

Merb

and why you potentially should care

By Luke Franci (look@recursion.org)

Presented at the Ruby Users of Minnesota, March 27, 2007.

Merb = Mongrel + ERB

By Ezra Zygmuntowicz

Merb = Mongrel + ERB, written by Ezra Zygmuntowicz happen.

When I heard Merb was Mongrel+ERB, I thought “Ah, PHP is making a comeback.” But Merb is actually a small MVC framework similar to Rails in many ways.

Key points about Merb

Merb is thread-safe

Merb does not use ActionPack (Rails controller/view), which is the main offender.

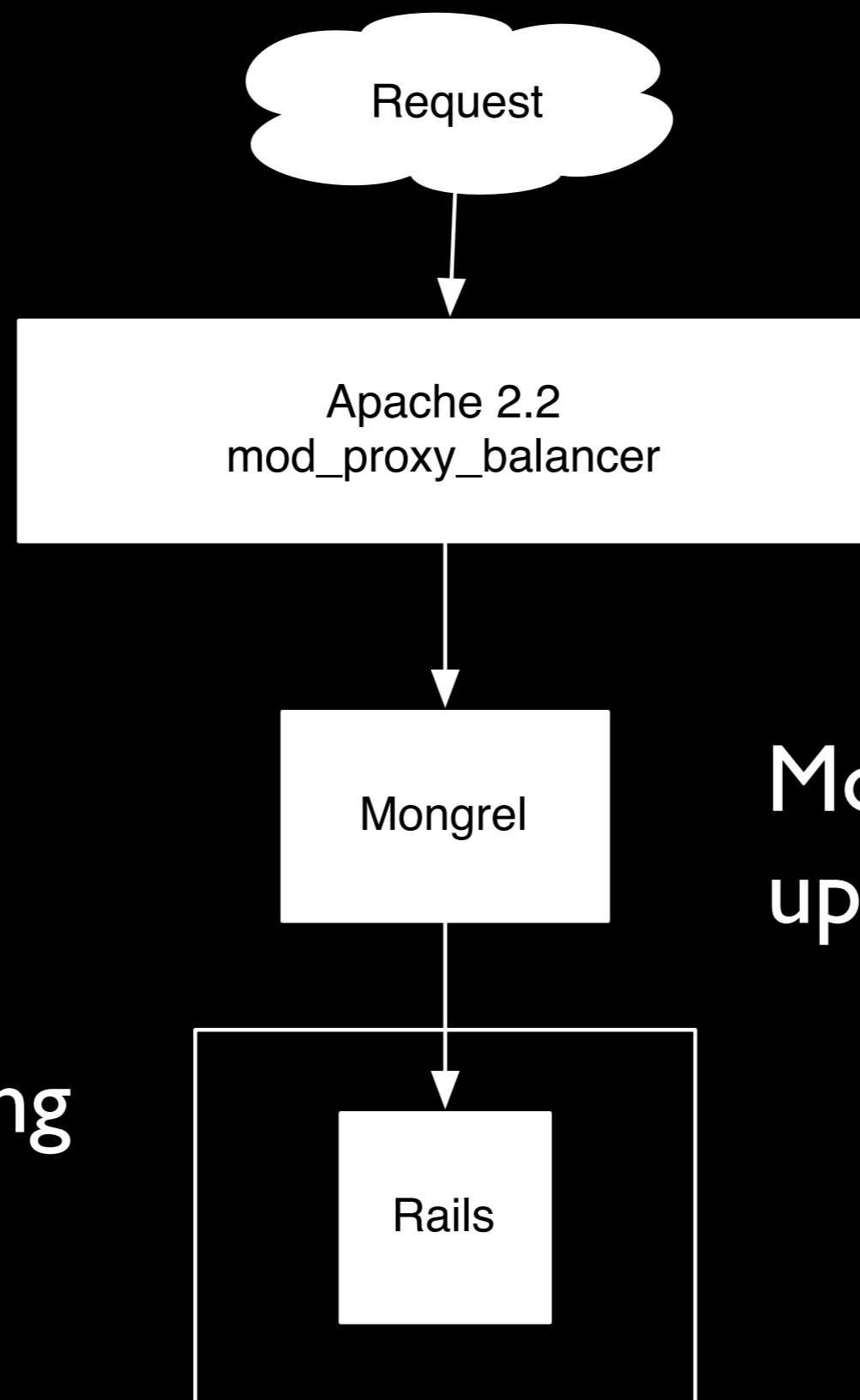
Note: Ruby uses "green threads", so I believe a that to take advantage of a multi-core computer, you will need to start one Ruby instace per processor core.

Merb uses Erubis for embedded Ruby

Merb uses Erubis for ERB

- 3x faster than ERB
- 10% faster than eRuby (written in C)

Merb does not use CGI.rb



Rails blocks
while processing
the request.
CGI.rb causes
CPU spike.

Mongrel buffers the
upload.

It is **NOT TRUE** that Mongrel blocks while you are uploading a file.

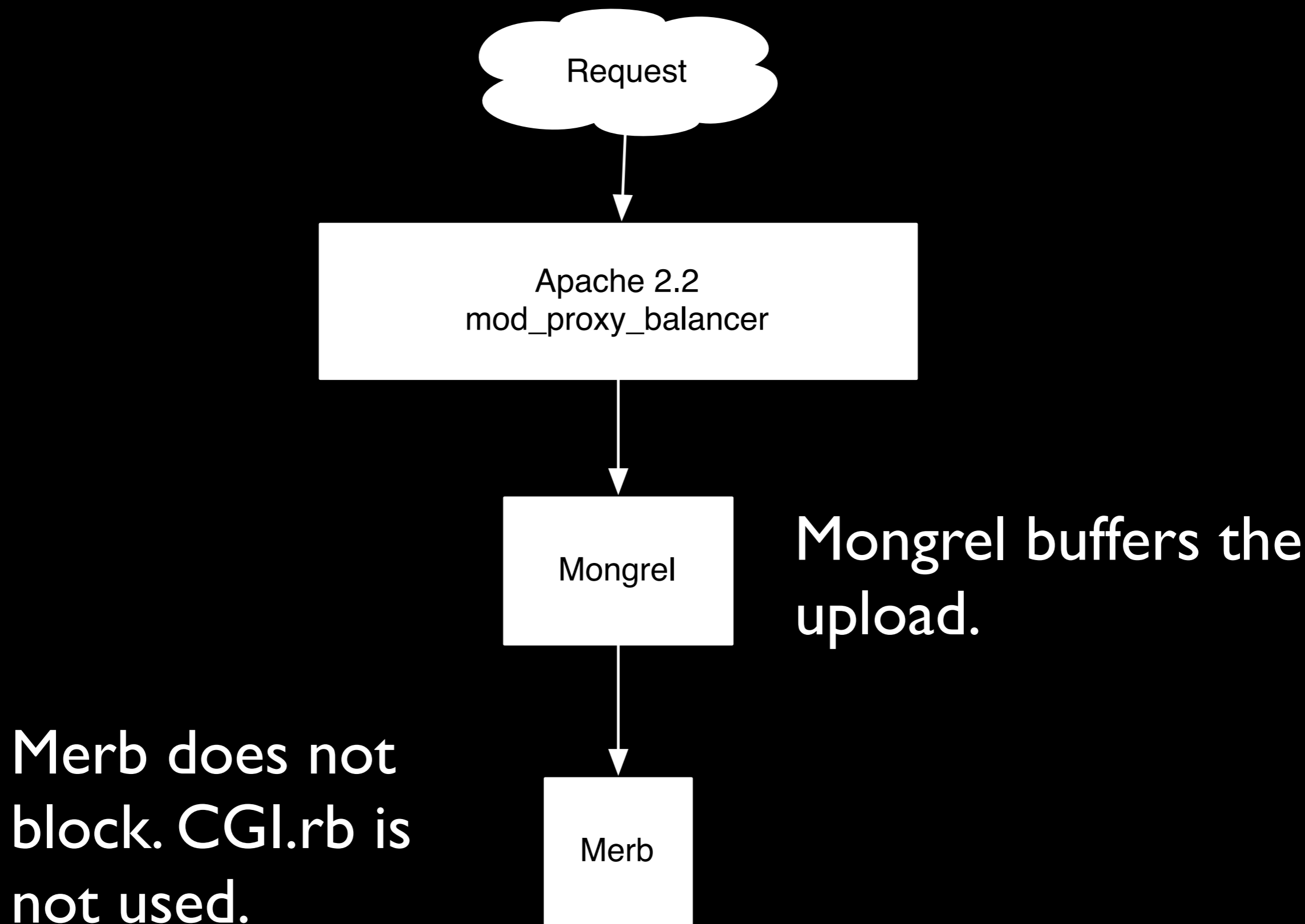
Rails blocks while you are **processing** the file.

[File] upload doesn't block Rails actions going on, when you finally pass this to Rails you'll block that Mongrel process while `cgi.rb` is going.

This is why you should make a separate Mongrel handler to do all of your upload processing and file preparation before you pass the fully cooked stuff to Rails. **Mongrel running `cgi.rb` in a thread is much more efficient than Rails running `cgi.rb` inside a lock.**

- Zed Shaw

So, if your app has to deal with the uploads of a lot of large files which you need to do processing on, a custom upload handler will save your butt. upload



This is why you should care about Merb -- if you need the capacity.
Think of it as a custom Mongrel file handler...that you don't have to write.

Installing Merb

```
sudo gem install mongrel json erubis  
archive-tar-minitar rspec -y
```

```
svn co http://svn.devjavu.com/merb/trunk  
merb && cd merb && rake install
```

Then you can generate a new merb app:

```
merb -g myapp
```

Merb is available as a gem, but it is rather rough right now. I recommend getting the trunk to keep abreast of the latest bug fixes.

There is a simple blogging application called MrBlog that will show you how Merb works.

Merb vs Rails

Merb uses ActiveRecord, so all your favorite features are there. It's M/V/C, so development is similar.

- generator scripts are not generally available
- routing is different (simple but functional)
- not many view helpers
- most configuration in Ruby rather than YAML
- In general, rougher around the edges. (Example: multipart form uploads were broken)

Merb + Rails

Perhaps more interesting is getting Merb working with your existing Rails app. In about half a days work, I was able to get Merb loading up my ActiveRecord objects from my rails project (to avoid code duplication), and get a Rails AR plugin working. Merb can piggy-back on your Rails sessions as long as you store them in the database.

```
class Item < ActiveRecord::Base
end
require DIST_ROOT + '/../../app/models/item.rb'
```

Many ActiveRecord plugins will work with Merb

...with a little modification.

I got Rick Olson's attachment_fu plugin working with Merb with a few tweaks to the code. Mostly I had to change how files were required, because Rails does a lot of magic to load plugins.

In this way, you can use Rails for your views and general app, and pass off file upload POST actions to Merb for

LET'S LOOK AT A SHORT DEMO.

Resources

Merb Docs: <http://merb.rubyforge.org>

Merb Trac: <http://merb.devjavu.com>

Ezra's blog: <http://brainspl.at/>

Erubis: <http://www.kuwata-lab.com/erubis>