Thrift

serviços para comunicação inter-linguagens

Remote Procedure Call

Comunicação inter-processos

Sintaxe “familiar”

“Distributed objects era of the 90s”

(91) Corba

(93) Microsoft COM

(97) Java RMI

(98) XML-RPC (depois SOAP)

(99) EJB

OVERHEAD

REST

Roy Fielding, 2000

“estado” abstraído para “recurso”

sintaxe universal para links

operações e content-types bem definidos

stateless, layered, cacheable

Thrift

Facebook, 2007

RPC

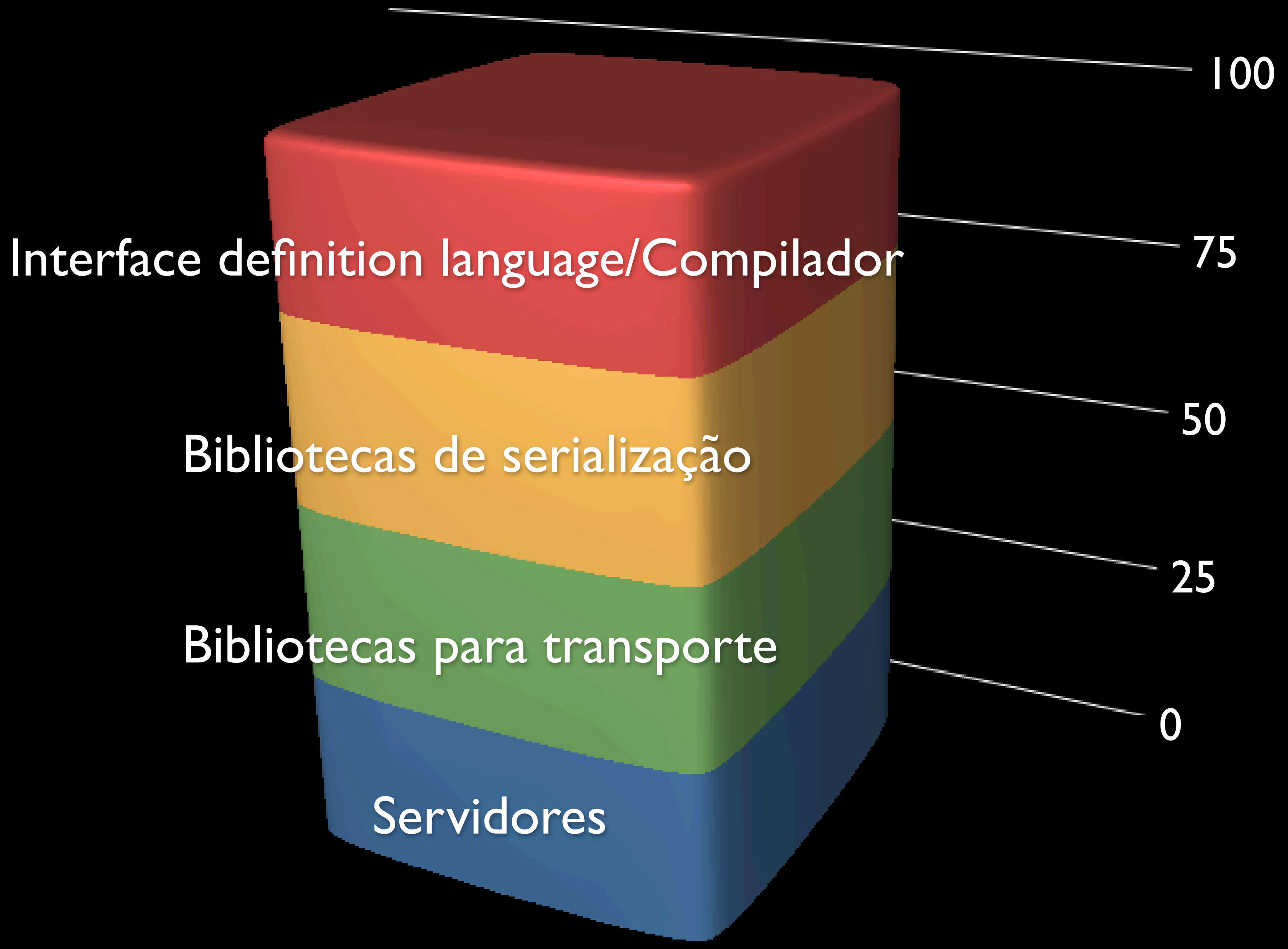
(de novo)

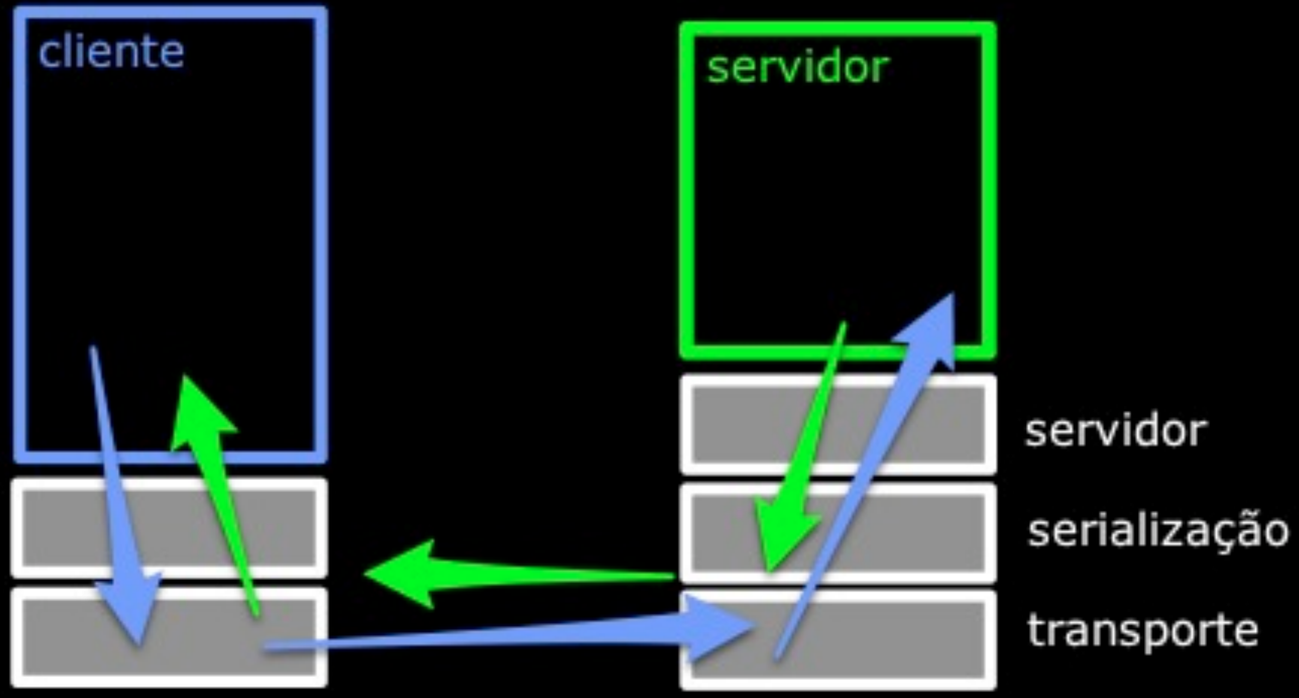
Adotado pelo Apache Incubator (2008)

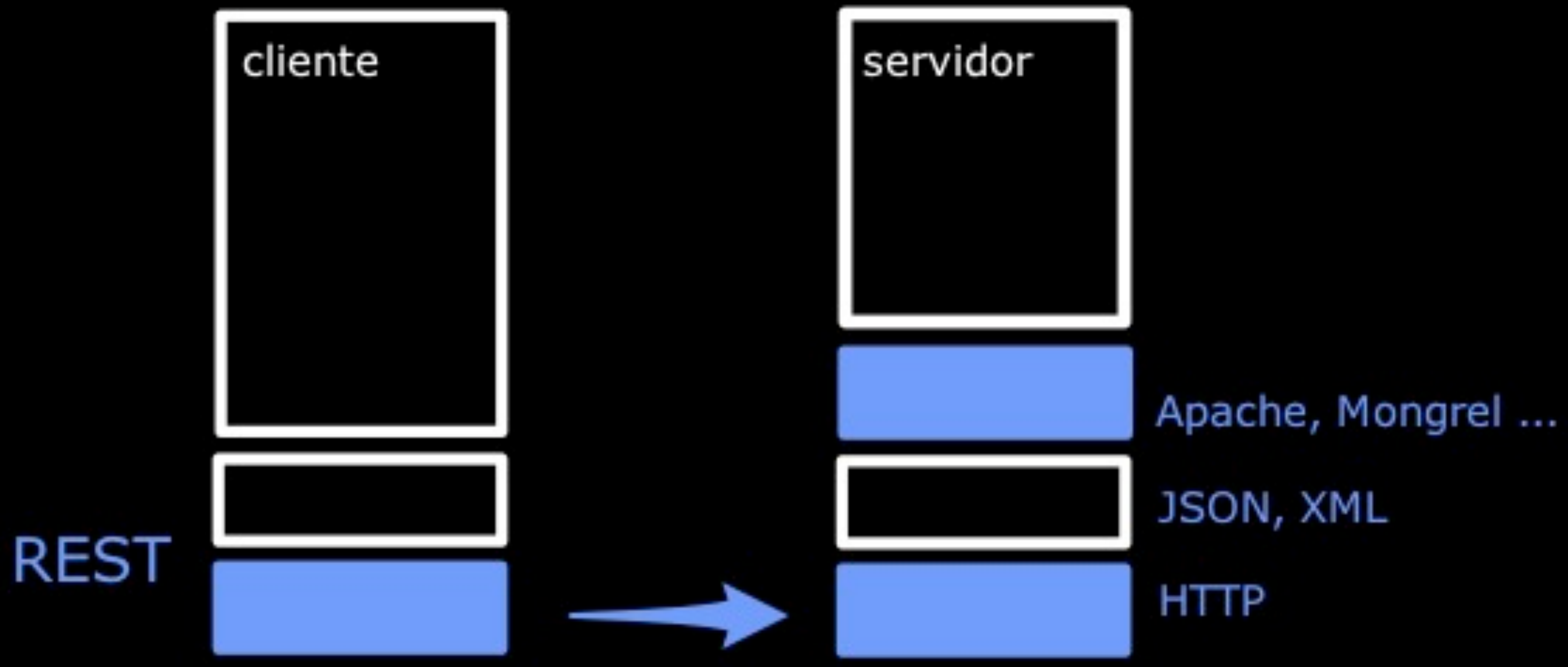
C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, Smalltalk, OCaml, ...

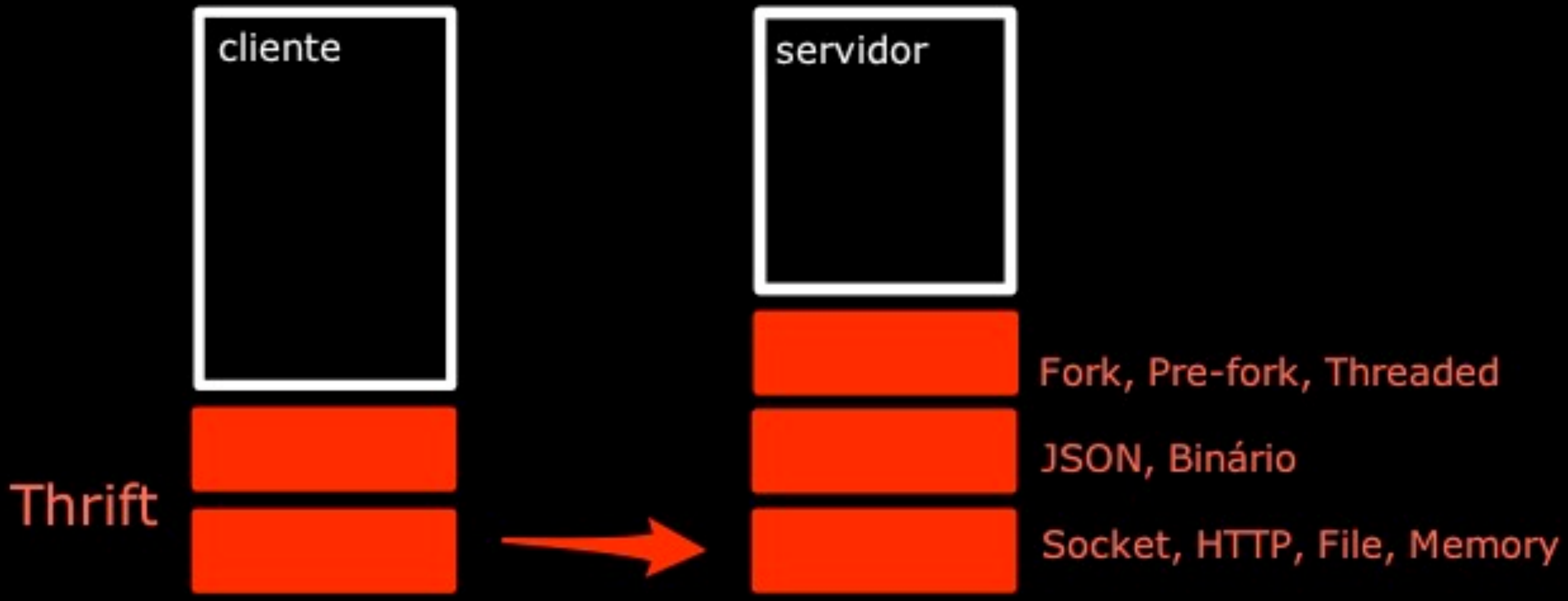
“Rápido, realmente rápido”

hein?


100
75
50
25
0
Interface definition language/Compilador
Bibliotecas de serialização
Bibliotecas para transporte
Servidores


cliente
servidor
servidor
serialização
transporte


cliente
servidor
Apache, Mongrel ...
JSON, XML
REST
HTTP


cliente
servidor
Thrift
Fork, Pre-fork, Threaded
JSON, Binário
Socket, HTTP, File, Memory

Interface Definition Language

Compilador

Tipos

Traduzidos em tipos “nativos”

Sem tipos especiais ou wrappers

Básicos

- bool
- byte
- i16, i32, i64
- double
- string

Containers

list<type>

set<type>

map<type1, type2>

Structs

```
struct Example {
  1:i32        number=10,
  2:i64        big_number,
  3:double   decimal,
  4:string    name="thrifty"
}
```

```ruby
class Example
  include ::Thrift::Struct

  ::Thrift::Struct.field_accessor self, :number, :big_number, :decimal, :name

  ...
end
```

```python
class Example(object):

  def __init__(self, number=10, big_number=None, decimal=None, name="thrifty",):
    self.number = number
    self.big_number = big_number
    self.decimal = decimal
    self.name = name

  ...
```

```php
<?php
class Example {
  public $number = 10;
  public $big_number = null;
  public $decimal = null;
  public $name = "thrifty";

  public function __construct($vals=null) {
    if (is_array($vals)) {
      if (isset($vals['number'])) {
        $this->number = $vals['number'];
      }
      if (isset($vals['big_number'])) {
        $this->big_number = $vals['big_number'];
      }
      if (isset($vals['decimal'])) {
        $this->decimal = $vals['decimal'];
      }
      if (isset($vals['name'])) {
        $this->name = $vals['name'];
      }
    }
  }
}
?>
```

```java
public class Example implements TBase, java.io.Serializable, Cloneable {
  public int number;
  public long big_number;
  public double decimal;
  public String name;

  public Example() {
    this.number = 10;
    this.name = "thrifty";
  }

  public Example(int number, long big_number, double decimal, String name)
  {
    this();
    this.number = number;
    this.big_number = big_number;
    this.decimal = decimal;
    this.name = name;
  }

  ...
}
```

Exceções

```
exception ExampleException {
  1:i32       number=10,
  2:i64       big_number,
  3:double   decimal,
  4:string    name="thrifty"
}
```

Python	`class ExampleException(Exception):`
Java	`public class ExampleException extends Exception {`
PHP	`class ExampleException extends TException {`
Ruby	`class ExampleException < ::Thrift::Exception`

Serviços

```
service RemoteHashMap {
    void       set(1:i32 key, 2:string value),
    string     get(1:i32 key) throws (1: KeyNotFound knf),
    async void delete(1:i32 key)
}
```

Protocolo

Métodos para leitura e escrita

Encoding dos tipos básicos, structs e containers

TProtocol

writeMessageBegin()

writeI16()

readI16()

readI32()

...

TBinaryProtocol

TCompactProtocol

TJSONProtocol

...

Transporte

Transferência de dados

Duas interfaces

TTransport

open

close

isOpen

read

write

flush

TServerTransport

open

listen

accept

close

TSocket

TFileTransport

TMemoryBuffer

THttpClient

...

Servidores

TThreadedServer

TThreadPoolServer

TForkingServer

...

Uso

1. Definir as estruturas de dados e serviços

```
enum MartialArt {
  AIKIDO    = 1,
  KARATE    = 2
}

struct UserProfile {
  1: i32            uid,
  2: string         name,
  3: MartialArt  style
}

service UserStorage {
  void              store(1: UserProfile user),
  UserProfile  retrieve(1: i32 uid)
}
```

1. Definir as estruturas de dados e serviços

Enum!

```
enum MartialArt {
  AIKIDO   = 1,
  KARATE   = 2
}

struct UserProfile {
  1: i32          uid,
  2: string       name,
  3: MartialArt   style
}

service UserStorage {
  void          store(1: UserProfile user),
  UserProfile   retrieve(1: i32 uid)
}
```

2. Gerar código “stub”

```
$ thrift --gen php py:new_style service.thrift
```

3. Implementar lógica do serviço

```
class UserStorageHandler:
    def store(self, user):
        self.do_store(user)

    def retrieve(self, id):
        return User.find_by_id(id)

    ...
```

4. Implementar o servidor

```python
handler = UserStorageHandler()
processor = example.UserStorage.Processor(handler)
transport = TSocket.TServerSocket(9090)
tfactory = TTransport.TBufferedTransportFactory()
pfactory = TBinaryProtocol.TBinaryProtocolFactory()

server = TServer.TThreadedServer(processor, transport, tfactory, pfactory)

print 'Starting the server...'
server.serve()
```

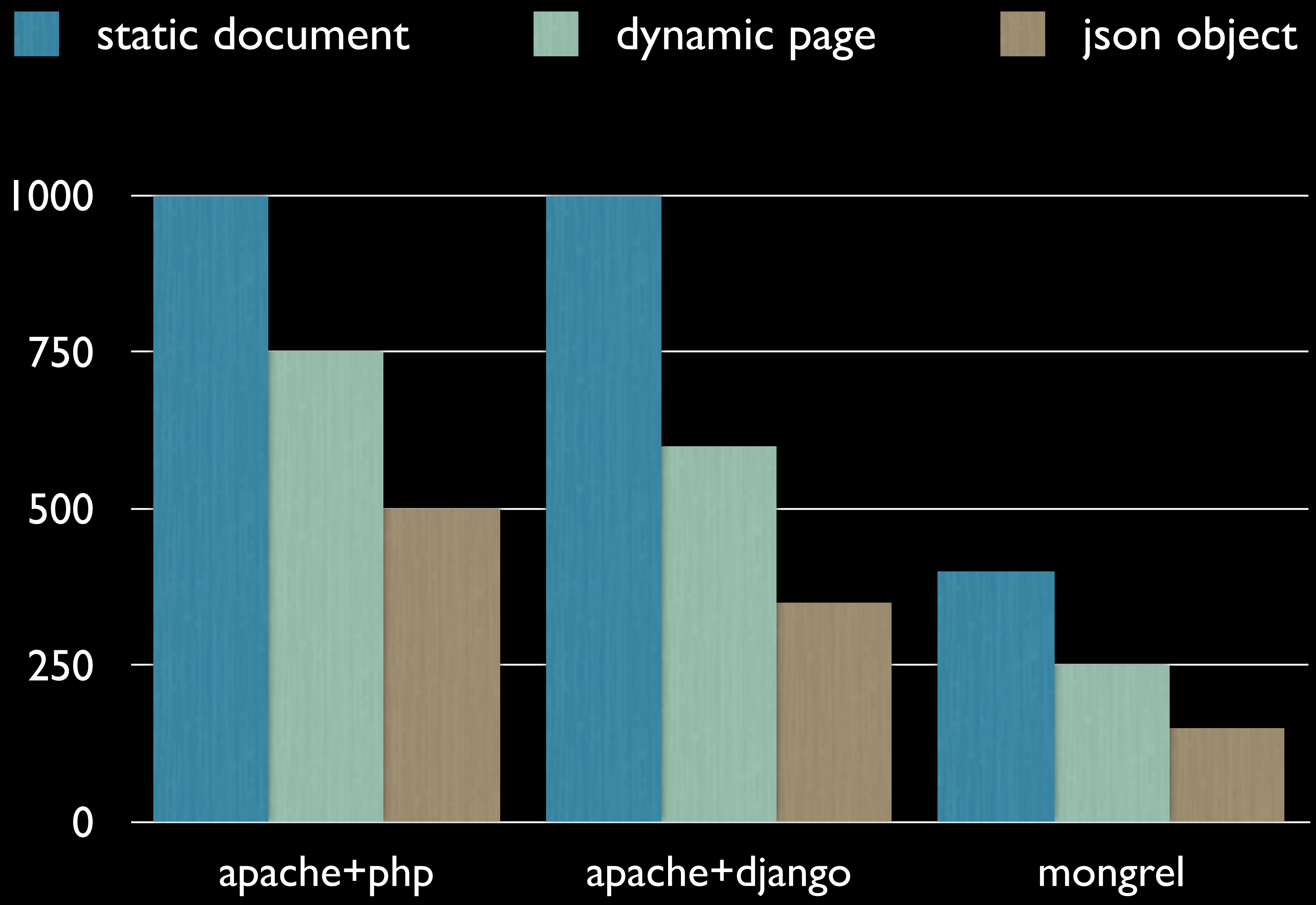
5. Implementar o cliente

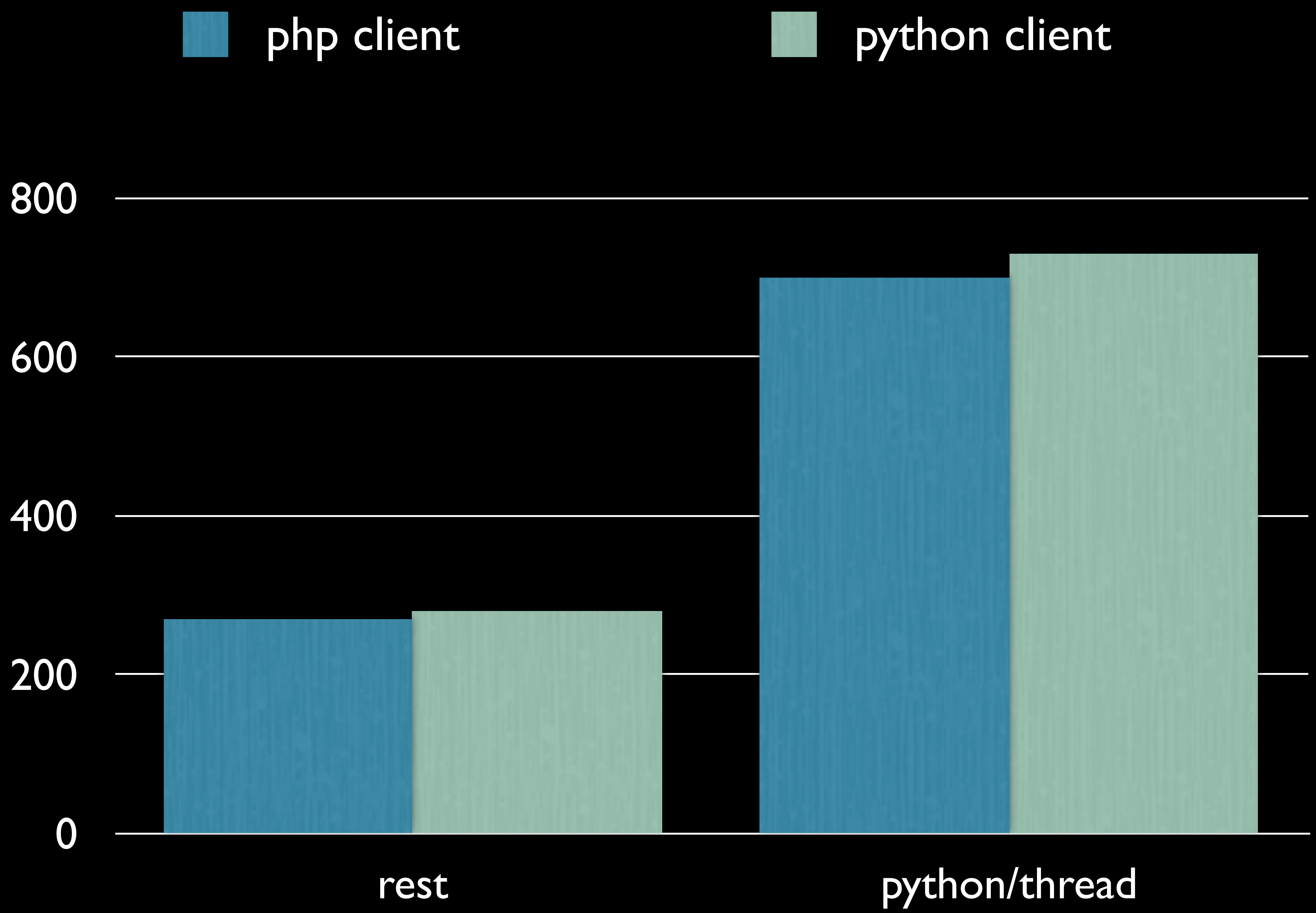
```php
<?php
  $socket = new TSocket('localhost', 9090);
  $transport = new TBufferedTransport($socket, 1024, 1024);
  $protocol = new TBinaryProtocol($transport);
  $client = new UserStorageClient($protocol);

  $transport->open();
  $new_user = new example_UserProfile(array(
    "uid" => '123',
    "name" => "Ralph Waldo Emerson",
    "style" => example_MartialArt::KARATE
  ));
  $client->store($new_user);

  $transport->close();
?>
```

6. Deploy!

```
$ python server.py
```

Benchmarks


static document
dynamic page
json object
1000
750
500
250
0
apache+php
apache+django
mongrel


php client
python client
800
600
400
200
0
rest
python/thread

Desvantagens

Não tão ubíquo (quanto HTTP)

Não tão maduro (quanto HTTP)

Pontos fortes

Compatibilidade entre linguagens

Serialização built-in

Performance!

Mais

Versionamento da interface

Thrift + protocol buffers

Referências

http://incubator.apache.org/thrift/

https://github.com/bzanchet/presentation-thrift-fisl10/