

# Od CGI k FastCGI

Ondřej Caletka



5. října 2013



Uvedené dílo podléhá licenci Creative Commons Uveďte autora 3.0 Česko.

- 1 Common Gateway Interface
- 2 FastCGI
- 3 Volací konvence FastCGI
- 4 Programování FastCGI aplikací

# Common Gateway Interface

- první interaktivní obsah na webu
- webserver namísto předání souboru spustí spustitelný program
  - HTTP hlavičky jsou předány jako proměnné prostředí
  - data od klienta jsou předány na standardní vstup
  - standardní výstup je předán klientovi
  - chybový výstup se zapíše do logu webserveru
- podpora ve všech web serverech

# Nejjednodušší CGI skript

```
#!/bin/dash
echo -n "Content-type: text/plain\r\n"
echo -n "Cache-Control: no-store, "
echo -n "no-cache, must-revalidate\r\n"
echo -n "\r\n"
echo -n "Hello World, "
echo -n "your address is $REMOTE_ADDR"
```

# Testování CGI skriptů

```
$ mkdir cgi-bin  
$ cat >cgi-bin/hello.cgi  
...  
$ chmod +x cgi-bin/hello.cgi  
$ python2 -m CGIHTTPServer  
Serving HTTP on 0.0.0.0 port 8000 ...
```

# Nevýhoda CGI

- pro každý požadavek klienta musí server nastartovat proces, který po obsluze klienta zanikne
- není možné držet spojení, např. s databází
- složitější skripty spotřebují spoustu prostředků na inicializaci

Na druhou stranu ale:

- není problém s úniky paměti
- nepřicházejí-li požadavky, jsou prostředky serveru uvolněny

- vzniklo kolem roku 1996 (jako CESNET ☺)
- cílem je zefektivnit CGI
- důraz na vysokou kompatibilitu s CGI
- FastCGI proces nezaniká na konci požadavku, čeká na další
- binární protokol pro komunikaci s HTTP serverem, založený na unixových a TCP/IP soketech
- knihovny pro jazyky C, C++, Perl a Java

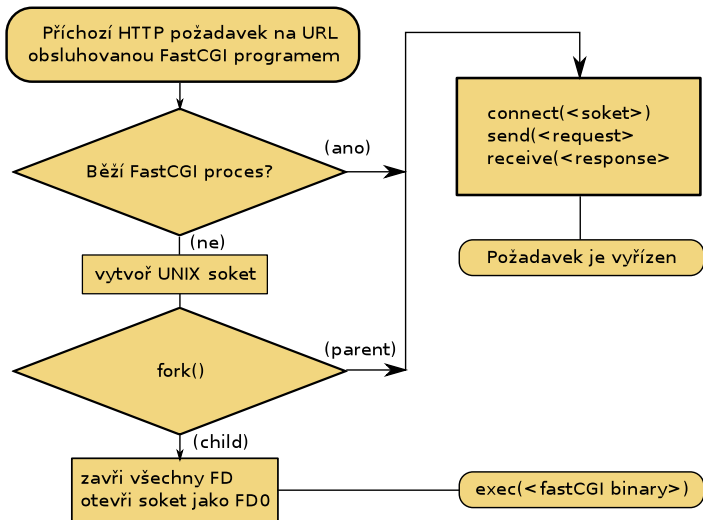
# Volací konvence FastCGI

- 1 ve výchozím stavu FastCGI program neběží
- 2 do HTTP serveru přijde požadavek
- 3 HTTP server založí soket (unix/TCP) a pošle do něj požadavek
- 4 zároveň spustí FastCGI program tak, že mu na FD0 předá „listening socket“
- 5 FastCGI program obslouží požadavek a zůstane běžet
- 6 další požadavky jsou jen předávány běžícímu programu
- 7 server také může spustit více instancí a paralelizovat





# Příklad volání FastCGI



# Zpětná kompatibilita FastCGI

- FastCGI skript může na základě stavu deskriptorů při spuštění detekovat, zda je spuštěn v režimu CGI nebo FastCGI
- je-li mu předán standarní vstup a výstup, přepne do režimu CGI a po obsluze prvního požadavku skončí
- existuje také utilita `cgi-fcgi` umožňující připojit k FastCGI skriptu HTTP server, který podporuje jen CGI

# Jednoduchý FastCGI program

```
#include "fcgi_config.h"
#include <stdlib.h>
#include <unistd.h>
#include "fcgi_stdio.h"
int main()
{
    int count = 0;
    while (FCGI_Accept() >= 0) {
        printf("Content-type: text/plain\r\n"
            "\r\n"
            "Hello World, your address is %s.\n"
            "Request number %d, Process ID: %d\n",
            getenv("REMOTE_ADDR"), ++count, getpid());
    } /* while */
    return 0;
}
```



# Testujeme FastCGI

- neexistuje mnoho FastCGI HTTP serverů vhodných pro *ad-hoc* použití
- díky podpoře zpětné kompatibility stačí spustit FastCGI stejně jako CGI
- k testování chování jako FastCGI můžeme využít `cgi-fcgi`

```
cgi-bin/hellofcgi.cgi
```

```
#!/usr/bin/cgi-fcgi -f  
-connect /var/tmp/fcgiapp.sock /var/tmp/hello.fcgi
```

Nastartujte FCGI pomocí

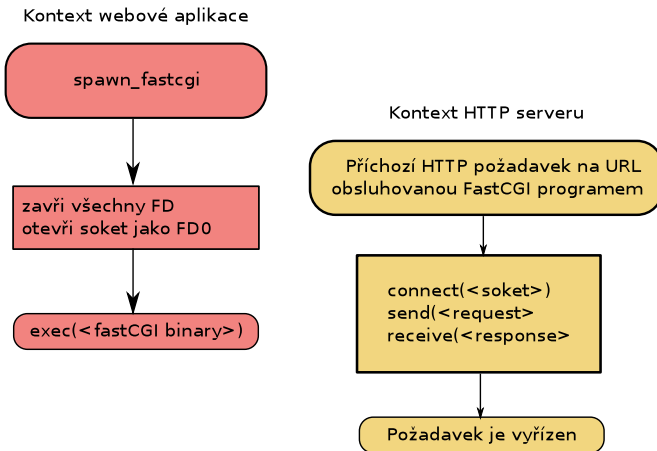
```
cgi-fcgi -start -connect /var/tmp/fcgiapp.sock  
/var/tmp/hello.fcgi 2
```



# FastCGI jako samostatný server

- některé HTTP servery (jako nginx) nepodporují spouštění procesů FastCGI
- očekávají adresu soketu, na kterém bude poslouchat FastCGI server
- vytvoření soketu a jeho předání FastCGI programu zařídí buď `cgi-fcgi` nebo `spawn-fcgi` od tvůrců Lighttpd
- výhodou je, že FastCGI skript může běžet pod jiným uživatelem než HTTP server

# FastCGI jako samostatný server



# PHP FastCGI Process Manager

- vylepšený spawner PHP interpreteru pro FastCGI (`php-cgi`)
- preferovaná cesta nasazení PHP prostřednictvím FastCGI
- dokáže spouštět PHP workery podle potřeby
- umí restartovat worker po  $n$  požadavcích
- je možné provozovat nezávislé pooly (pro každého uživatele samostatný)

# Jméno URL u FastCGI

- u CGI je zvykem mít jeden CGI pro jednu funkci
- pro FastCGI naopak dává smysl kombinovat všechny funkce do jednoho programu
- volaná URL se detekuje třeba pomocí proměnné `SCRIPT_NAME`
- přístup Apache: pomocí `modRewrite` přepíšeme všechna URL na FastCGI soubor
- přístup Lighttpd: určíme podoblast URL, která bude celá obsluhovaná FastCGI skriptem



# FastCGI jako Authorizer

- FastCGI může fungovat také jen jako autorizační brána
- Příchozí požadavek je předán FastCGI aplikaci
- Když FCGI odpoví 200 OK, je uživateli namísto výstupu FCGI poslán soubor z cesty v URL
- Příklad využití: download server pro e-shop:
  - Při příchodu požadavku FCGI aplikace zkontroluje, zda stahuje oprávněný zákazník
  - Pokud ano, vlastní odbavení statického souboru provede HTTP serverech
  - Zjednodušení FCGI aplikace a úspora prostředků
- Nepodporováno v nginx



Děkuji za pozornost

