

Ruby?

ネットワーク応用通信研究所

かずひこ



XML-RPCって何？

HTTP 上で XML を用いてリモート手続き呼び出しを行うもの

- 日記とかblogとかからping送るやつ
- 弱いSOAP

● XML-RPCの型とRubyの型

XML-RPC 型 ☒

int/i4 ☒

boolean ☒

string ☒

double ☒

dateTime.iso8601 ☒

base64 ☒

array ☒

struct ☒

送信可能 Ruby 型

Integer

TrueClass/FalseClass

String/Symbol

Float

Time/Date/XMLRPC::DateTime

XMLRPC::Base64

Array

Hash/Struct

● XML-RPCの型とRubyの型 (2)

実はどんなオブジェクトでも送れなくはない

```
class Hoge
```

```
   include XMLRPC::Marshallable
```

```
   attr_reader :a, :b
```

```
   ....
```



```
{ "a"=>"hoge", "b"=>"fuga", "__class__"=>"Hoge"}
```

● XML-RPCクライアントの例

```
#!/usr/bin/env ruby
require "xmlrpc/client"
server = XMLRPC::Client.new2( "http://host/path/xmlrpc.cgi")
begin
  ☒ p server.call("sample.sumAndDifference", 5, 3)
rescue XMLRPC::FaultException => err
  ☒ p err.faultString
end

# => {"sum"=>8, "difference"=>2}
```

XML-RPCの限界

- HTTPなので「つなぎっぱなし」にはできない
- 今回のようなリアルタイムものでは、定期的に
 ポーリングして状態を取得する必要がある

● 対戦n-Queensのアルゴリズム (理想)

- ・ あらかじめ想定される盤面サイズに対して
- あらゆる対戦パターンを調べて、任意の
- 局面におけるベストな手を記録しておく



- 計算量が多すぎて時間切れ...

● 対戦n-Queensのアルゴリズム (現実)

- ・☒とりあえず一つでも手がみつかったら候補
- ・☒あとは時間が許すかぎり計算
- ・☒「負けない」解が見つかったら計算終了

● 対戦n-Queensのアルゴリズム (現実)

```
require 'timeout'
```

```
begin
```

```
  ☒ timeout( 持ち時間 ) do
```

```
    ☒ ☒ とりあえず置ける手の探索
```

```
    ☒ ☒ よりよい解が見つければ追加していく
```

```
    ☒ ☒ 負けな解が見つかったらそれを返す
```

```
  ☒ end
```

```
rescue TimeoutError
```

```
  ☒ 最後の解を返す
```

```
end
```