

FastRPC

meziserverová komunikace

Vlastimil Pečínka
<vlastimil.pecinka@firma.seznam.cz>

<http://fastrpc.sourceforge.net/>

- **Aplikační servery** co a proč
- **Middleware** vysvětlení, RPC
- **XML-RPC** middleware protokol založený na XML
- **FastRPC** zpětně kompatibilní binární verze
- **Metaserver** framework pro tvorbu backendů
- **Příklad** server pro matematické operace
- **Diskuse**

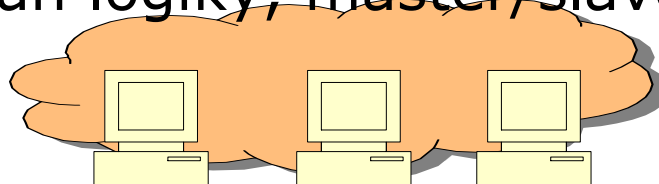
Aplikační servery

- N-vrstvá architektura aplikací, distribuované prostředí
- Logika v aplikačním serveru (backend)
- „Tenký“ fronted (např. webová aplikace v PHP/Python)
- Komunikace vrstev - middleware

- V Seznamu aplikační servery původně ze dvou důvodů:

- vyřešit timeouty (např. do databáze)
- vyřešit failover (např. restart databáze na pozadí)

- Následně přesun logiky, master/slave databáze, ...

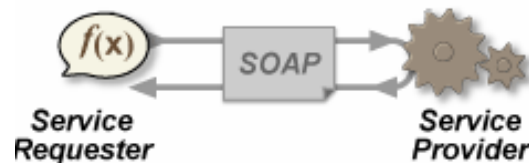


Middleware

- Software spojující softwarové komponenty nebo aplikace
- Umožňuje aplikacím sdílet data (a logiku)
- Spojován s pojmy jako XML-RPC, SOAP, Web services
- Různé typy middleware
 - RPC (remote procedure call)
 - MOM (message oriented middleware)
 - ORB (object request broker)
 - SQL oriented data access

Middleware - RPC

- Vzdálené volání procedur klientem na serveru (vs. „lokální“ volání v programu)
- Synchronní a asynchronní
- Klient - server model distribuovaného výpočtu
- Klient sestaví dotaz, odešle **transportní vrstvou** na server, tam dojde k provedení akce, sestavení odpovědi a odeslání na server
- První známá implementace: SUN RPC (NFS)



XML-RPC

- Sestavení dotazu a odpovědi ve formě XML dokumentu
- Transportní vrstva – HTTP
- Velmi jednoduchý protokol (definice na dvě stránky)
- Upovídaný protokol - z povahy XML (tagy, escaping)
- Náročný u vysokozátěžových aplikací, nevhodný pro výměnu velkého objemu dat (parsing)
- Pouze 8 datových typů (+ <nil/> jako XML-RPC extenze)

XML-RPC – dotaz

```
<?xml version="1.0"?>
<methodCall>
  <methodName>suma</methodName>
  <params>
    <param>
      <value><i4>12</i4></value>
    </param>
    <param>
      <value><i4>24</i4></value>
    </param>
  </params>
</methodCall>
```

XML-RPC – odpověď

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <memeber>
            <name>status</name>
            <value><i4>404</i4></value>
          </member>
          <member>
            <name>statusMessage</name>
            <value><string>Not found</string></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```


XML-RPC – datové typy

Název	Příklad
array	<pre><array> <data> <value><i4>1212</i4></value> <value><string>1212</string></value> </data> </array></pre>
base64	<pre><base64>eW91IGNhbid0IHgdGhpcyE=</base64></pre>
boolean	<pre><boolean>1</boolean></pre>
datetime	<pre><dateTime.iso8601>19980717T14:08:55</dateTime.iso8601></pre>
double	<pre><double>-12.53</double></pre>
integer	<pre><i4>42</i4> nebo <int>42</int></pre>
string	<pre><string>Hello world!</string></pre>
struct	<pre><struct> <member> <name>foo</name> <value><i4>12</i4></value> </member> </struct></pre>
nil	<pre><nil/></pre>

FastRPC

- Zpětně kompatibilní se XML-RPC, vlastní implementace xml parseru, autodetekce klienta
- Transportní vrstva – HTTP, možnost chunkování
- Stejných 8 datových typů a 3 non-datové (method call, method response, fault response)
- Proprietární binární forma protokolu – viz. SourceForge



- Značná úspora procesorového času a objemu přenesených dat

FastRPC – režim práce

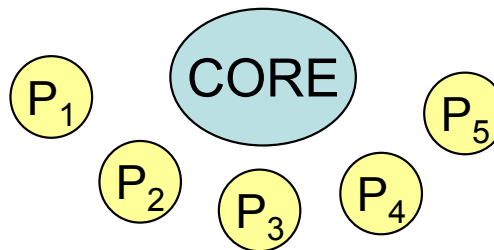
- XML-RPC požadavek, XML-RPC odpověď
 - zpětná kompatibilita s XML-RPC klienty
 - klient neposílá FastRPC capability
- XML-RPC požadavek, FastRPC odpověď
 - FastRPC klient, který neví o schopnostech server
 - dostane-li FastRPC odpověď, nadále tak komunikuje
- FastRPC požadavek, FastRPC odpověď
 - pokud klient ví, že jde o FastRPC server
 - všechny nové aplikace v Seznamu jsou FastRPC

Metaserver

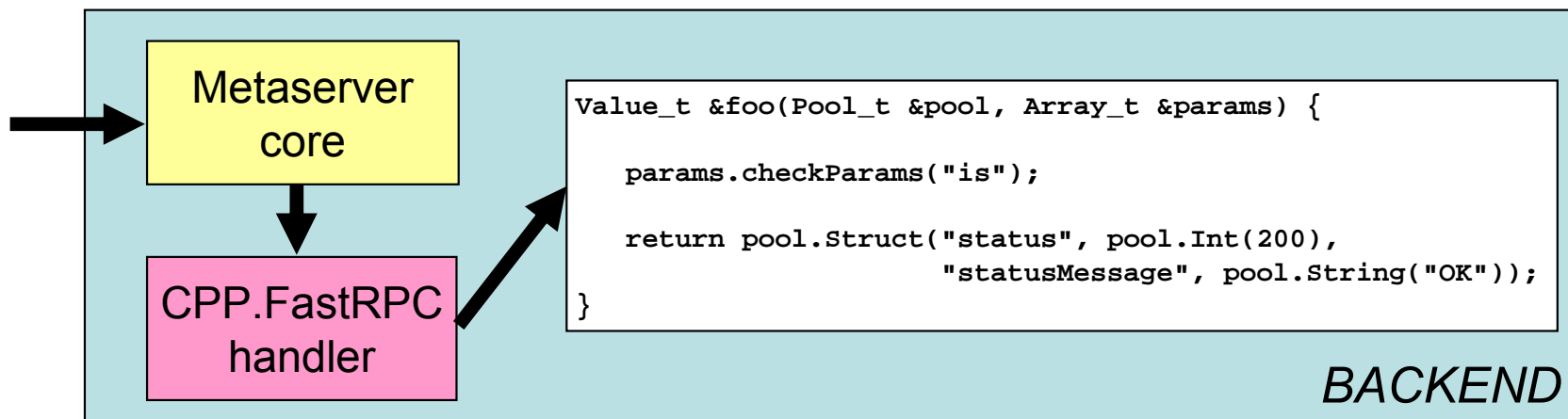
- Univerzální server s **prefork** managementem
- Kostra serveru, která dokáže „pouze“ načíst konfiguraci, inicializovat rozhraní a řídit chod serveru
- Vlastní programátorská práce na tvorbě serveru se omezuje na implementaci protokolového rozhraní formou **handlerů** a psaní logiky serveru
- Používáme handlery Python.FastRPC a CPP.FastRPC
- Handler dostává od metaserveru síťové rozhraní, nad ním implementuje požadovaný protokol a umožní psát logiku serveru v požadovaném programovací jazyce
- Pro C++ je aplikační server „.so“ soubor, pro Python jde o „.py“ soubory prováděné embedded Pythonem

Metaserver

- Prefork management jako u Apache web serveru
spareServers, maxServers, maxRequestPerChild, ...



- Server řídí chod, volá handler na příchozí požadavek,
handler volá metody podle potřeb protokolu



Příklad – server aritmetických operací

- metoda `add(int a, int b)` pro součet dvou zadaných celých čísel `a`, `b`
- metoda `agregate(string op, array x)` provede operaci `op` nad polem celočíselných operandů `x`
- obě metody vrací následující strukturu

```
{  
    int status                200=OK, 405=Chyba  
    string statusMessage     Textový popis výsledku  
    [int result]             Hodnota výsledku pro 200=OK  
}
```

Diskuse...

... a přehled

- XML-RPC, <http://www.xmlrpc.org>
- FastRPC, <http://fastrpc.sourceforge.net>
- Metaserver – jen pro interní potřebu Seznamu 😊
- Python, <http://www.python.org>, <http://www.py.cz>
- Middleware, RPC, ... <http://www.wikipedia.org>

Vlastimil Pečínka <pecinka@email.cz>