD_CHAIN = /
    (?:\$\{@\{1,2\}\})? # optional global, instance or class variable
    (?:#\{IDENTIFIER\}\.)* # followed by an optional chain of zero argument
    (?<dynamic>#\{IDENTIFIER\}) # and a final valid identifier, captured as DYNAMIC
  /x

  # A simple string literal. e.g. "School's out!"
  STRING = /
    (?<quote>[""]) # an opening quote
    (?<static>.*?) # with anything inside, captured as STATIC
    \k<quote> # and a matching closing quote
  /x

  # Part of any hash containing the :partial key
  PARTIAL_HASH_KEY = /
    (?:\blayout:|:partial\$\s*=>) # partial key in either old or new style hash
    \s*
  /x

  # Part of any hash containing the :layout key
  LAYOUT_HASH_KEY = /
    (?:\blayout:|:layout\$\s*=>) # layout key in either old or new style hash
    \s*
  /x

  # Matches:
  #   partial: "comments/comment", collection: @all_comments => "Comments/comment"
  #   (object: @single_comment, partial: "comments/comment") => "Comments/comment"
  #
  #   "comments/comments"
  #   'comments/comments'
  #   ('comments/comments')
  #
  #   (@topic)      => "topics/topic"
  #   topics        => "topics/topic"
  #   (message.topics) => "topics/topic"
  RENDER_ARGUMENTS = /\A
    (?:\s*\(\?\s*)
    (?::.*#\{PARTIAL_HASH_KEY\}|#\{LAYOUT_HASH_KEY\})? # optional hash, to the partial
    (?:#\{STRING\}|#\{VARIABLE_OR_METHOD_CHAIN\}) # finally, the dependency name
  /xm

  LAYOUT_DEPENDENCY = /\A
    (?:\s*\(\?\s*)
    (?::.*#\{LAYOUT_HASH_KEY\})
    (?:#\{STRING\}|#\{VARIABLE_OR_METHOD_CHAIN\}) # optional opening paren surround
  # optional opening paren surround
  # check if the file has layout
  # finally, the dependency name
end

```

**Digestor constructs trees of dependencies
using ERBTracker to parse renders.**

Fragment Caching: Cache Invalidation





3.1.2. Default Values for Target Version 7.2

- `config.active_record.postgresql_adapter_decode_dates: true`
- `config.active_record.validate_migration_timestamps: true`
- `config.active_storage.web_image_content_types: %w[image/png image/jpeg image/gif image/webp]`
- `config.yjit: true`

THE GHOST OF
ACTION VIEH
PRESENT

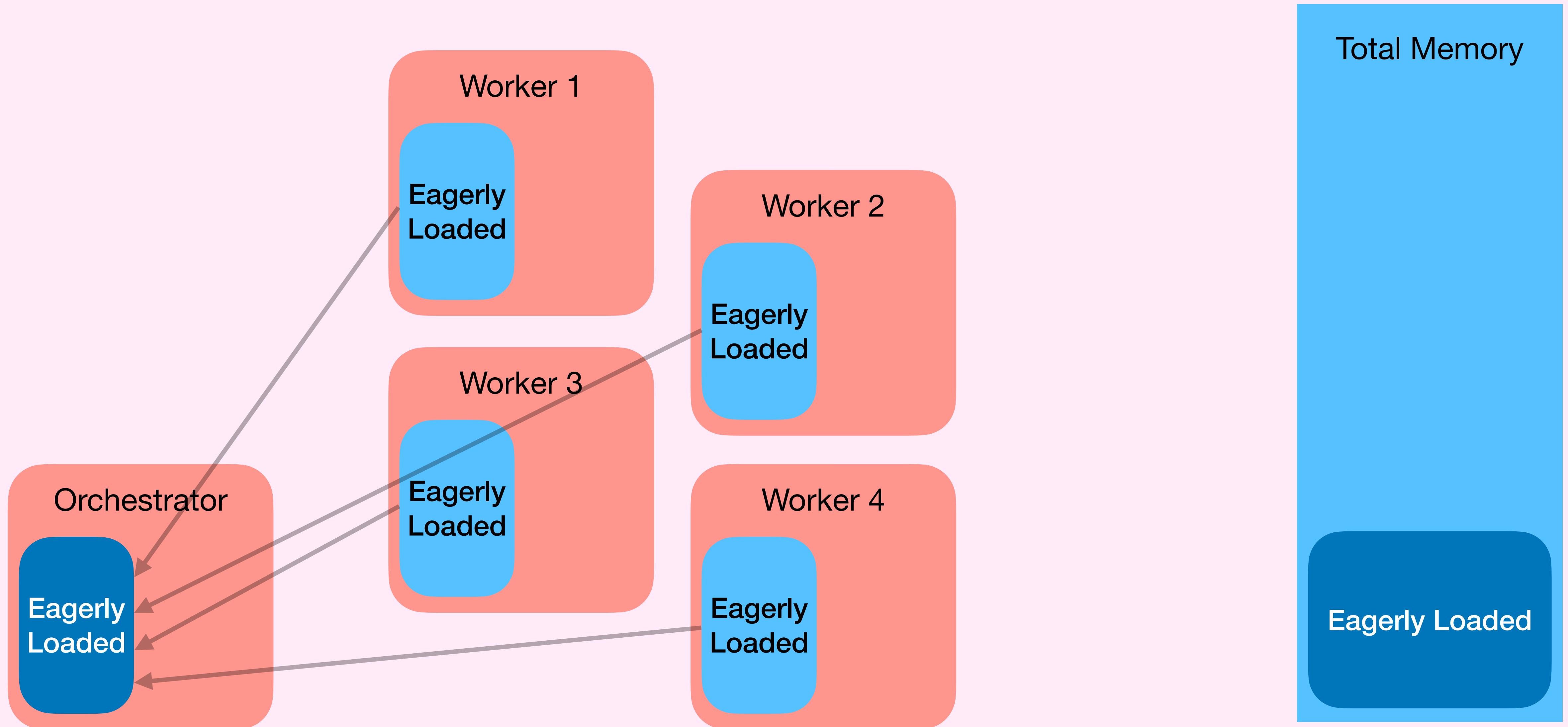
Eager Load, then Fork



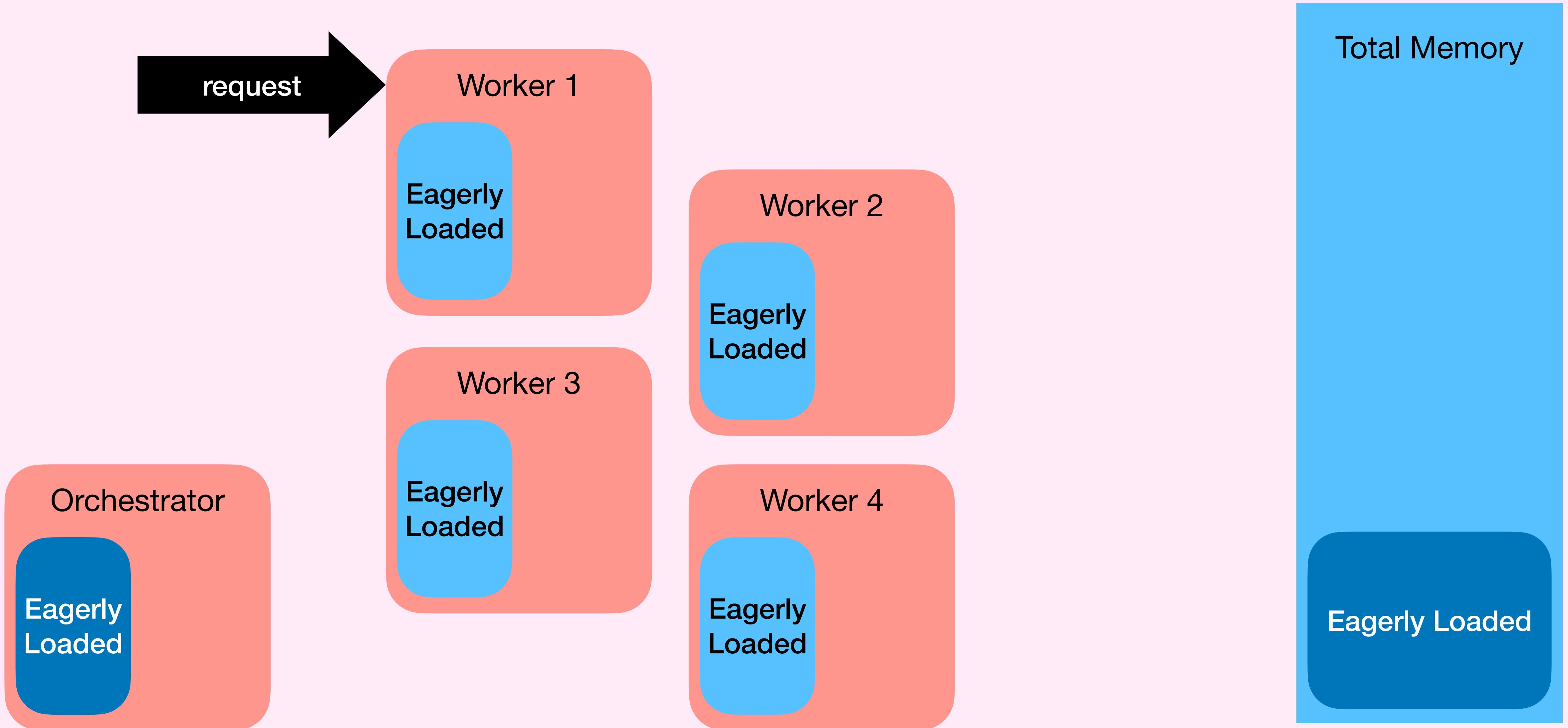
Eager Load, then Fork



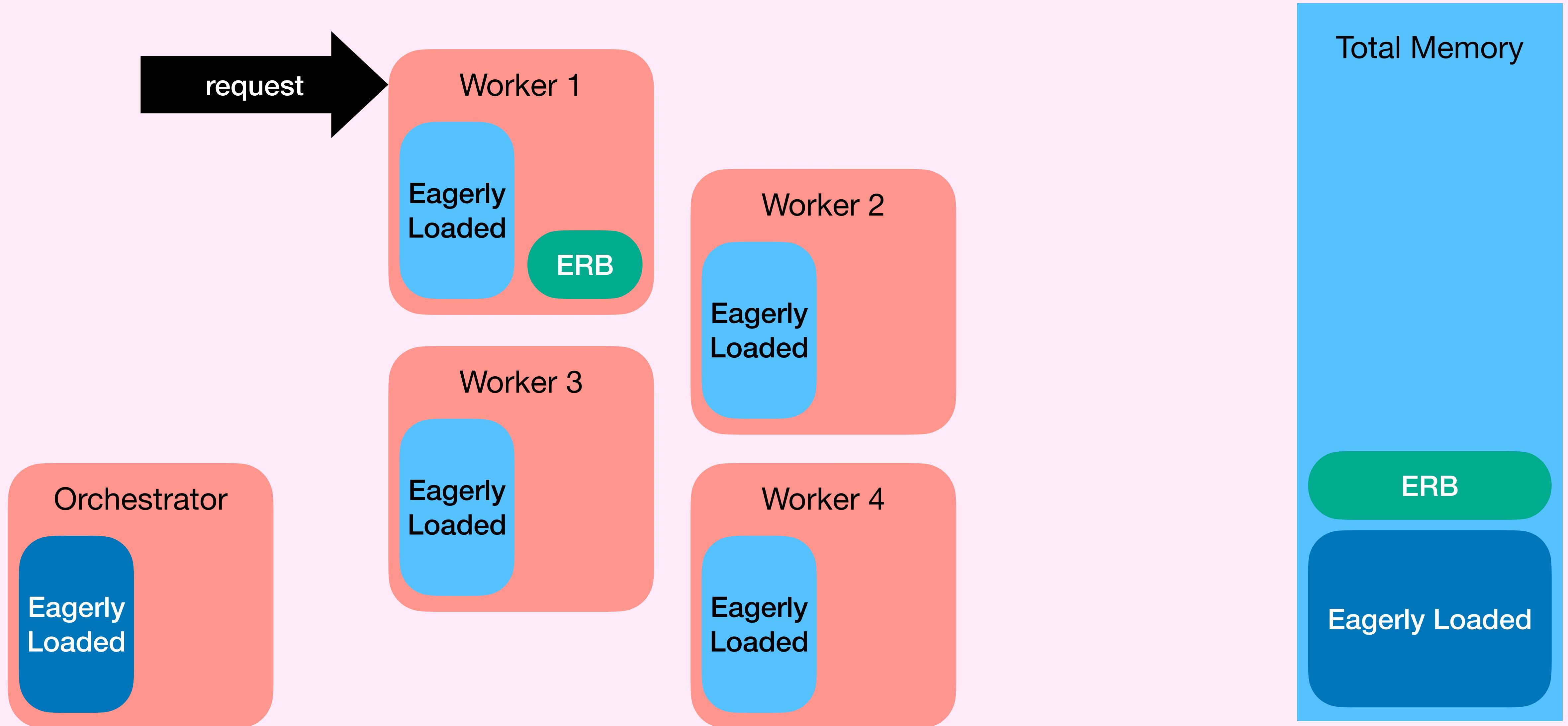
Eager Load, then Fork



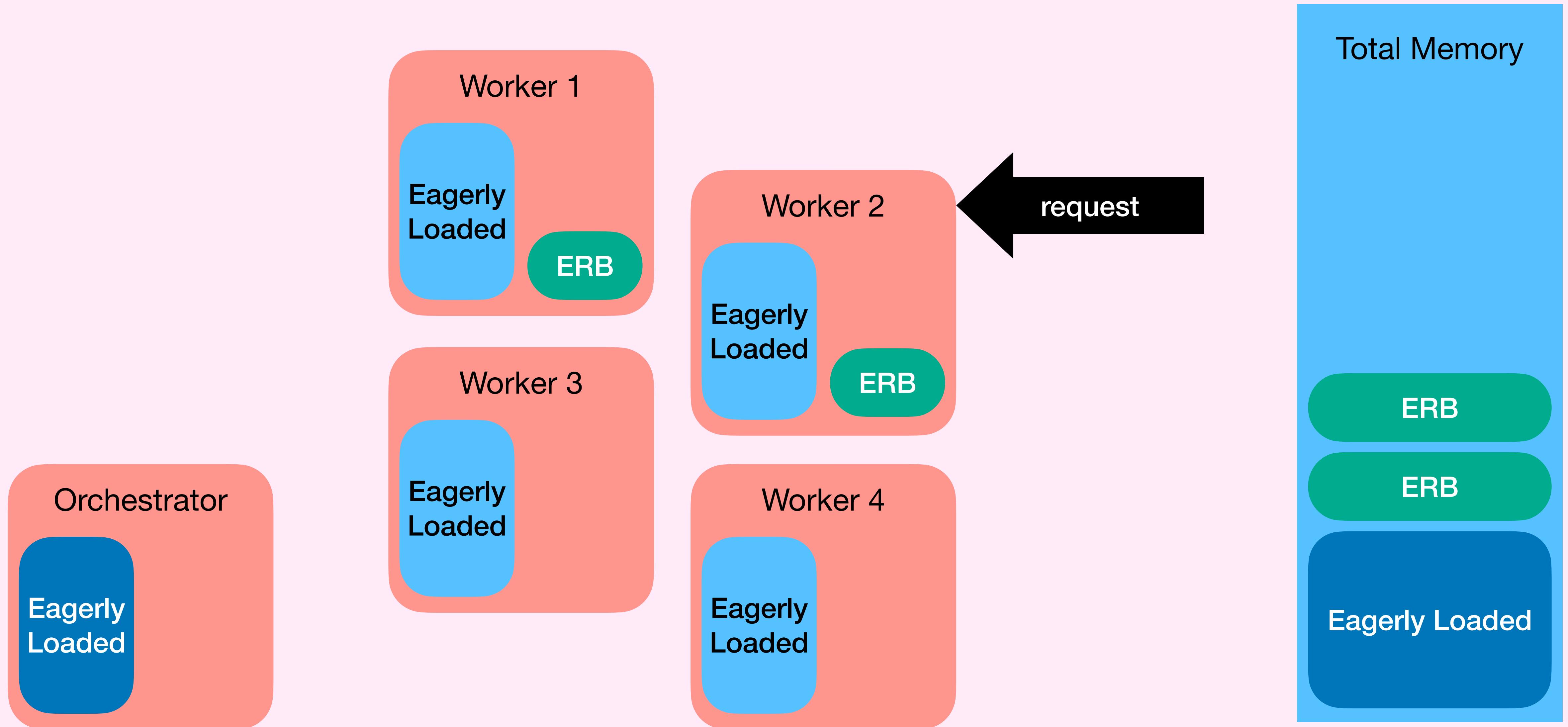
Memory Usage of ERB



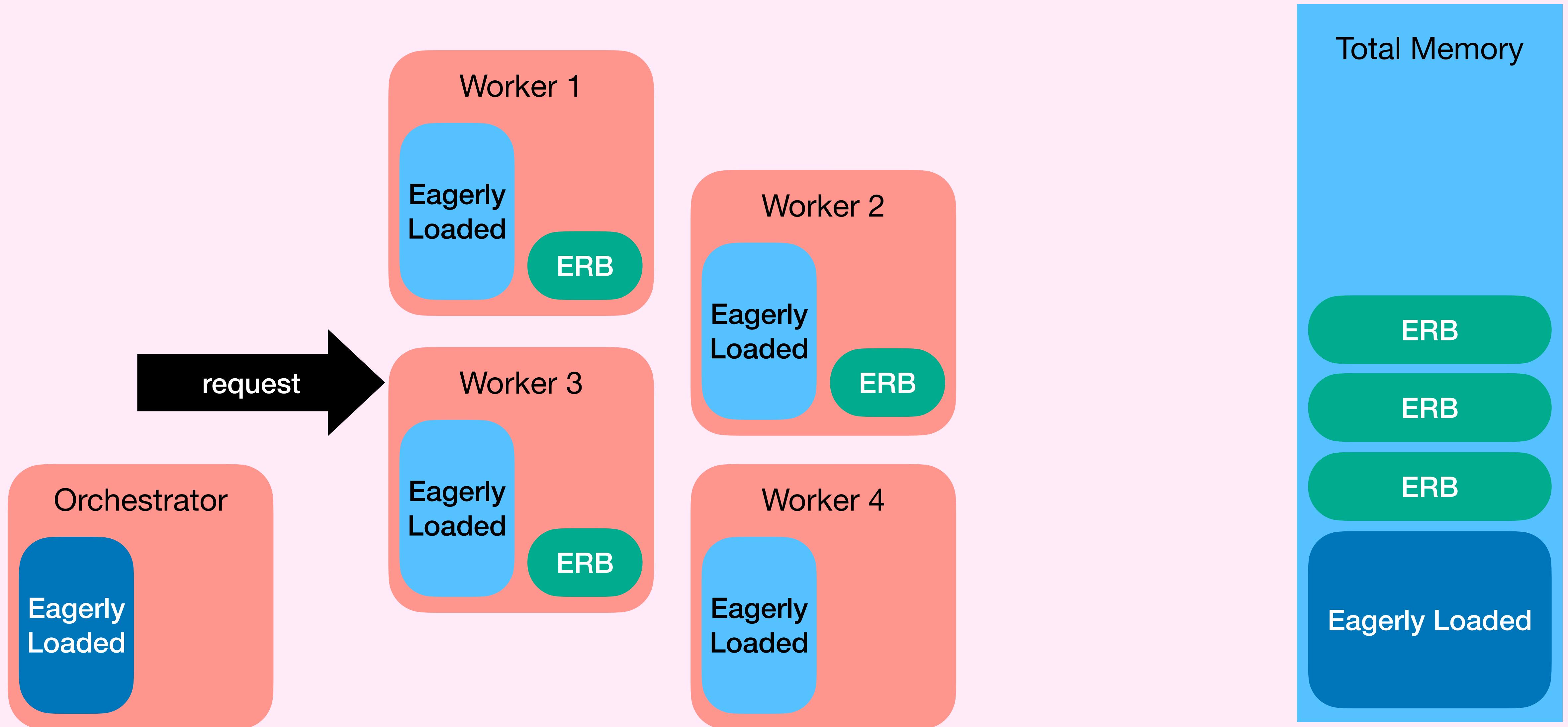
Memory Usage of ERB



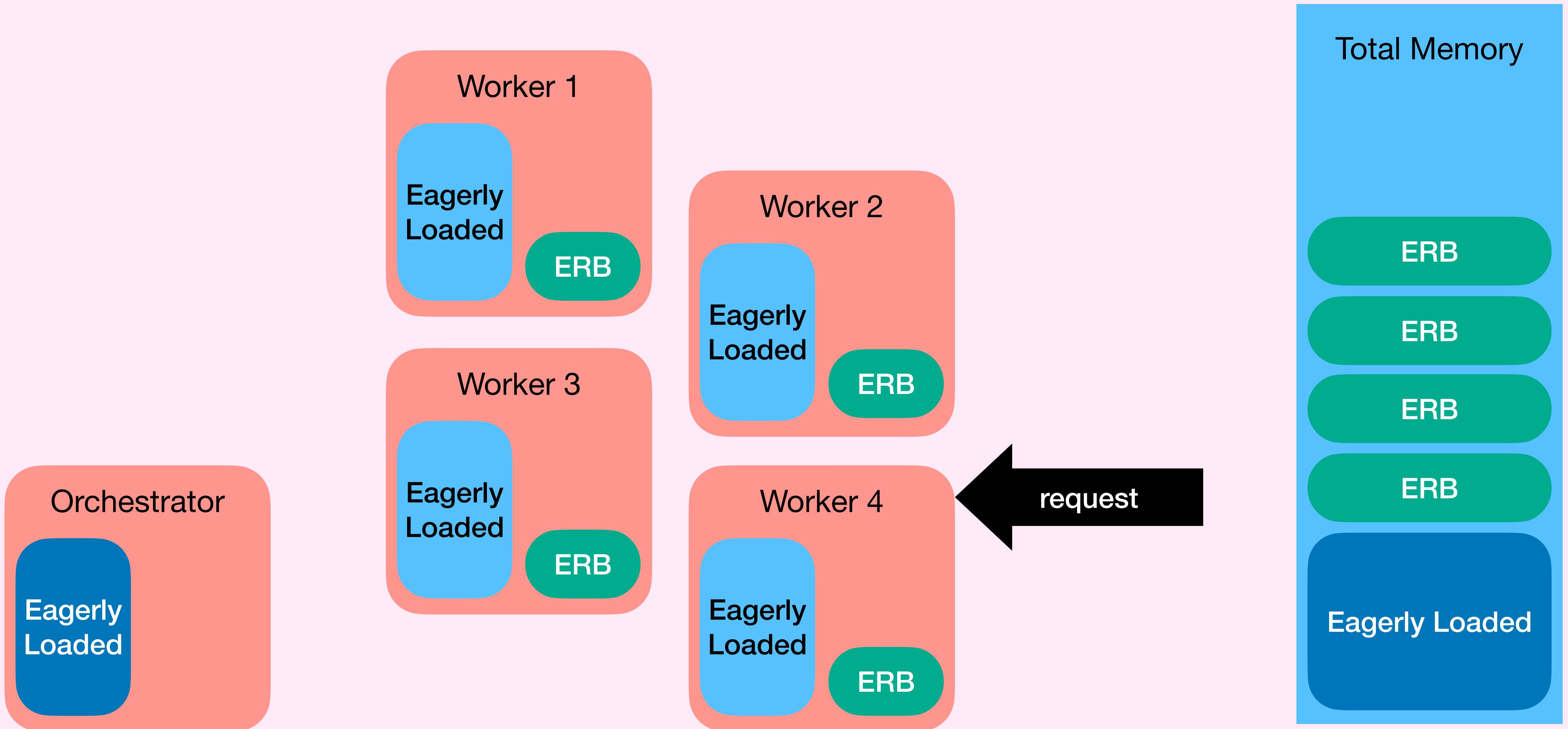
Memory Usage of ERB



Memory Usage of ERB



Memory Usage of ERB



```
def __view_post
  # is link_to a local?
  link_to = local_assigns[:link_to]

  # is post a local?
  post = local_assigns[:post]

<%= link_to post %>          @buf << link_to post
end
```



**jhawthorn/
actionview_precompiler**

Gathering Renderers

```
templates = ActionController::Base.view_paths.each |path|
  # this actually works with non-ERB templates too!
  Dir["**/*", base: path]
end
```

```
helpers = Rails.application.paths["app/helpers"].each |path|
  Dir["**/*_helper.rb", base: path]
end
```

```
controllers = Rails.application.paths["app/controllers"].each do |path|
  # includes Concerns
  Dir["**/*.rb", base: path]
end
```

Extracting Renders

```
(templates + helpers + controllers).each do |path|
  template_name, locals = RipperParser.renders_for(path)
  template_renders << [template_name, locals]
end
```

Compiling Templates

```
lookup_context = ActionController::Base.new.lookup_context

templates = template_renders.flat_map do |name, locals|
  lookup_context.find_all(name, locals)
end

templates.each do |template|
  # warning: private API!
  template.send(:compile!)
end
```

Replace dependency tracker with ripper based tracker #42458

Merged

jhawthorn merged 1 commit into `rails:main` from `HParker:use-ripper-dependency-tracker` on Aug 4, 2021



HParker commented on Jun 11, 2021

Contributor

...

Instead of using the ERBTracker, we can use RipperTracker which is extracted from https://github.com/jhawthorn/actionview_precompiler.

Using a parser finds dependencies that would otherwise be difficult to find with the regular expressions. It should also theoretically work with other template systems since it operates on the compiled template instead of the contents of the file.

Co-authored-by: John Hawthorn



4



**Precompiler extracts template names and
locals using Prism to parse renders.**

[jhawthorn/actionview_precompiler](#)





Marking and Sweeping Dead Code

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb

app/views/layouts/application.html.erb

app/views/posts/_form.html.erb

app/views/posts/_post.html.erb

app/views/posts/edit.html.erb

app/views/posts/index.html.erb

app/views/posts/new.html.erb

app/views/posts/show.html.erb

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb

app/views/layouts/application.html.erb

app/views/posts/_form.html.erb

app/views/posts/_post.html.erb

app/views/posts/edit.html.erb

app/views/posts/index.html.erb 

app/views/posts/new.html.erb

app/views/posts/show.html.erb

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb

app/views/layouts/application.html.erb

app/views/posts/_form.html.erb

app/views/posts/_post.html.erb 

app/views/posts/edit.html.erb

app/views/posts/index.html.erb 

app/views/posts/new.html.erb

app/views/posts/show.html.erb

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb

app/views/layouts/application.html.erb

app/views/posts/_form.html.erb 

app/views/posts/_post.html.erb 

app/views/posts/edit.html.erb 

app/views/posts/index.html.erb 

app/views/posts/new.html.erb

app/views/posts/show.html.erb

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb

app/views/layouts/application.html.erb

app/views/posts/_form.html.erb 

app/views/posts/_post.html.erb 

app/views/posts/edit.html.erb 

app/views/posts/index.html.erb 

app/views/posts/new.html.erb 

app/views/posts/show.html.erb

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb

app/views/layouts/application.html.erb

app/views/posts/_form.html.erb 

app/views/posts/_post.html.erb 

app/views/posts/edit.html.erb 

app/views/posts/index.html.erb 

app/views/posts/new.html.erb 

app/views/posts/show.html.erb 

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb

app/views/layouts/application.html.erb 

app/views/posts/_form.html.erb 

app/views/posts/_post.html.erb 

app/views/posts/edit.html.erb 

app/views/posts/index.html.erb 

app/views/posts/new.html.erb 

app/views/posts/show.html.erb 

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb 

app/views/layouts/application.html.erb 

app/views/posts/_form.html.erb 

app/views/posts/_post.html.erb 

app/views/posts/edit.html.erb 

app/views/posts/index.html.erb 

app/views/posts/new.html.erb 

app/views/posts/show.html.erb 

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb

Marking and Sweeping Dead Code

app/views/components/_badge.html.erb

app/views/layouts/application.html.erb

app/views/posts/_form.html.erb

app/views/posts/_post.html.erb

app/views/posts/edit.html.erb

app/views/posts/index.html.erb

app/views/posts/new.html.erb

app/views/posts/show.html.erb

app/views/posts/what.html.erb

app/views/unknown/unknown.html.erb





**skipkayhil/
loose_erbs**

All Templates

```
$ bundle exec loose_erb --all
```

All ERBs:

```
app/views/components/_badge.html.erb
app/views/layouts/application.html.erb
app/views/posts/_form.html.erb
app/views/posts/_post.html.erb
app/views/posts/edit.html.erb
app/views/posts/index.html.erb
app/views/posts/new.html.erb
app/views/posts/show.html.erb
app/views/posts/what.html.erb
app/views/unknown/unknown.html.erb
```

All Templates

```
# 👍 Fast, no initialization
```

```
# 👎 Doesn't find gems/engines
```

```
all_views = Dir["app/views/**/*.html.erb"]
```

All Templates

```
# 🤐 Slow, runs initialization  
require "config/environment"
```

```
# 👍 Definitive  
all_views = ActionController::Base  
  .view_paths  
  .flat_map do |path|  
    Dir["#{path}/**/*.html.erb"].to_a  
  end
```

Template Dependencies

```
$ bundle exec loose_erb --all --trees
```

```
All Trees:
```

```
  app/views/posts/edit.html.erb
```

```
    └── app/views/posts/_form.html.erb
```

```
  app/views/posts/index.html.erb
```

```
    └── app/views/posts/_post.html.erb
```

```
  app/views/posts/new.html.erb
```

```
    └── app/views/posts/_form.html.erb
```

```
  app/views/posts/show.html.erb
```

```
    └── app/views/posts/_post.html.erb
```

```
  app/views/posts/what.html.erb
```

```
  app/views/unknown/unknown.html.erb
```

```
    └── UNKNOWN TEMPLATE: does/not/exist
```



Marking Roots: Templates are Alive

```
roots = nodes.filter { |n| Partial !== n }
```

 Simple

 Generally effective for finding loose partials

 Low chance of false-positive

 High chance of false-negative

Marking Roots: Render Parsing

```
roots = controller_renders + helper_renders
```

👍 More accurate for Templates

👍 Low chance of false-negative

👎 High chance of false-positive

👎 Turbo makes this harder

👎 actionview_precompiler owns this logic

All Templates

```
$ bundle exec loose_erb
```

Loose ERBs:

```
app/views/posts/what.html.erb
```

```
app/views/unknown/unknown.html.erb
```

THE GHOST OF
ACTION VIEH
FUTURE

Keep improving the foundations.

When rendering a partial, any mention of the string "render" inside a `cache` block results in the logged error message "Couldn't find template for digesting: strings/string" #55161

🕒 Open



cannikin opened on Jun 5

Edits ▾ ⋮

Steps to reproduce

Create a file `_partial.html.erb` with the following contents alongside the minimal reproduction shown below:

```
<%# _partial.html.erb %>

<% cache "key" do %>
  <div class="render flex" data-action="controller#render other-controller#click">
    Hello, world!
  </div>
<% end %>
```



Tighter integration.

```
$ bin/rake --tasks cache_digests
```

```
# Lookup first-level dependencies for TEMPLATE  
# (like messages/show or comments/_comment.html)  
rake cache_digests:dependencies
```

```
# Lookup nested dependencies for TEMPLATE  
# (like messages/show or comments/_comment.html)  
rake cache_digests:nested_dependencies
```

commit 502d5e24e28b3634910495d0fb71cb20b1426aee
Author: David Heinemeier Hansson <david@loudthinking.com>
Date: Wed Aug 29 14:23:15 2012 -0500

Add automatic template digests to all CacheHelper#cache calls (originally spiked in the cache_digests plugin) *DHH*

Use Action View!