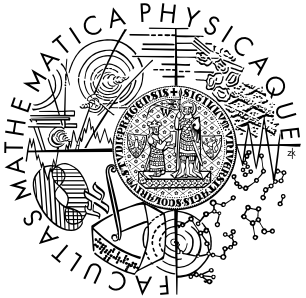
Skenování otevřených zdrojů

Leo Galamboš

Katedra Softwarového Inženýrství
Matematicko-fyzikální fakulta UK

2008-05-19



Obsah

Věcný obsah

- technologie robota: DNS, HTTP, zpracování odkazů
- navigace robota po webu
- architektura Egothor2 robota

Web – Osoby a obsazení

- webový server (httpd) poskytuje webové dokumenty
- webový dokument je opatřen metadaty (formát, délka, čas modifikace. . .)
- webový klient si stahuje webové dokumenty z webového serveru nad HTTP(S)
- robot[1] stahuje automaticky a systematicky požadované dokumenty

[1] bot, crawler, spider

Web – komunikace

Kritická místa HTTP

- Content-Type k URL sděluje server: *.html⇏text/html
- skutečný cíl neurčuje IP, ale IP + Host v hlavičce

```
GET /robots.txt HTTP/1.0
Host: www.example.com

HTTP/1.1 200 OK
Date: Sat, 18 Feb 2006 00:44:20 GMT
Last-Modified: Mon, 02 Feb 2004 21:34:29 GMT
Content-Length: 402
Connection: close
Content-Type: text/html

<html><title>....
```

Odlišnosti od klasických knihovních full-textů

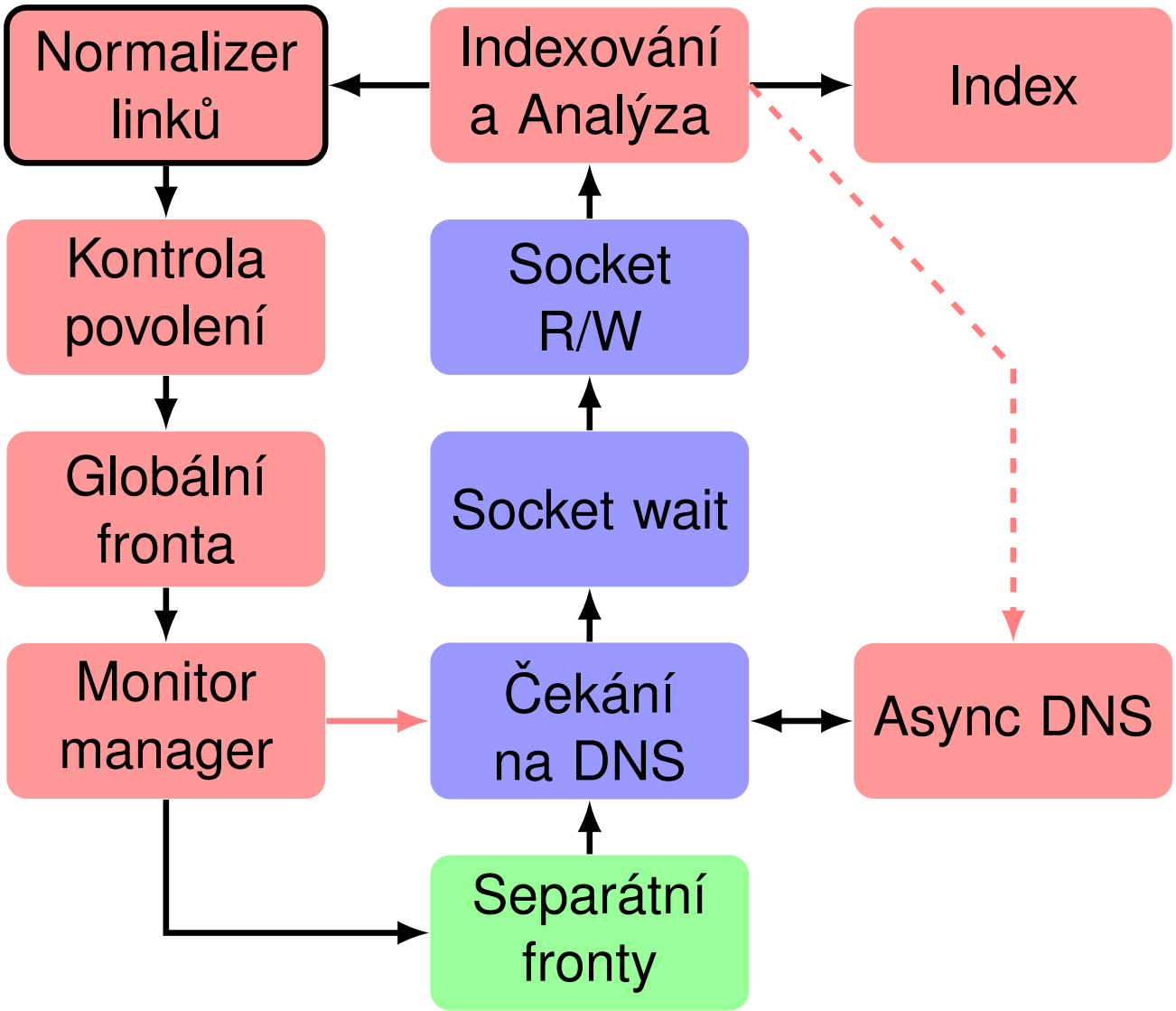
- až za běhu objevujeme co ještě stahovat
- nutnost tahat neagresivně, ale přesto co možná nejvíce

Robot	strojů/CPU	str/sec	tok (MB/s)	rok
Googlebot	4/?	25-32	0,2	1998
Mercator	1/2	55	0,84	1999
Mercator	4/8	72	1,72	2001
Xyro	4/?	12	?	2001
Nutch	1/?	?	0,5	2004
Become	50/?	100	0,3?	2004
Egothor 1	1/2	40	0,6	2004
Dominos	5/?	154	0,9	2004
Egothor 2	1/2	70-100	1,5-5,0	2005,2007
Larbin	1/2	80	2,1	2006
VN	9/14	2?	0,1	2006

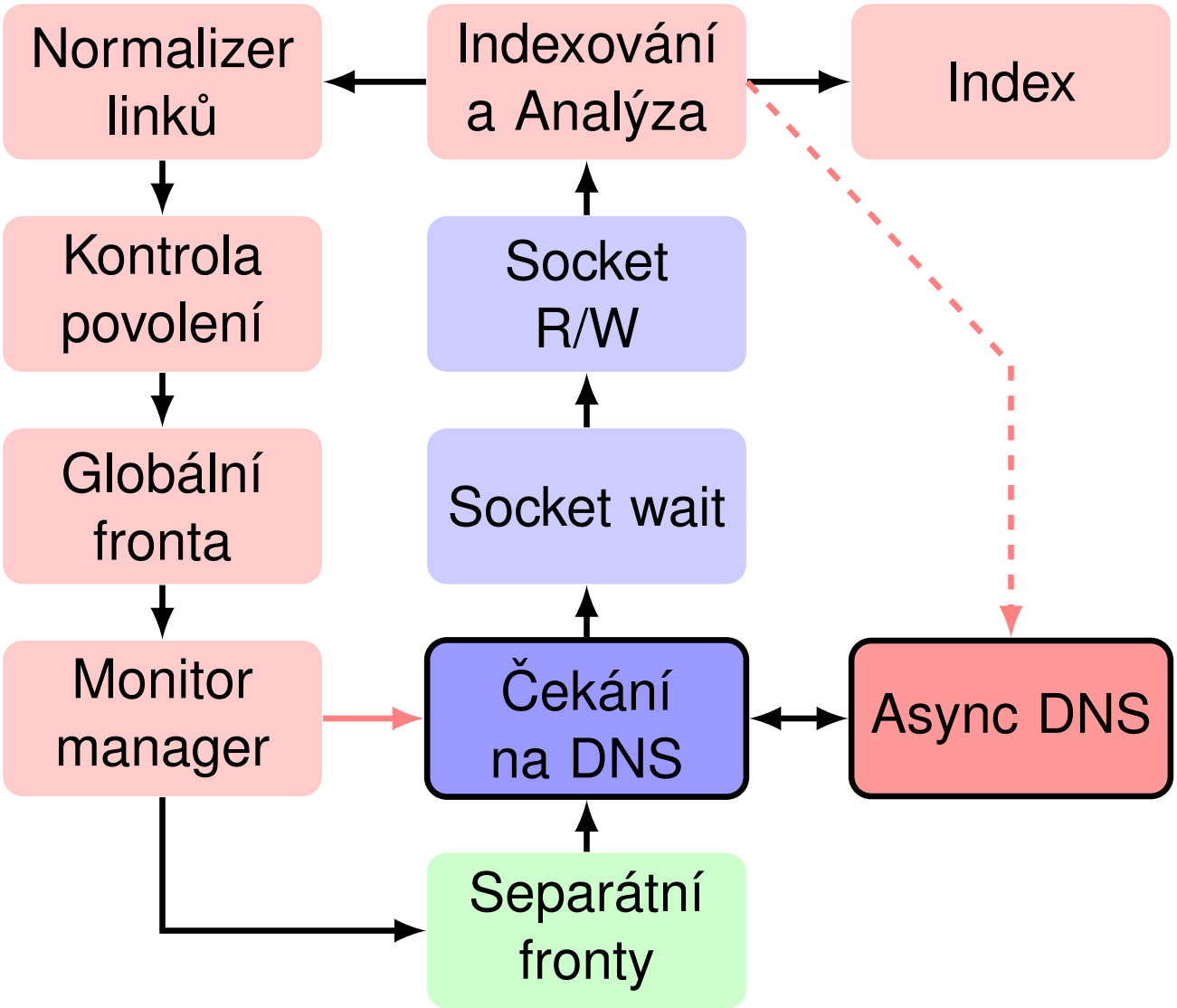
Otázka

Co způsobuje tak velké rozdíly ve výkonu výkonu?

Zjednodušená struktura robota



DNS



Problematika DNS

Robot se snaží netahat dlouho z jednoho serveru $\Rightarrow$ více dotazů do DNS a snížení "lokality" přístupu do cache.

Reálný provoz

- nedodržují se expirační časy
- obnova zón (typicky) v idle režimu
- `gethostbyname(3)` neefektivní
  - cache jako spojový seznam (JAVA)
  - neumožňuje souběžné dotazy

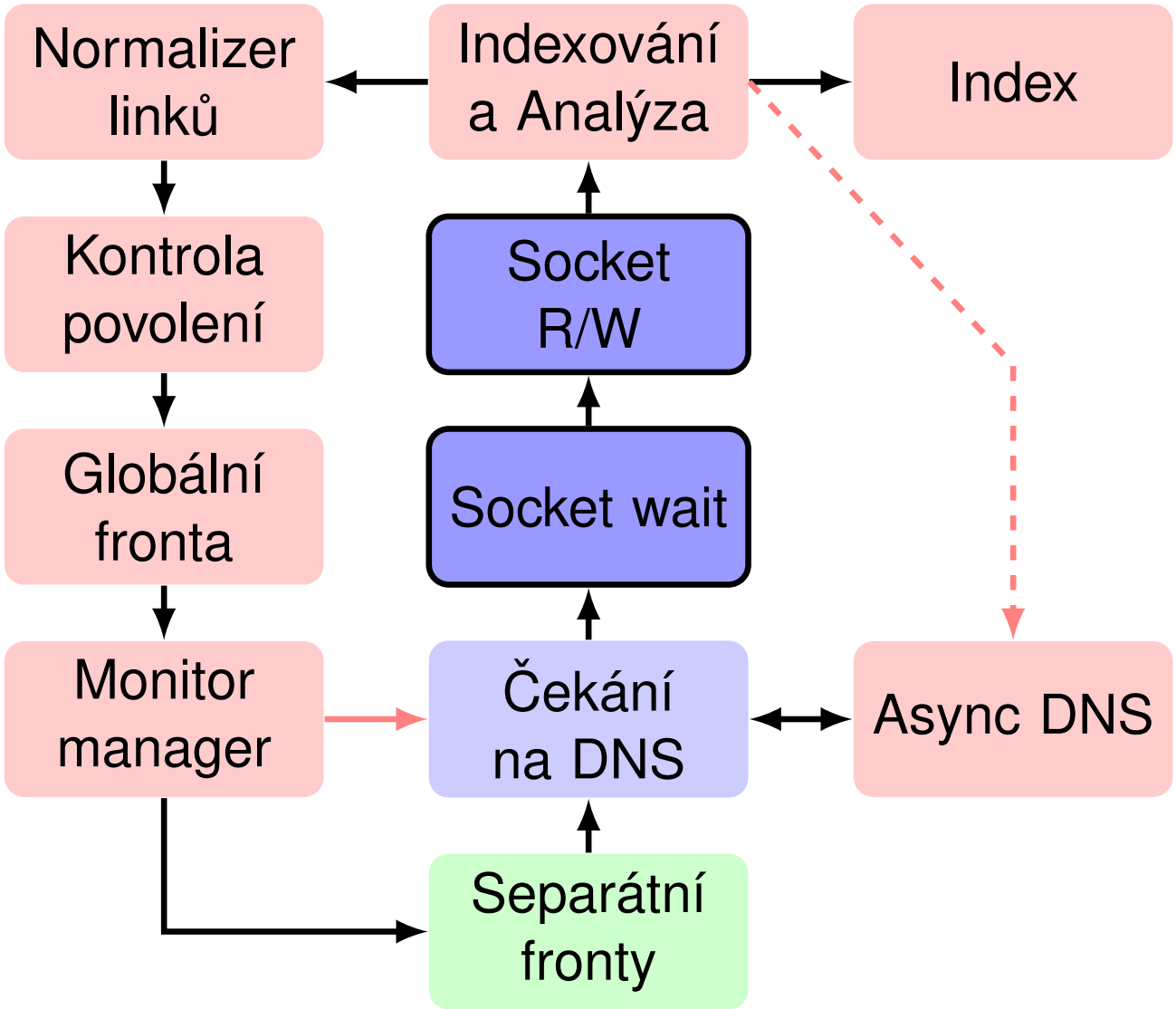
Řešení

- více resolverů s velkými cache (Mercator až 3 s 1GB)
- používá se prefetch
  - jeden dotaz může implikovat až desítku UDP dotazů
  - $\Rightarrow$ komunikace mezi link parser a async DNS modulem

Praxe

- ADNS knihovna (asynchronous DNS client library)
- Mercator (Compaq): úspora po přepsání DNS knihovny, čas v DNS z 87% na 25%
- Egothor2: DNS modul není brzdou (dané technikou plánování front)

Stahování via HTTP



Implementace

- latence při stahování $\Rightarrow$ stahuje se řádově $n \times (100 - 1000)$ stránek najednou
  - vědecký robot obvykle řádově desítky až stovky
  - distribuované farmy tisíce na jednom uzlu
- je nevýhodné používat systémová vlákna, používá se `select(2)`

Blokované sockety

- používá se statický počet vláken
- vlákno má vlastní stack, stav a přístup do sdílené paměti

Výhody

- snadná implementace
- ladění a krokování (JAVA)

Nevýhody

- synchronizace vláken stojí výkon
- při pádu vlákna obtížná detekce a oprava (C/C++)
- vlákna dotahují relativně nezávisle $\Rightarrow$ náhodné přístupy do repository, z toho plynoucí velký interleave a snížení výkonu

Neblokované sockety

- funkce `select(2)` s následným zpracováním vůči nosnému datovému bloku

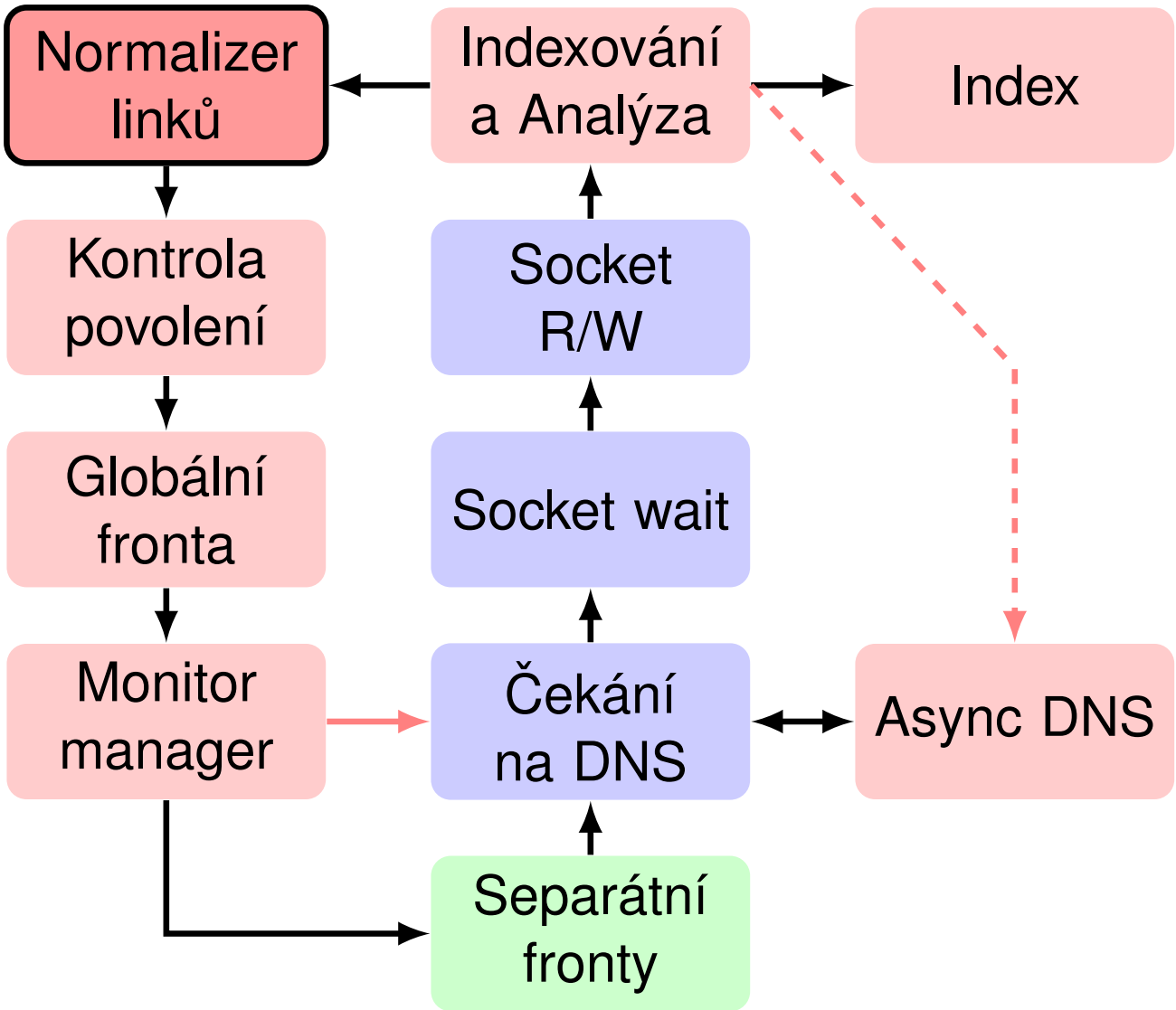
Výhody

- velký výkon
- možnost “sériového výstupu”

Nevýhody

- paměťová náročnost, správa heap-u
- problém se správou timeoutů

Normalizace



Extrakce linků, normalizace

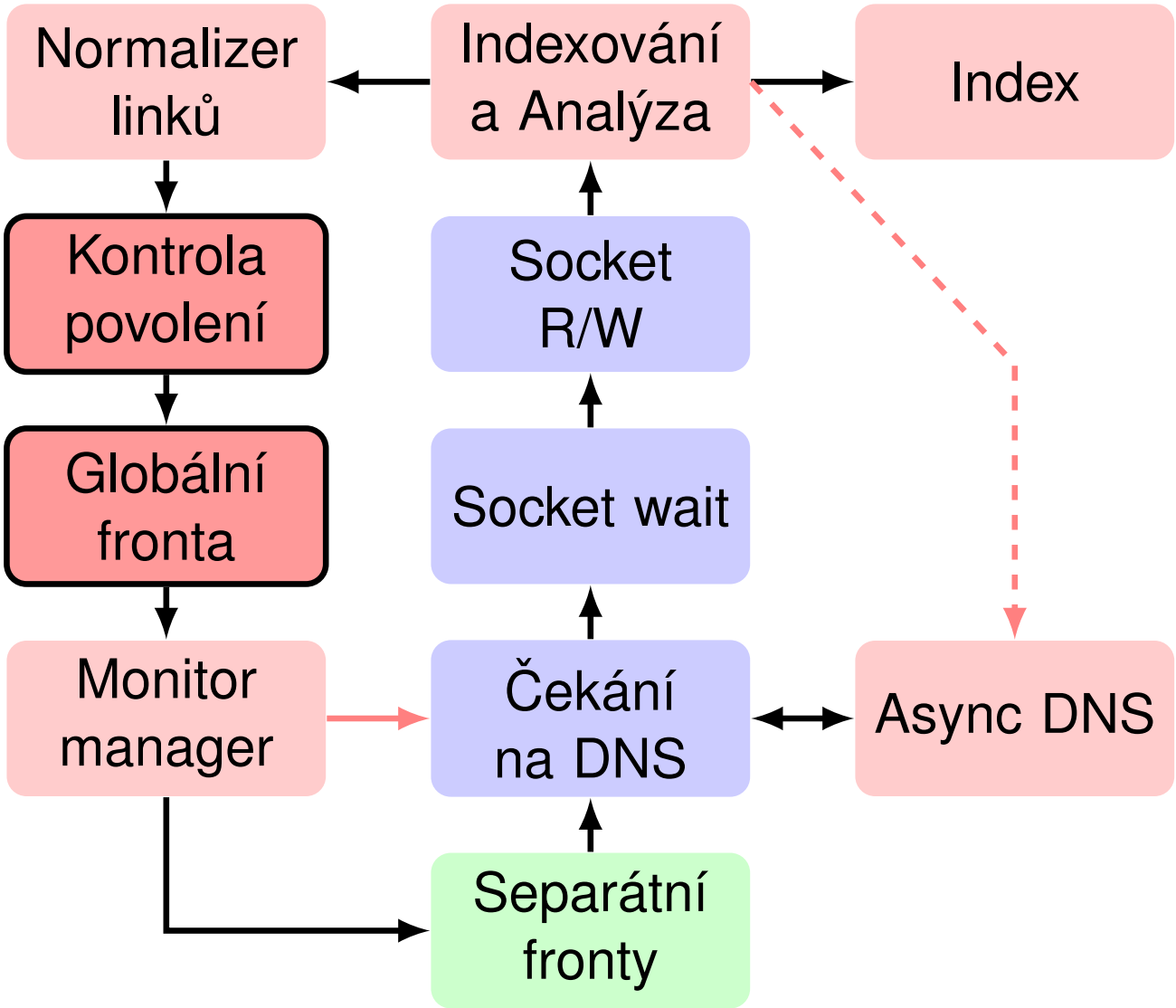
Snaha o omezení duplicitních URL:

- čisté porovnání nestačí
  - `--/~plha/--/%7Eplha/--/%7eplha/--`
  - pořadí parametrů za ?
- duplicitní obsah se zjišťuje až dodatečně

Normalizace:

- alespoň částečně snižuje počet duplicitních URL
- klasicky:
  1. protokol i hostname malými písmeny
  2. přidání čísla portu
  3. normalizace znaků (escape sekvence)
  4. vyčištění fragmentů `/./` a `/../`
  5. -36%: index.htm(l)
  6. -30%: egothor.org v. www.egothor.org
  7. -5%: /dir v. /dir/
  8. +26%: auto-redirect

Eliminace duplicit



Eliminace známých URL

Problém

Při 1000URL/sec produkuje robot až 40000 “nových” URL. Zapisuje se i struktura odkazů, a proto musíme znát příslušná ID dokumentů. Jak to technicky zvládnout?

1. URL se převede na pevnou délku (MD5 16B): *key*
2. klíč přidáváme do hash tabulky – problémy:
   - tabulka se většinou nevejde do RAM (Larbin)
   - dobrý převod na pevnou délku nám rozbije lokálnost

Řešení

Standardní způsob

- lokálnost si zase zavedeme:
  $key := (key_{hostname}; key)$
- lze použít B-tree

Dominos přišel s novinkou: Judy-array.

Bezedný web

Nevhodná přemapování na straně serveru nebo httpd s chybami[2] mohou vytvořit bezedný web a z toho plynoucí pasti[3]:

- můžeme omezit délku URL, ale dokonalé to není
- je možné vytvářet statistiky a anomálie ručně ošetřovat
- lze detekovat opakující se fragmenty v URL
- pomáhá i nestahování stránek s nízkým rankem

Příklady

```
.../archiv/archiv/archiv/archiv/05.html
.../calendar.php?date=1456-02-28
```

[2] Některé starší verze Apache
[3] Spider traps

Eliminace duplicit

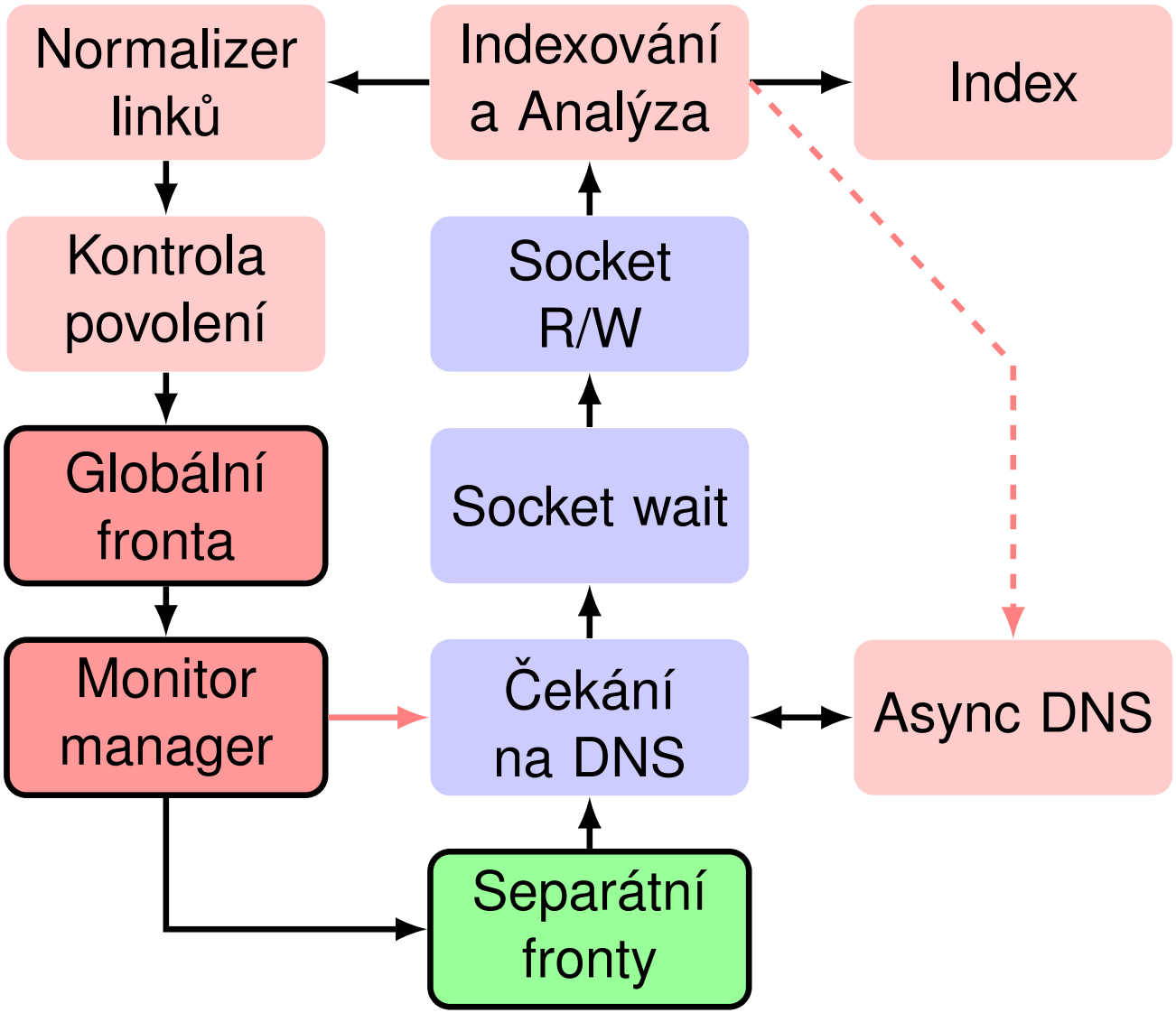
Robustnější řešení přes detekci duplicit

Sledujeme unikátnost dvojic ($hash_{dokument}$; *rel*), kde *rel* je relativní odkaz.

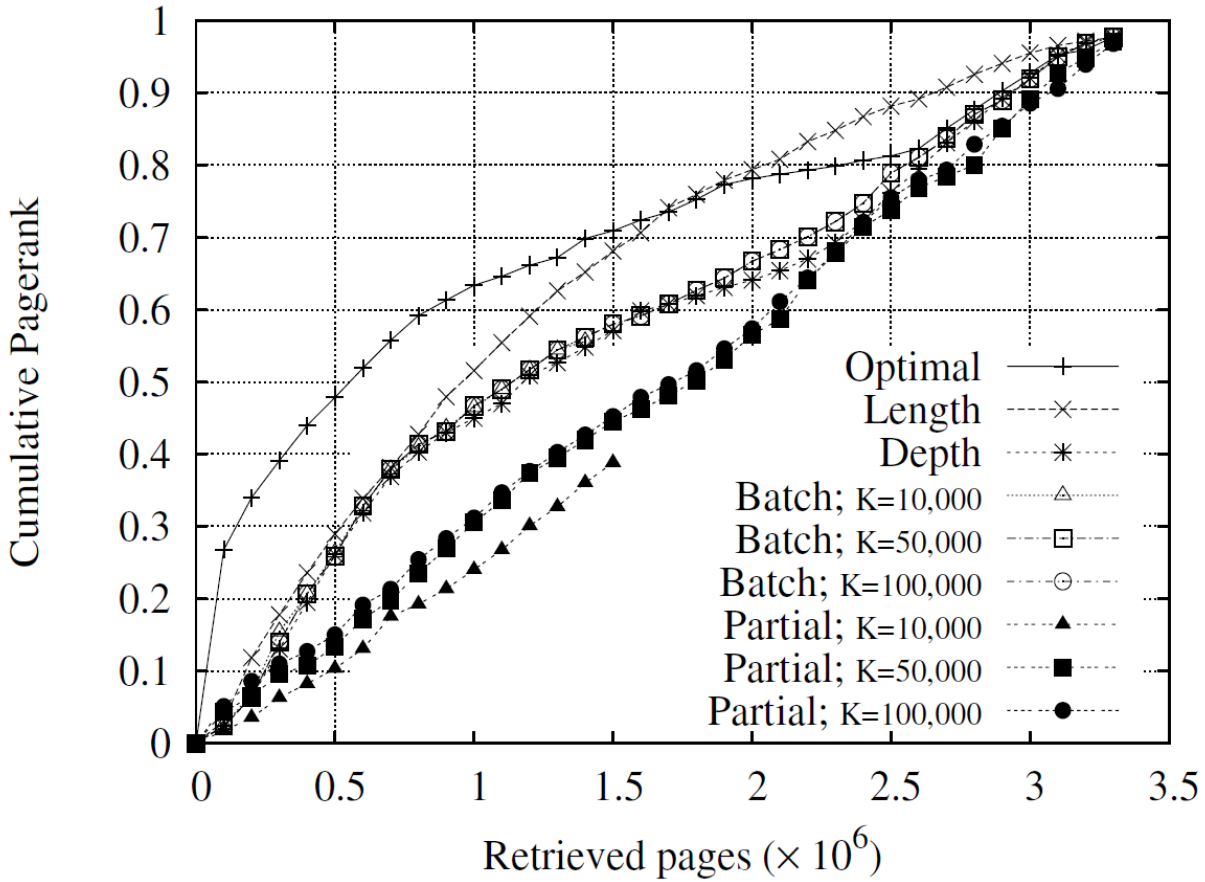
Dokonalé řešení neexistuje

Omezené řešení nabízí shingling (šindlování), které je schopné detekovat duplicitní obsah.

Navigace robota



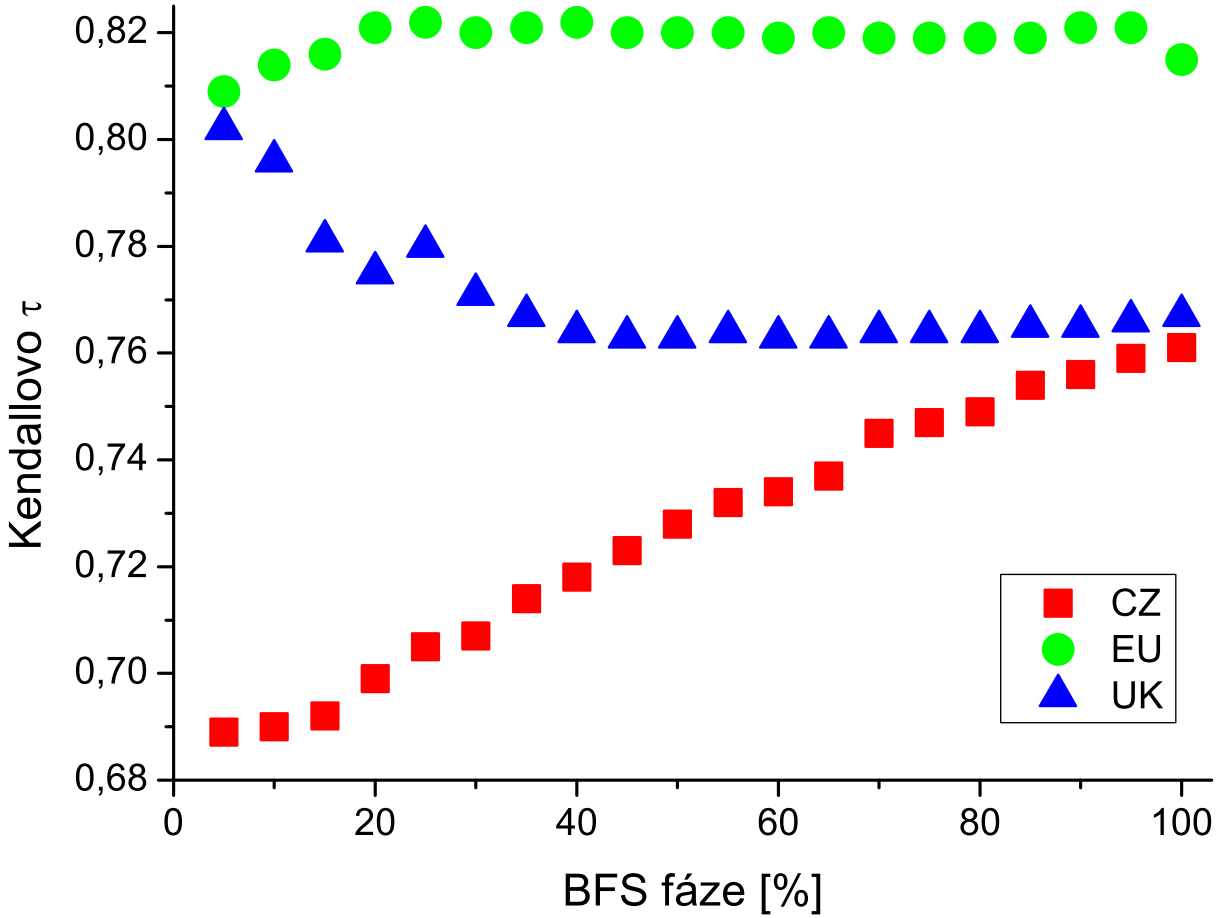
Procházení webu



metoda	BFS	DFS	OPIC	PD	NVW
konvergence	3	1	1*	5	
rychlost	2			1	**

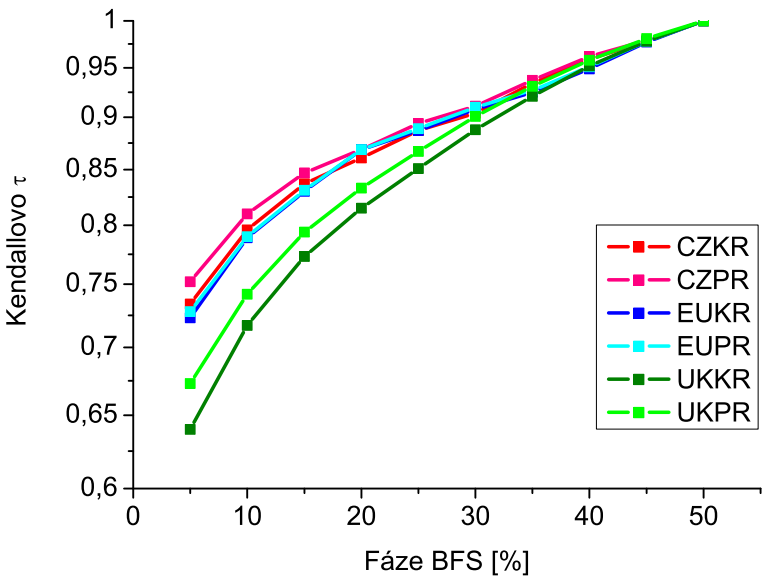
[Castillo et al.]

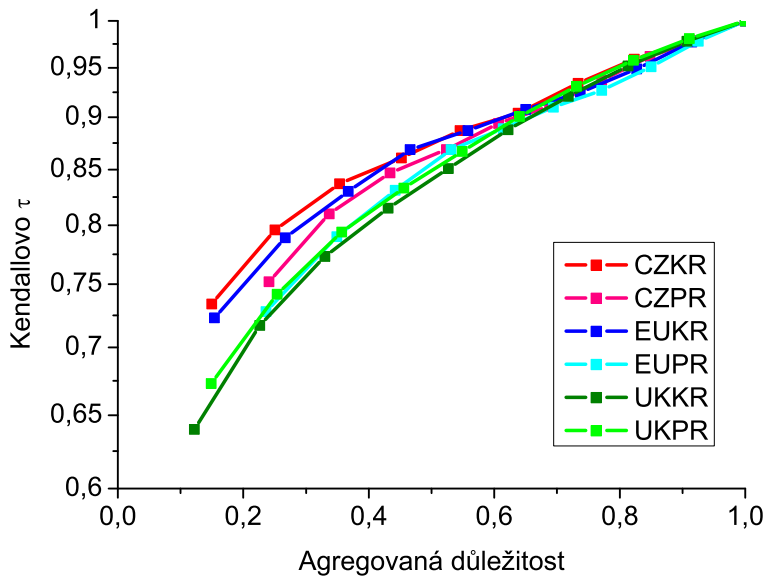
Anti SPAM = K-rank



doména	CZ	UK	EU (Uni)
URL	200M	90M	180M
odkazů	3,0G	1,1G	2,1G

Anti SPAM = K-rank

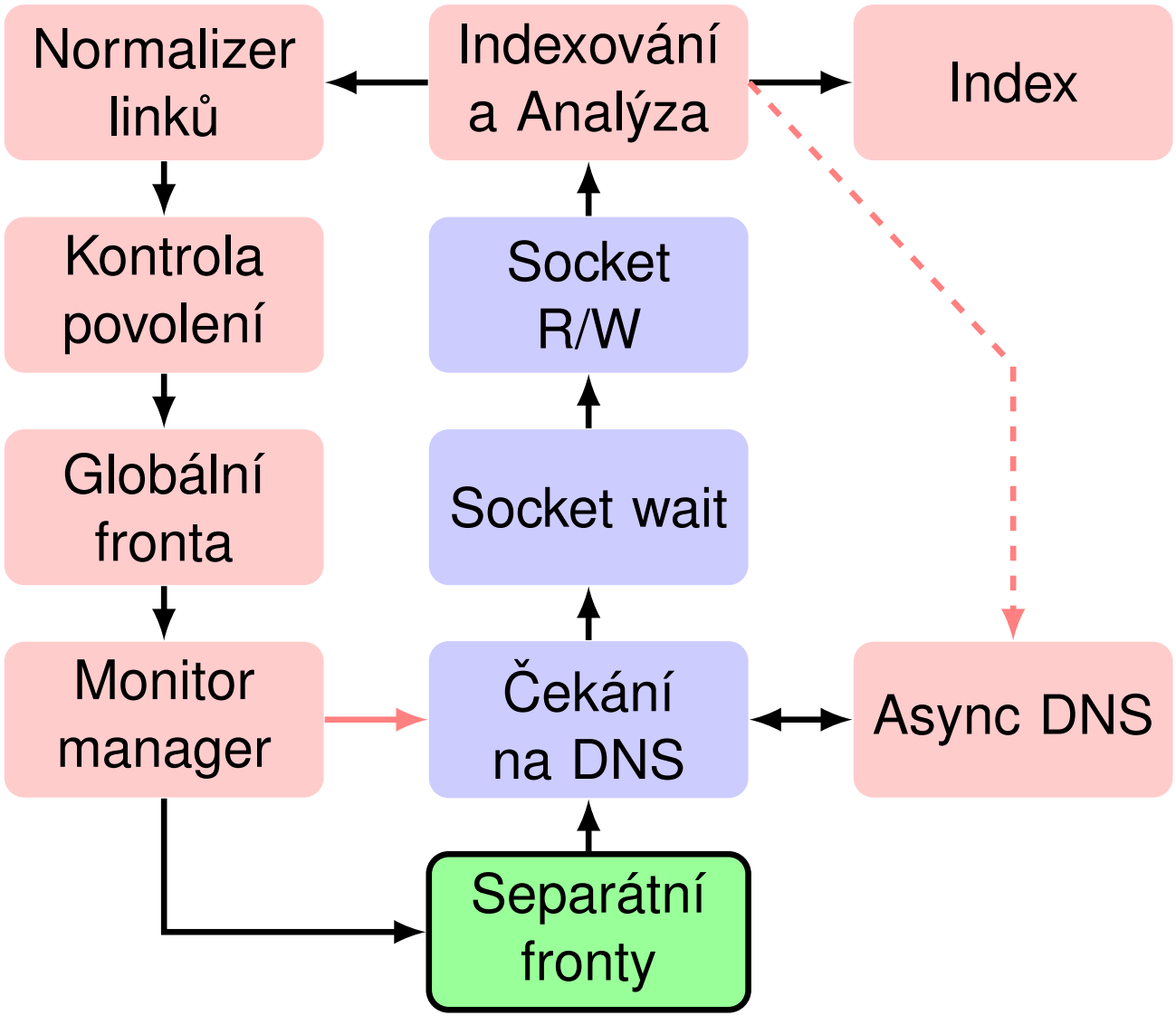




K-rank. . .

potřebuje až o třetinu menší matici než Page rank

Aktualizace dokumentů

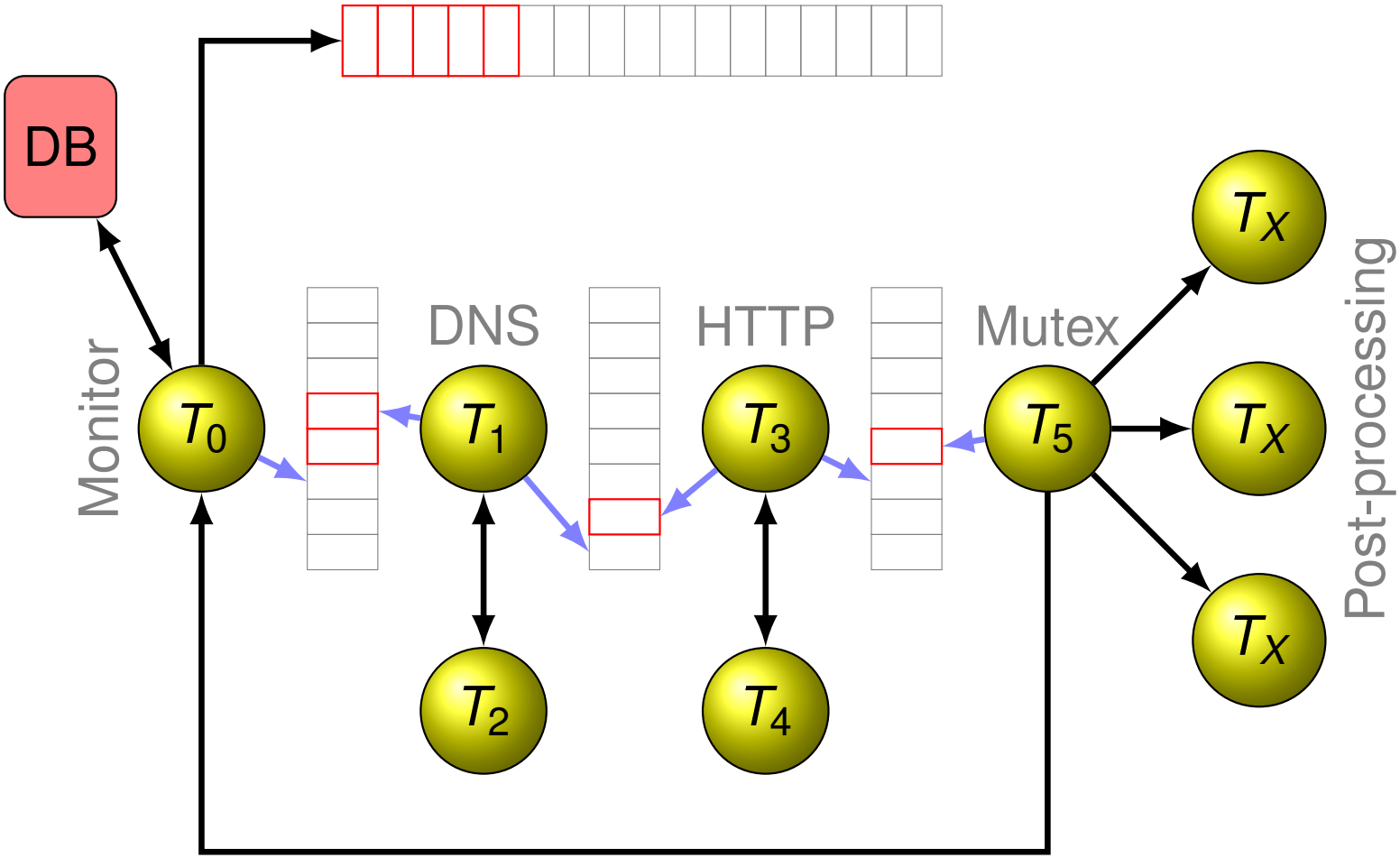


Aktualizace dokumentů

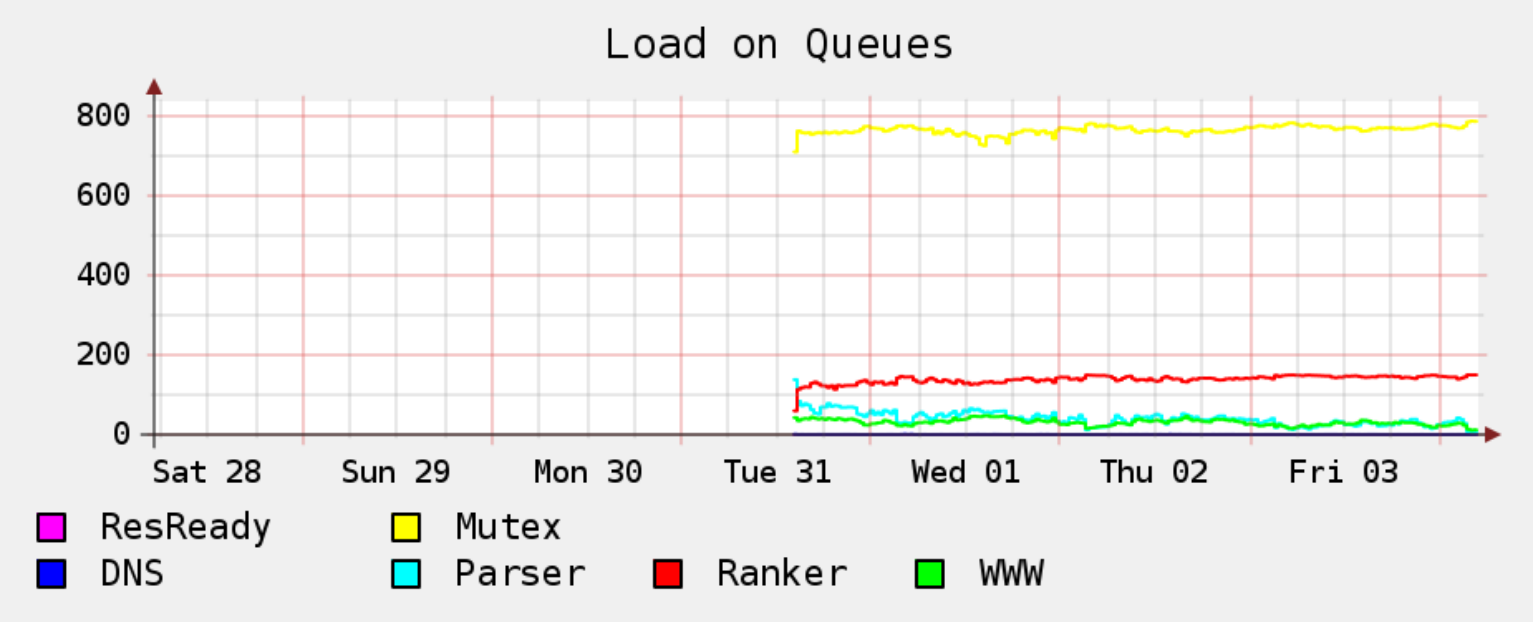
1. HTTP nabízí:

   If-Modified-Since ignoruje se

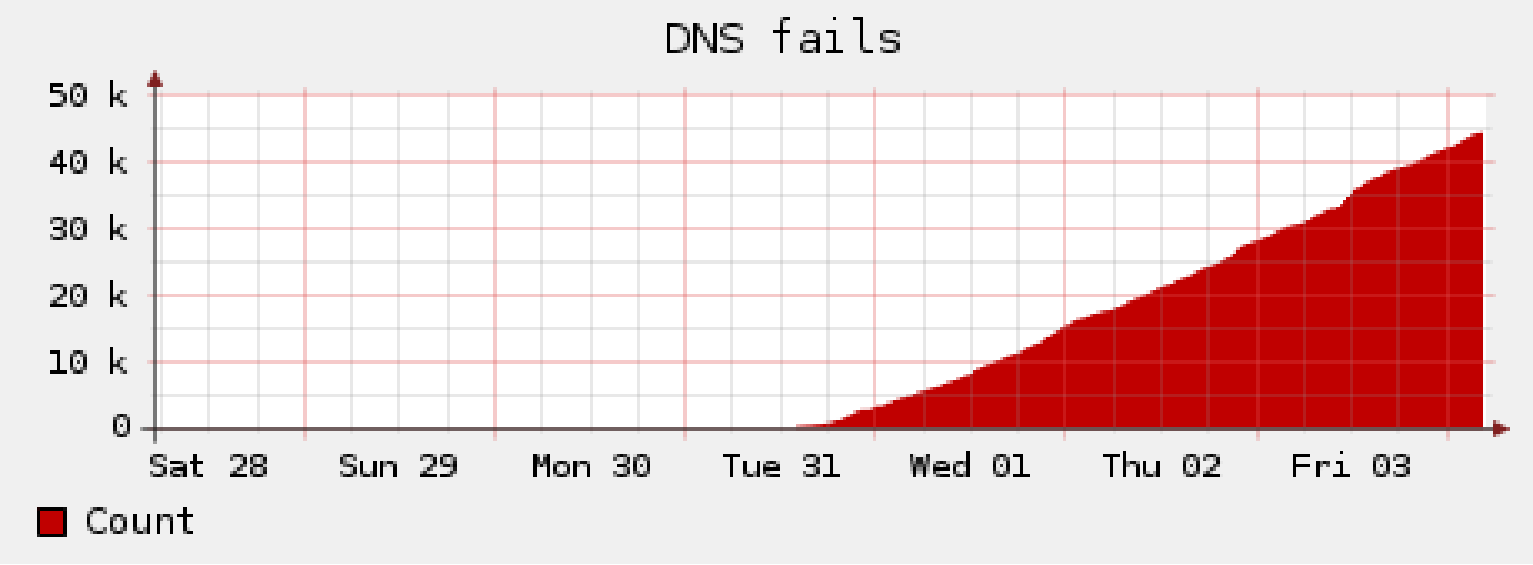
   Expire ignoruje se
2. robot se sám učí frekvenci aktualizací
3. nabízená řešení (definitivní řešení zatím není):
   - zohlednit frekvenci změn každého jednotlivého dokumentu (egothor2)
   - zohlednit významnost dokumentu nebo serveru
   - zohlednit významnost změny (obtížný NLP problém)

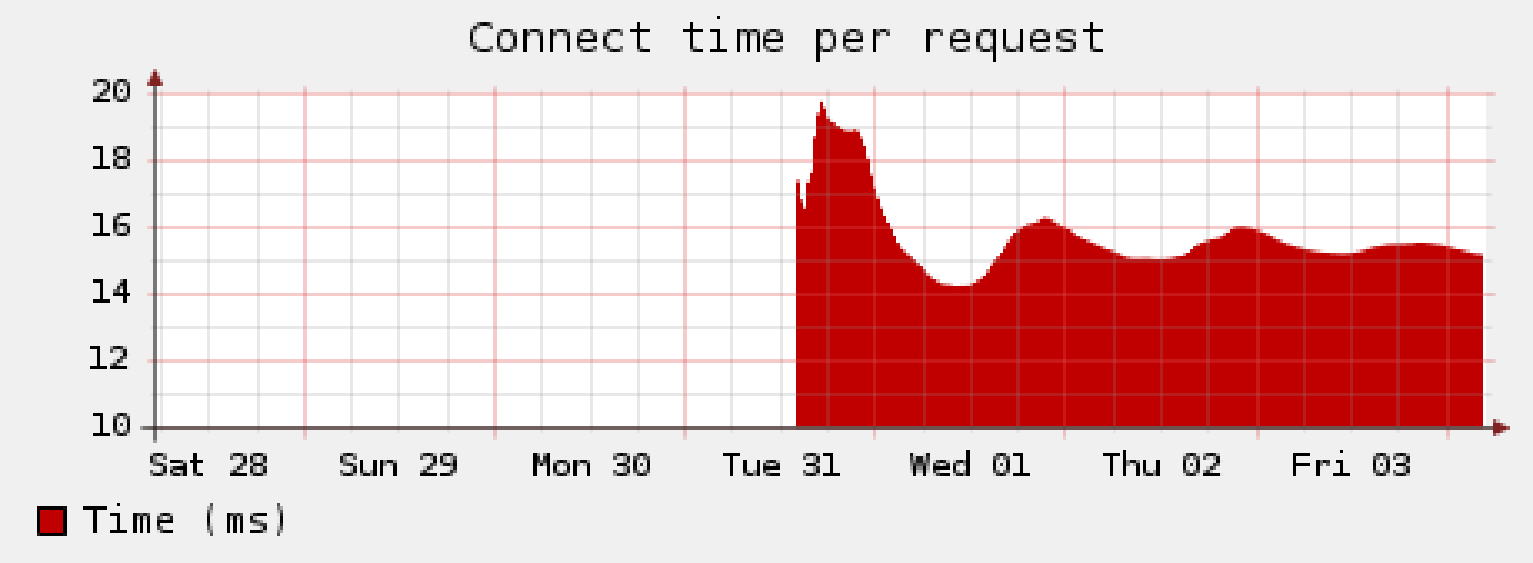


Zatížení front

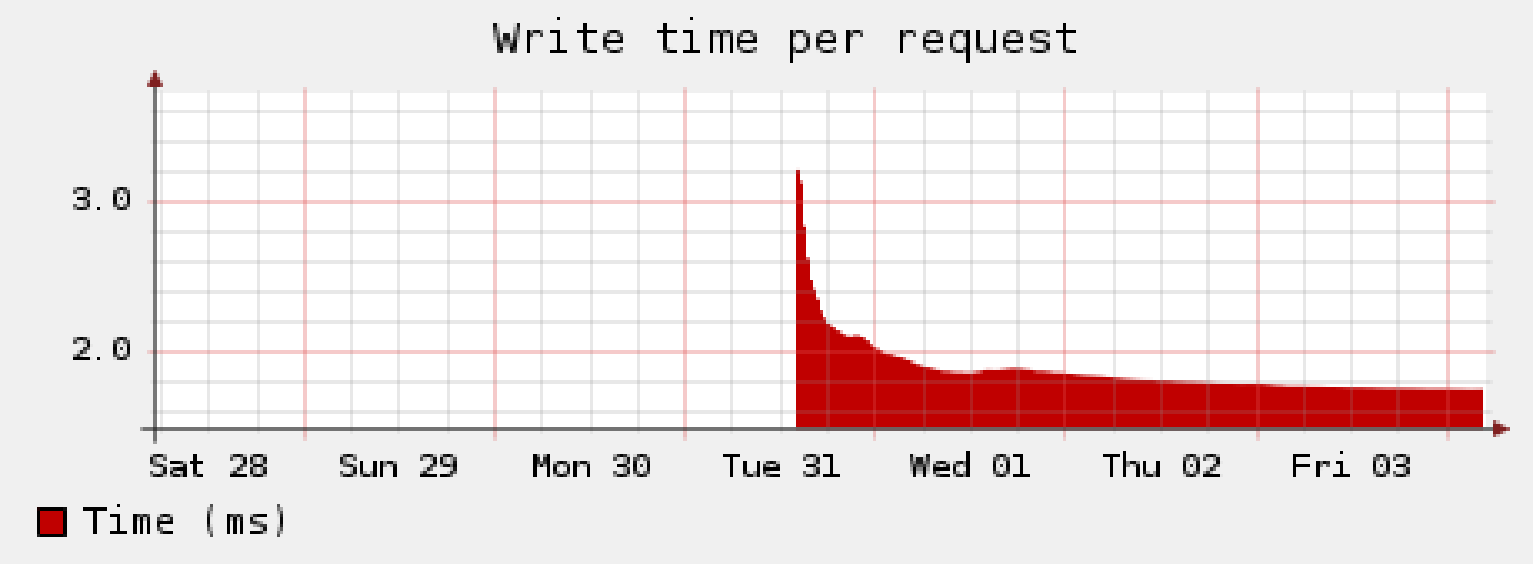


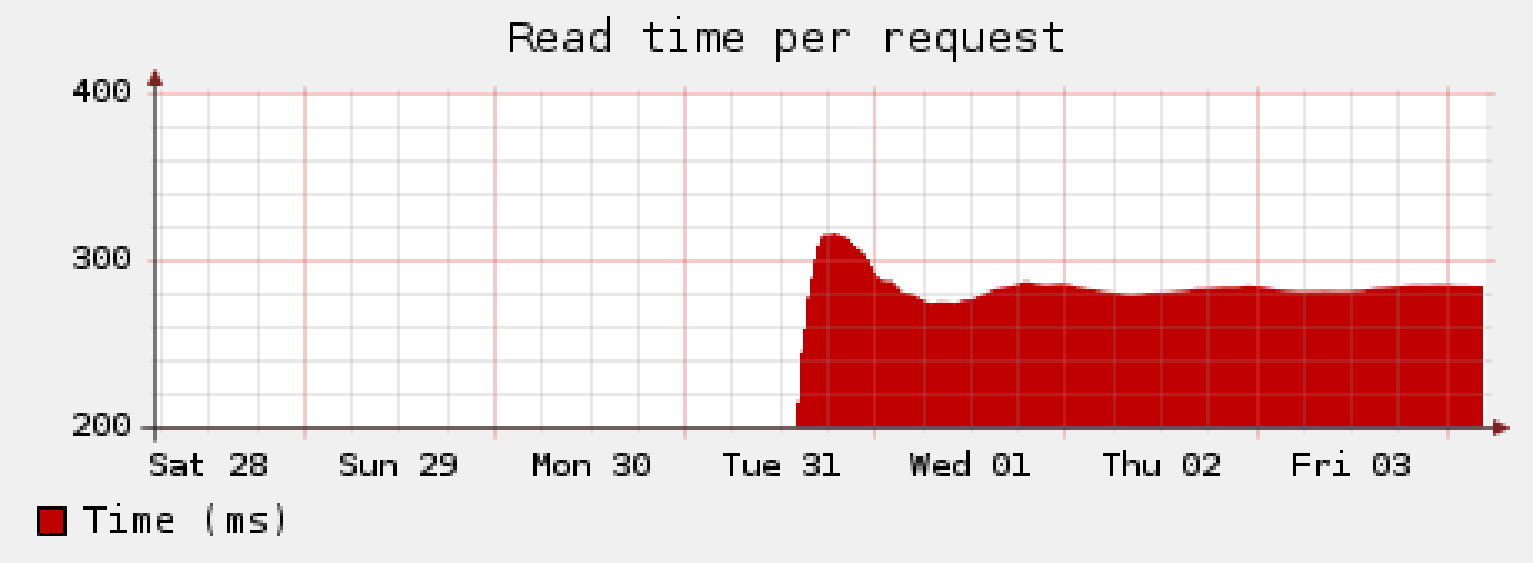
Inicializace spoje



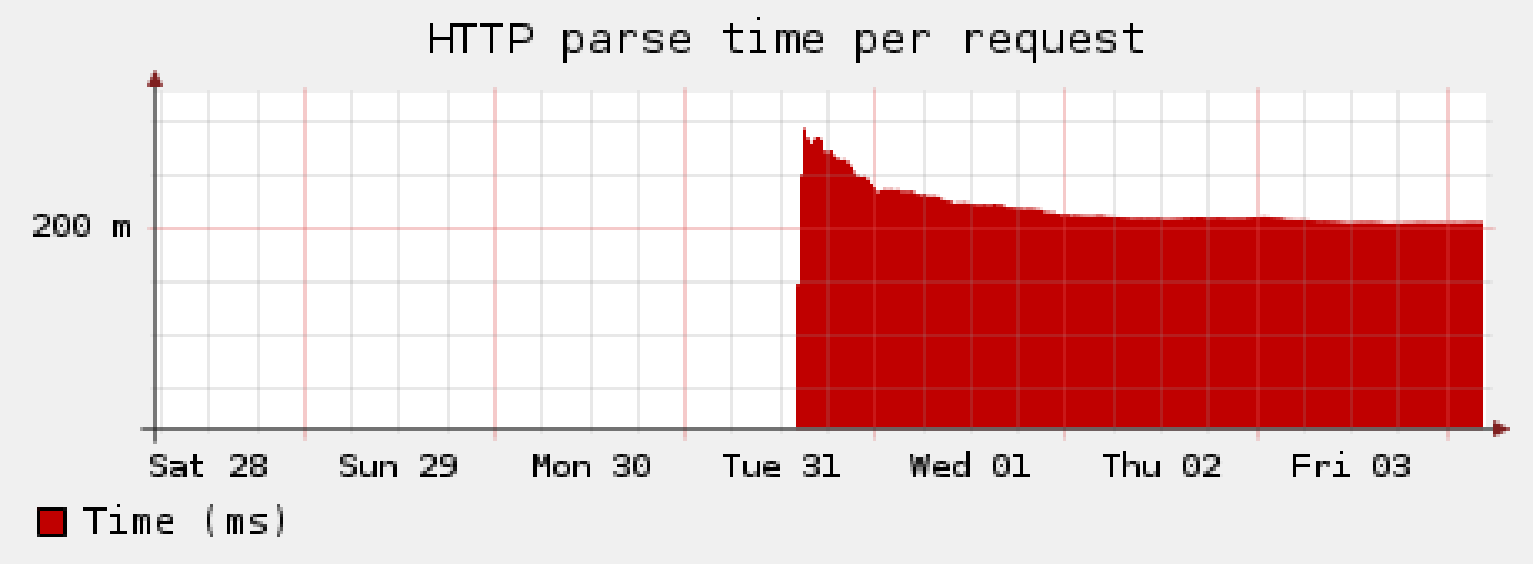


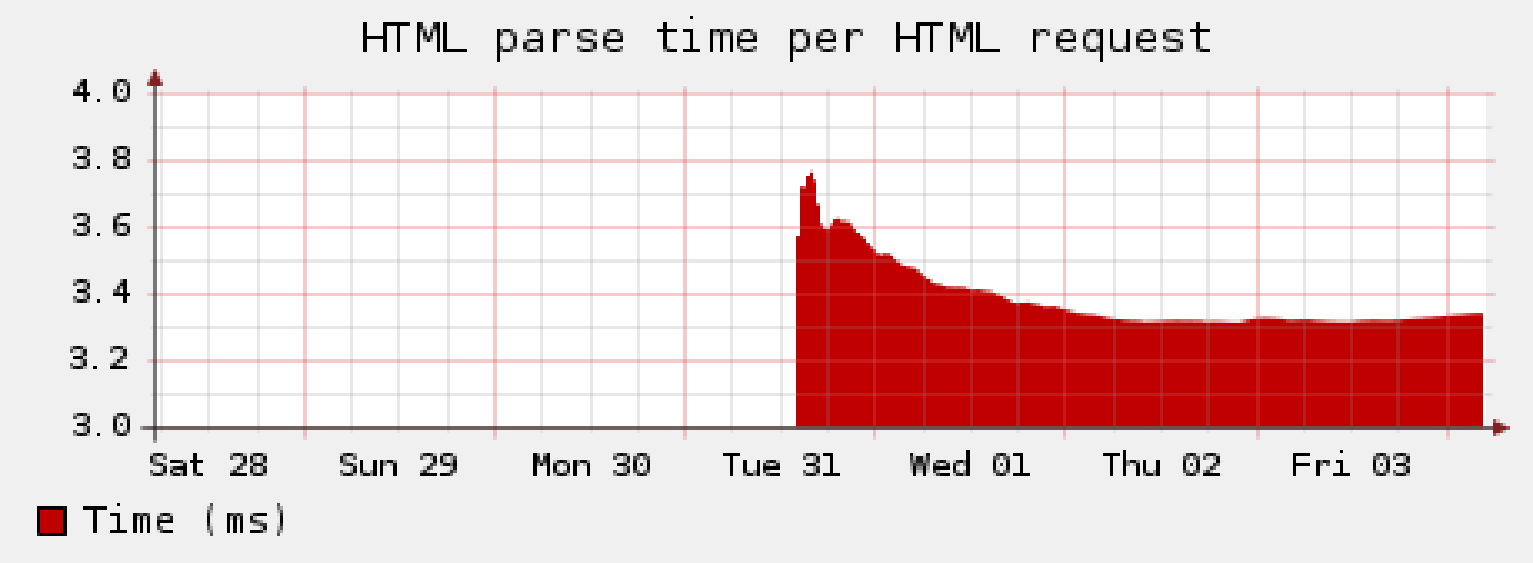
Zápis a čtení do socketu



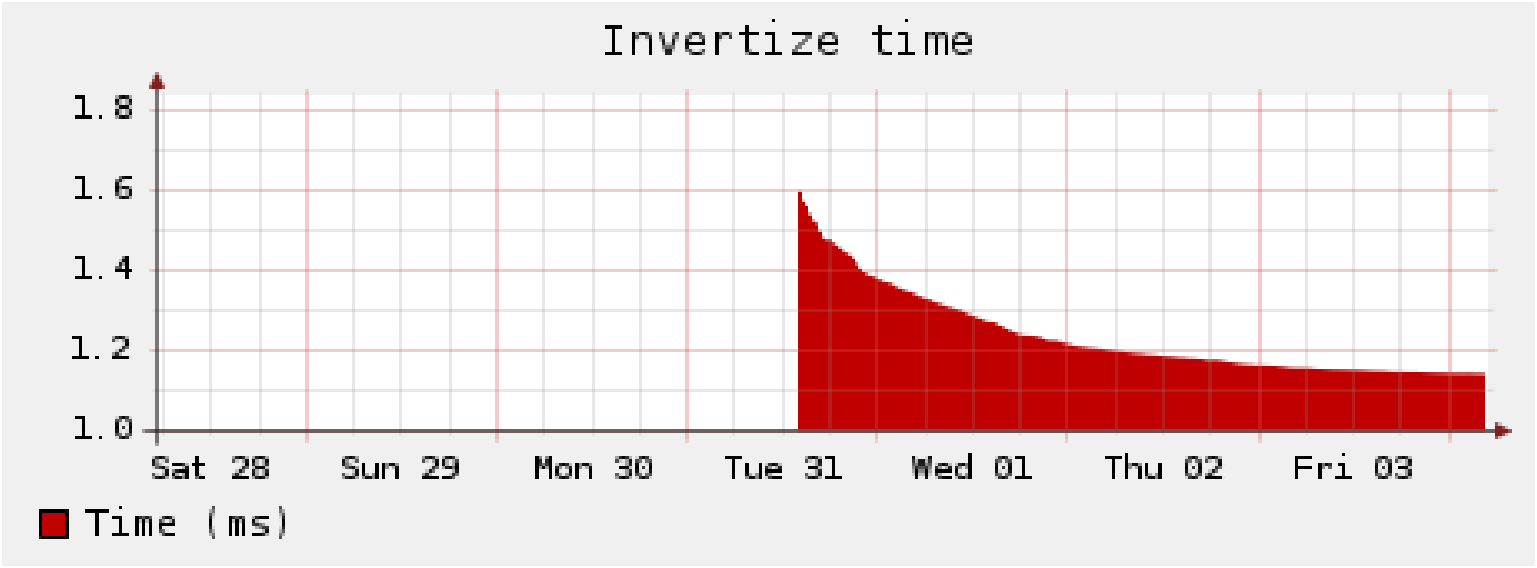


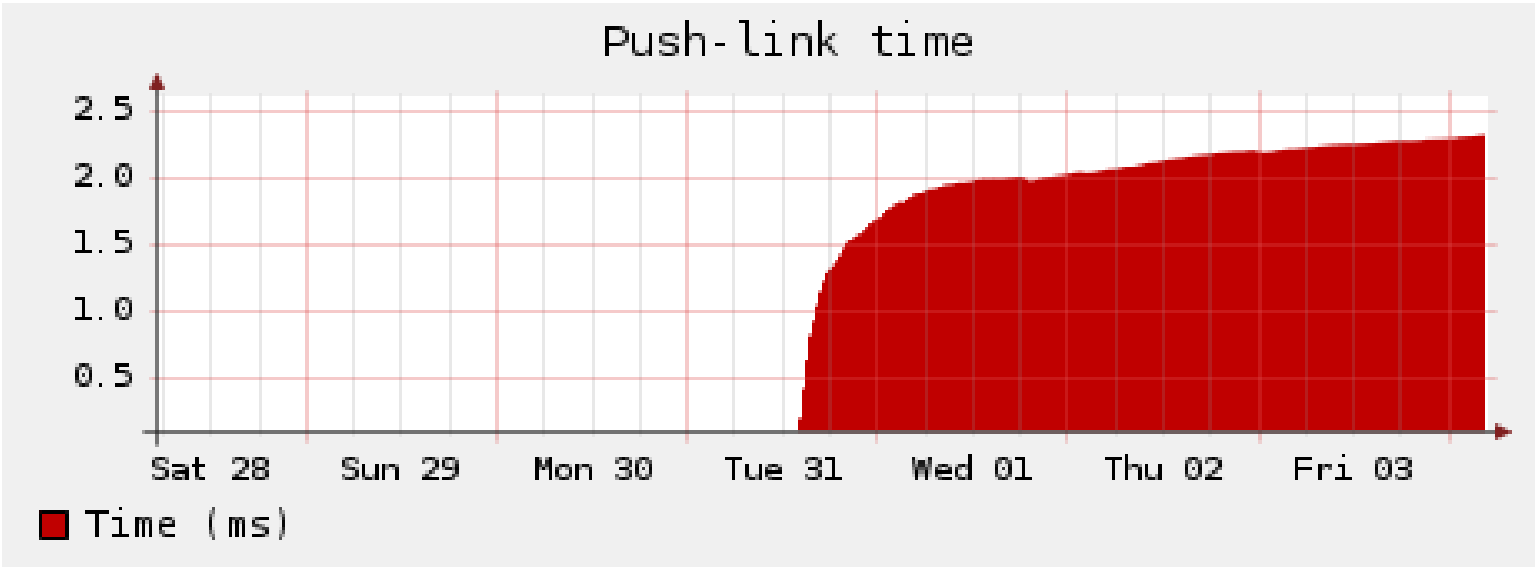
Zpracovaní HTTP a HTML





Zpracování dokumentu





Závěr

- plánovač robota může ovlivnit kvalitu výsledků
- hrubá síla versus komplikované plánování
- test na (ne)existenci URL
- DNS (Mercator) $\Rightarrow$ Indexer-Async DNS hook
- duplikáty - základní heuristiky
  - -36%: index.htm(l)
  - -30%: egothor.org v. www.egothor.org
  - -5%: /dir v. /dir/
  - +26%: auto-redirect
- SPAM...


Castillo, C.
Effective web crawling.
SIGIR Forum, 39(1), 55-56, ACM Press, NY, USA, 2005.


Cho, J., Garcia-Molina, H.
Estimating frequency of change.
ACM TOIT, 3(3), 256-290, 2003.


Craswell, N., et al
Performance and cost tradeoffs in web search.
In Proc. of the 15th ADC, 161-169, Dunedin, NZ, 2004.


Cutting, D., Pedersen, J.
Optimization for dynamic inverted index maintenance.

In Proc. of the 13th SIGIR, 405-411, ACM Press, 1990.

Fagni, T., et al

Boosting the performance of Web search engines.

ACM TOIS, 24(1), 51-78, 2006.

Galamboš, L.

Dynamic Inverted Index Maintenance.

IJCS, 1(2), 157-162, 2006.

Gomes, D., Silva, M.J.

The Viuva Negra crawler.

FI-FCUL, TR-2006-06-21, Lisboa, 2006.

Hafri, Y., Djeraba, C.

Dominos: A New Web Crawler's Design.

IWAW, Bath, UK, 2004.


Heydon, A., Najork, M.
Mercator: A scalable, extensible web crawler.
WWW 2(4), 219-229, 1999.


Hirai, J., et al
WebBase: A repository of web pages.
In Proc. of the 9th WWW Conf., Amsterodam, 2000.


Lawrence, S., Giles, C.L.
Accessibility of information on the web.
Intelligence, 11(1): 32-39, 2000.


Lempel, R., Moran, S.
Predictive cahing and prefetching of query results in search engines.
In Proc. of the WWW12, 19-28, ACM, 2003.


Lim, L., et al
Dynamic Maintenance of Web Indexes Using Landmarks.
In Proc. of the 12th WWW Conf., 2003.


Lyman, P., Varian, H.R.
How much information.
http://www.sims.berkeley.edu/how-much-info-2003


Mignet, L., et al
Xyro: The Xyleme Robot Architecture.
In DIWeb, 91-99, 2001.


Najork, M, Heydon, A.
High-performance web crawling.
COMPAQ SRC173, 2001.


Saraiva, P.C., et al
Rank-Preserving Two-Level Caching for Scalable Search Engines.
In Proc. of the SIGIR2001, 51-58, ACM, 2001.


Silvestri, F.
High Performance Issues in Web Search Engines.
Ph.D. Thesis, Univ. Pisa, 2004.


Shkapenyuk, V., Suel, T.
Design and implementation of a high-performance distributed web crawler.
In Proc. of the 18th DE Int Conf., 357-368, San Jose, 2002.


Zobel, J., Moffat, A.
Inverted files for text search engines.
ACM CSUR, 38(2), 6pg, ACM Press, 2006