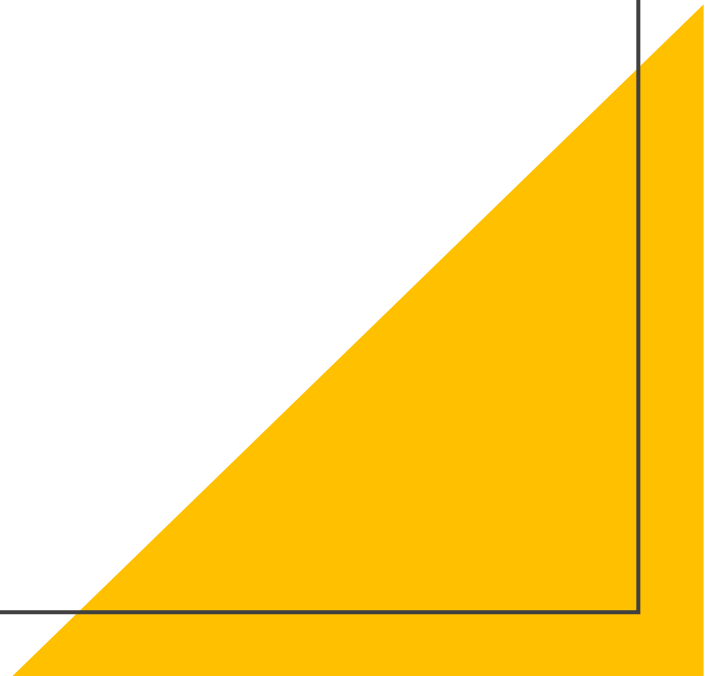


Meranie redundancie v dátach

Školiteľ: Ing. Dušan Bernát, PhD.

Matúš Hluch



Motivácia

- Exponenciálne rastúce množstvo dát a informácií
 - Veľa duplicít
 - Deduplikácia
- Súborové systémy
 - Komprimujúce Btrfs, ZFS...
 - Copy-on-write
 - Deduplikácia - štandardne v blokoch rovnakej veľkosti
 - 512 bajtov, niektoré nastaviteľné

Ciele práce

- Preskúmať vlastnosti redundancie v dátach
 - Deduplikácia s fixnou veľkosťou delenia
 - Podľa iných prác neefektívne
 - Podľa typu dát
 - Dokumenty, obrázky, videá, virtuálne stroje...
 - Podľa veľkosti bloku
 - Vplyv použitej hešovacej funkcie
 - Aj nie kryptograficky bezpečné

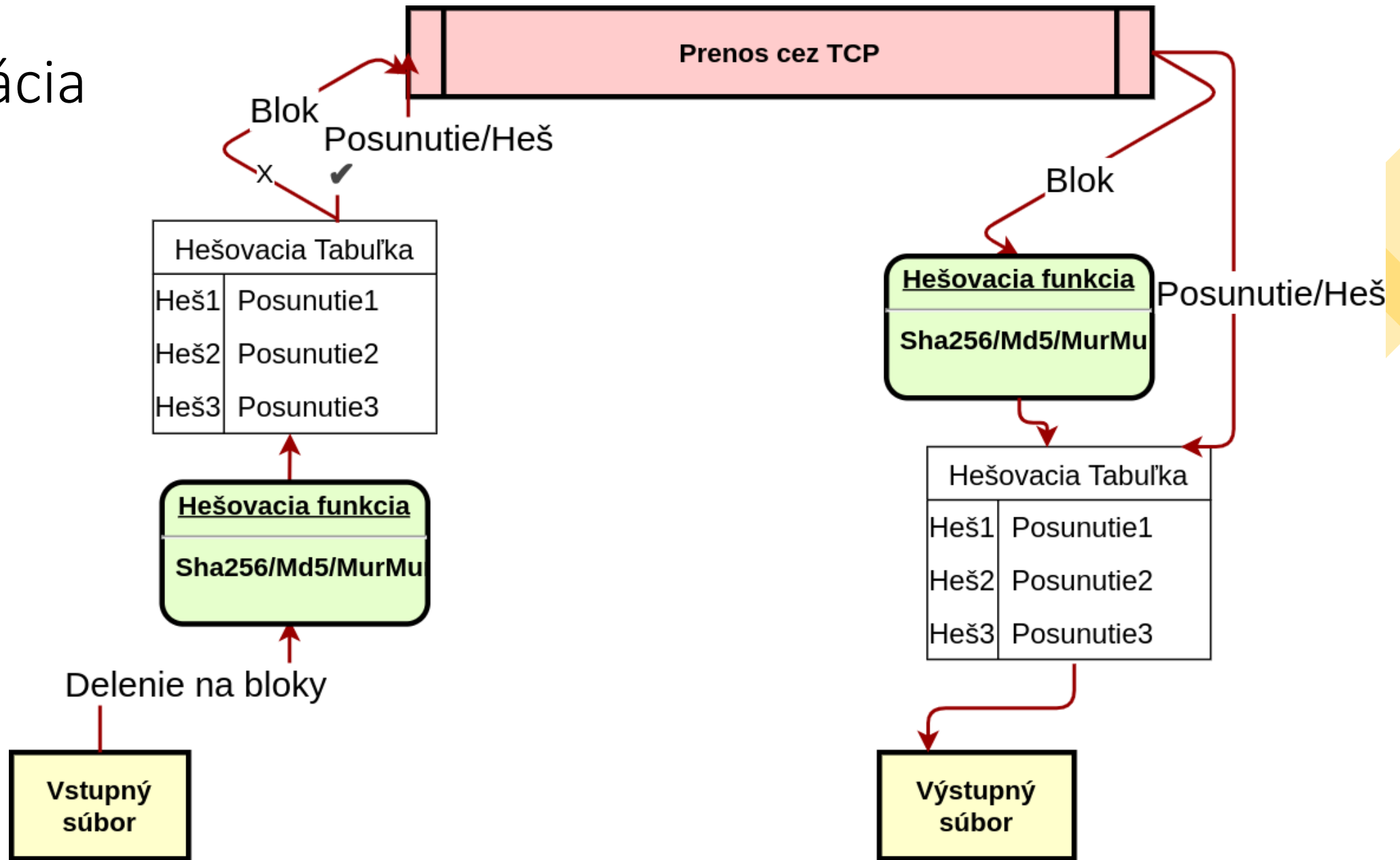
Rozdelenie práce

- Prvá časť
 - Prakticky využiteľný nástroj s deduplikáciou
 - Výstup dát z nástroja
- Druhá časť práce
 - Analyzovanie výstupu dát z nástroja

Náš prístup

- Deduplikácia v súborovom systéme
 - Zložitá – väčšie pracovné tímy
- V našej práci – deduplikácia dát pri prenose cez sieť
 - Rozdelenie súboru na bloky rovnakej (fixnej) veľkosti
 - Hľadanie duplicitných blokov
 - Porovnávanie hešov
 - Blok dát sa pošle iba raz
 - Duplicitné – iba posunutie

Aplikácia



Aplikácia

Send deduplicated data over TCP.

First parameter is window size (number of bytes into which data will be divided).

Second parameter is host.

Third parameter is port.

Fourth parameter is file name, from which data will be sent (applicable only if not using the stdin flag).

Usage:

```
dedupnetgo send window_size host port *file_name [flags]
```

Flags:

-c, --collisionDetection	Detect hash collisions.
-l, --hash_cache_limit uint	Set limit on number of hashCache entries.
-f, --hash_function string	Choose a hash function: sha256/md5/murmur/xxh/simple. (default "sha256")
-h, --help	help for send
-o, --output_seeks	Output seeks from cache to stderr.
-m, --sendHash	Send duplicates hashes instead of seek (if set, blocks will be stored in memory).
-s, --stdin	Send data from stdin.

Deduplikácia dát pri prenose cez sieť

- Aplikácia pre príkazový riadok
 - Implementovaná v Go
 - Knižnica Cobra
 - Kubernetes, GitHub CLI...
- Prenos súborov cez TCP
 - Navrhnutý vlastný komunikačný protokol nad TCP



Aplikácia

- Nastaviteľné parametre
 - Veľkosť bloku
 - Hešovacia funkcia
 - Kolízie, rýchlosť, spotreba pamäte...
 - Detekcia kolízií
 - Vyššia spotreba pamäte
 - Veľkosť pamäte

Komunikačný protokol nad TCP

- Prvý bajt (hlavička) určuje typ prenášaných dát
- 1. Prenosové nastavenia
 - Nastavuje odosielateľ
- 2. Prenos dát
 - a) Blok dát
 - b) Posun
 - c) Posledný blok (môže mať inú veľkosť ako ostatné)
- 3. Koniec prenosu

Analýza prenášaných dát

- Spracovanie výstupu z nástroja
 - Implementované v Pythone
 - Matplotlib, SciPy...
- Analýza viacerých parametrov
 - Redundancia podľa veľkosti okna a typu súboru
 - Vzdialenosti medzi duplikátmi
 - Početnosti duplikátov
 - A ďalšie

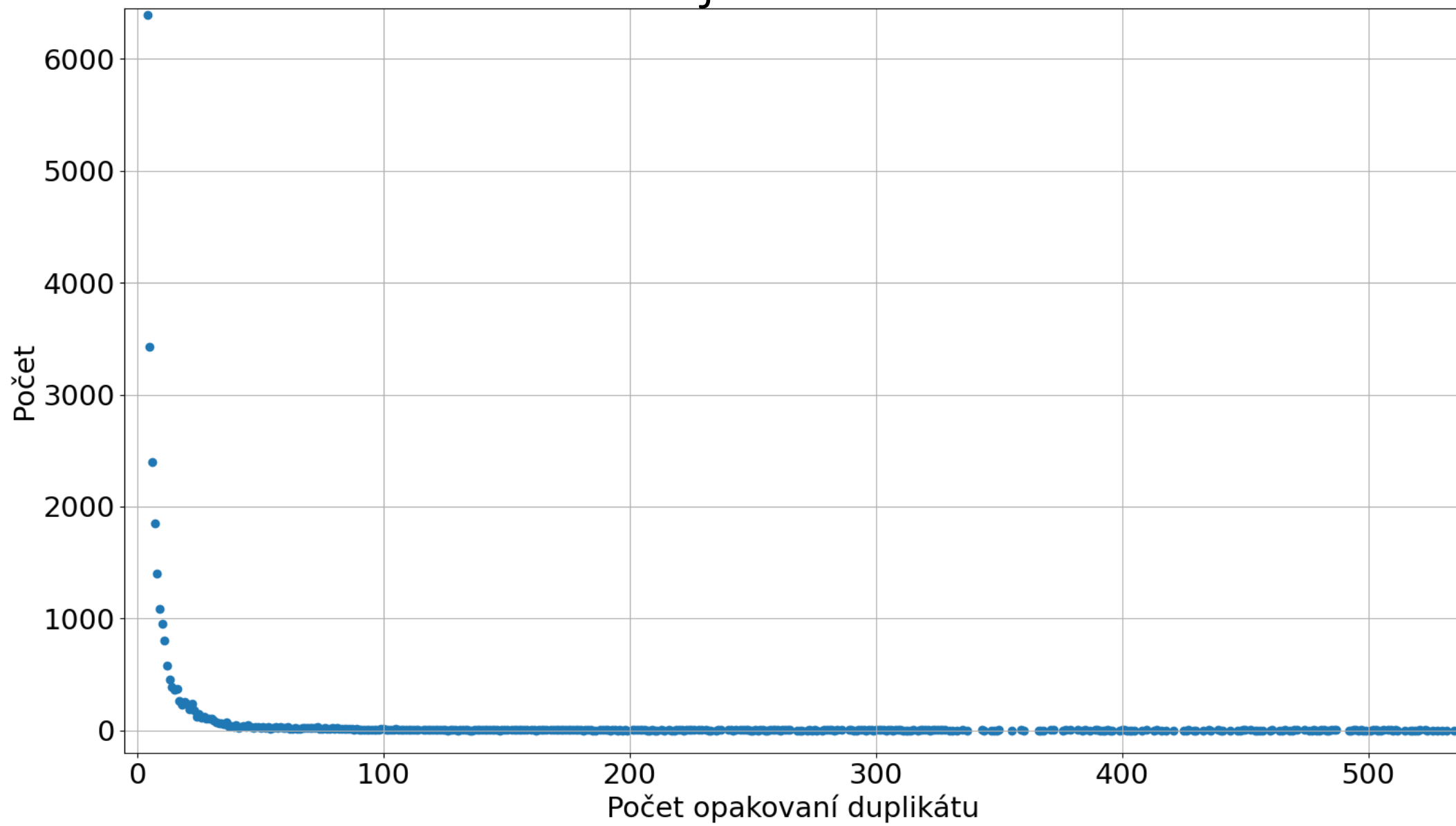
Vstupné dáta

- Rôzne súborové typy
 - Väčšina dátových množín rovnakého typu
 - Určené na strojové učenie, ...
- Govdocs1
 - Milión súborov rôznych typov, voľne šíriteľné
 - Súbory z .gov domén
 - pdf, jpg, txt, html, ppt...
- Využívame
 - Niekoľko GB dát - rozdelených podľa typu
 - Okolo 10 000 súborov

Vstupné dáta

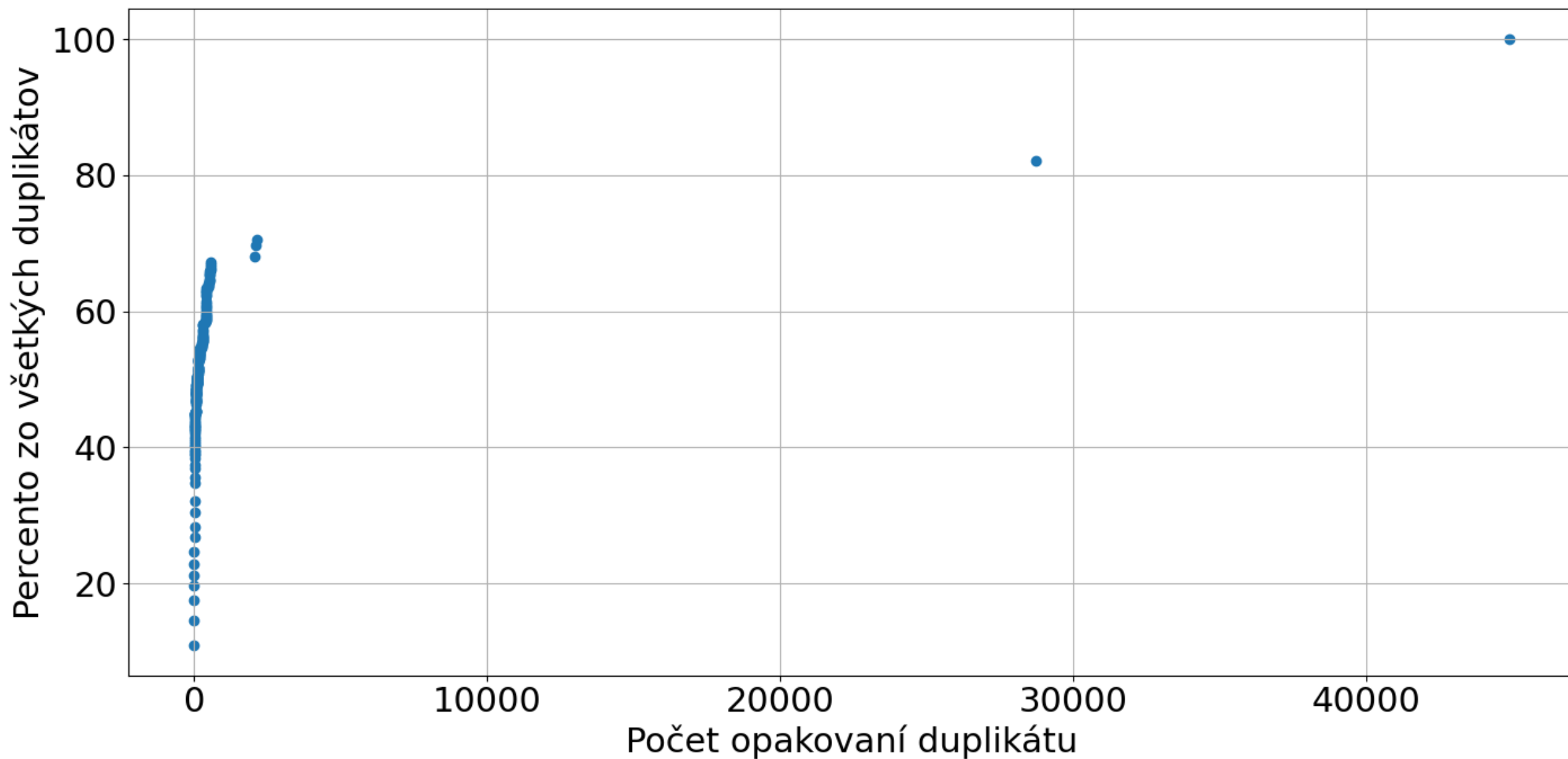
- Virtuálne stroje
 - Linuxové distribúcie
- Virtuálne disky
 - Po inštalácii
 - Základný softvér (prehliadač, úprava dokumentov...)

TXT - veľkosť bloku 32 bajtov

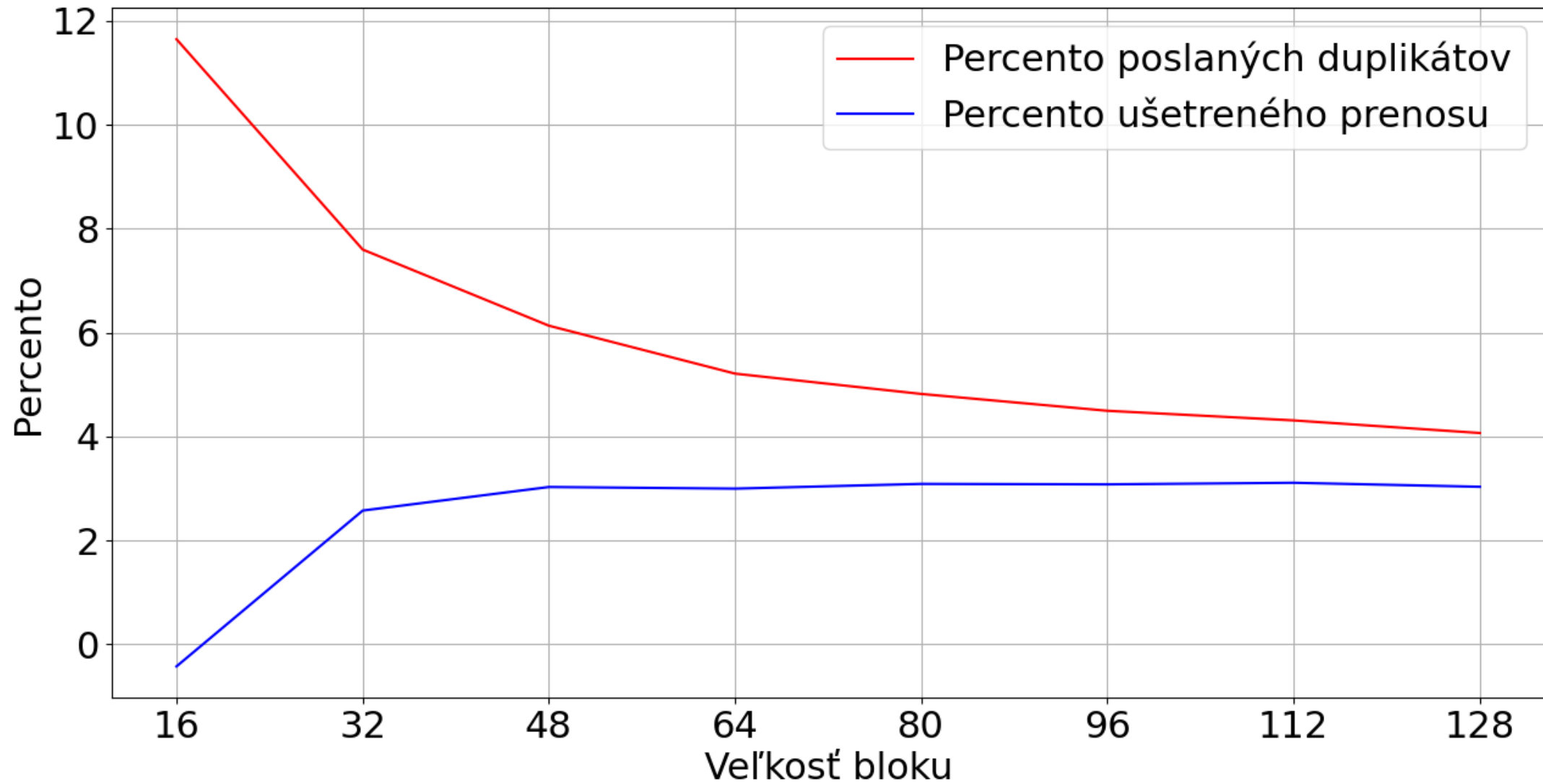


Meranie redundancie v dátach

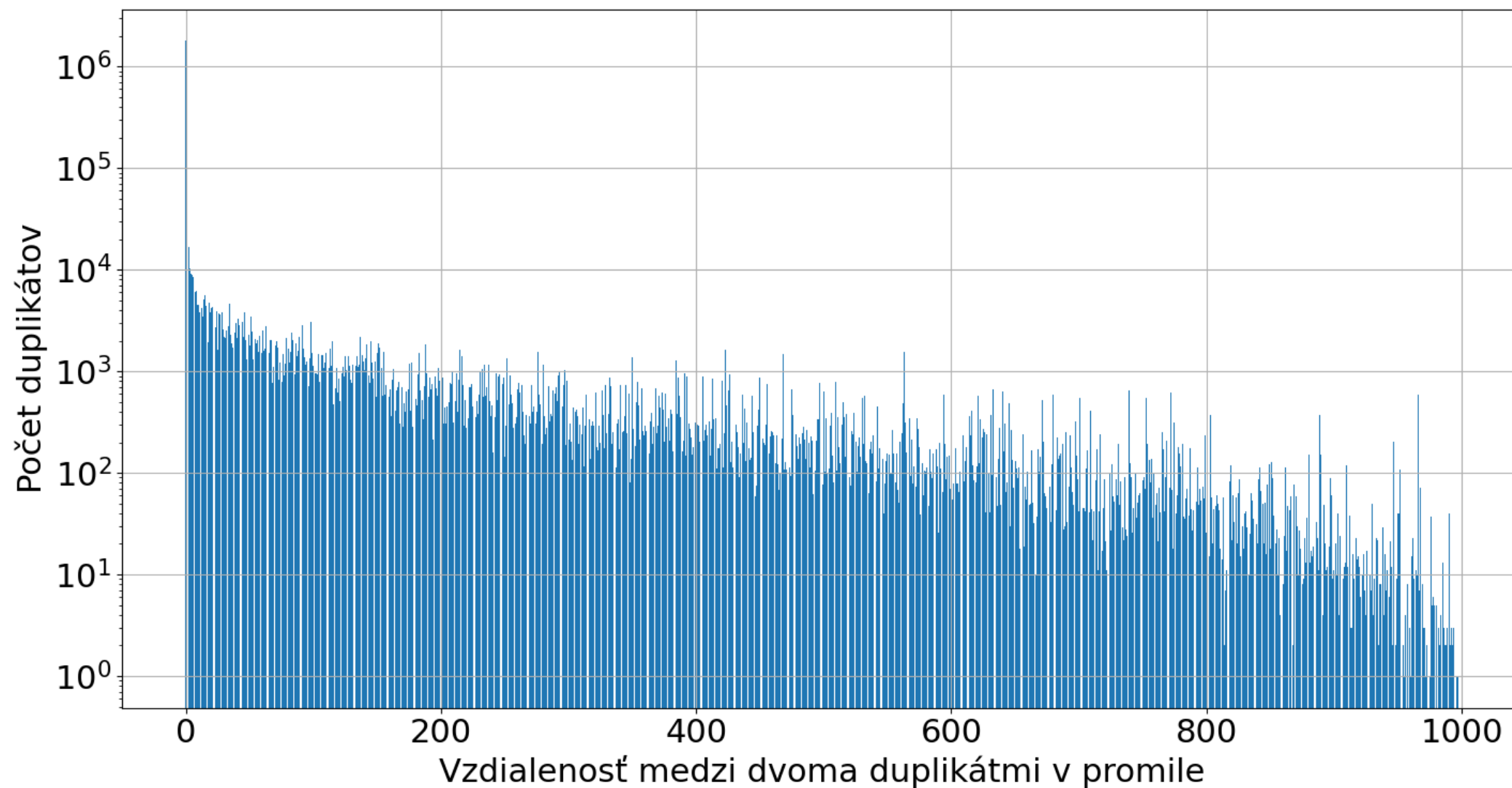
JPG – veľkosť okna 32 bajtov



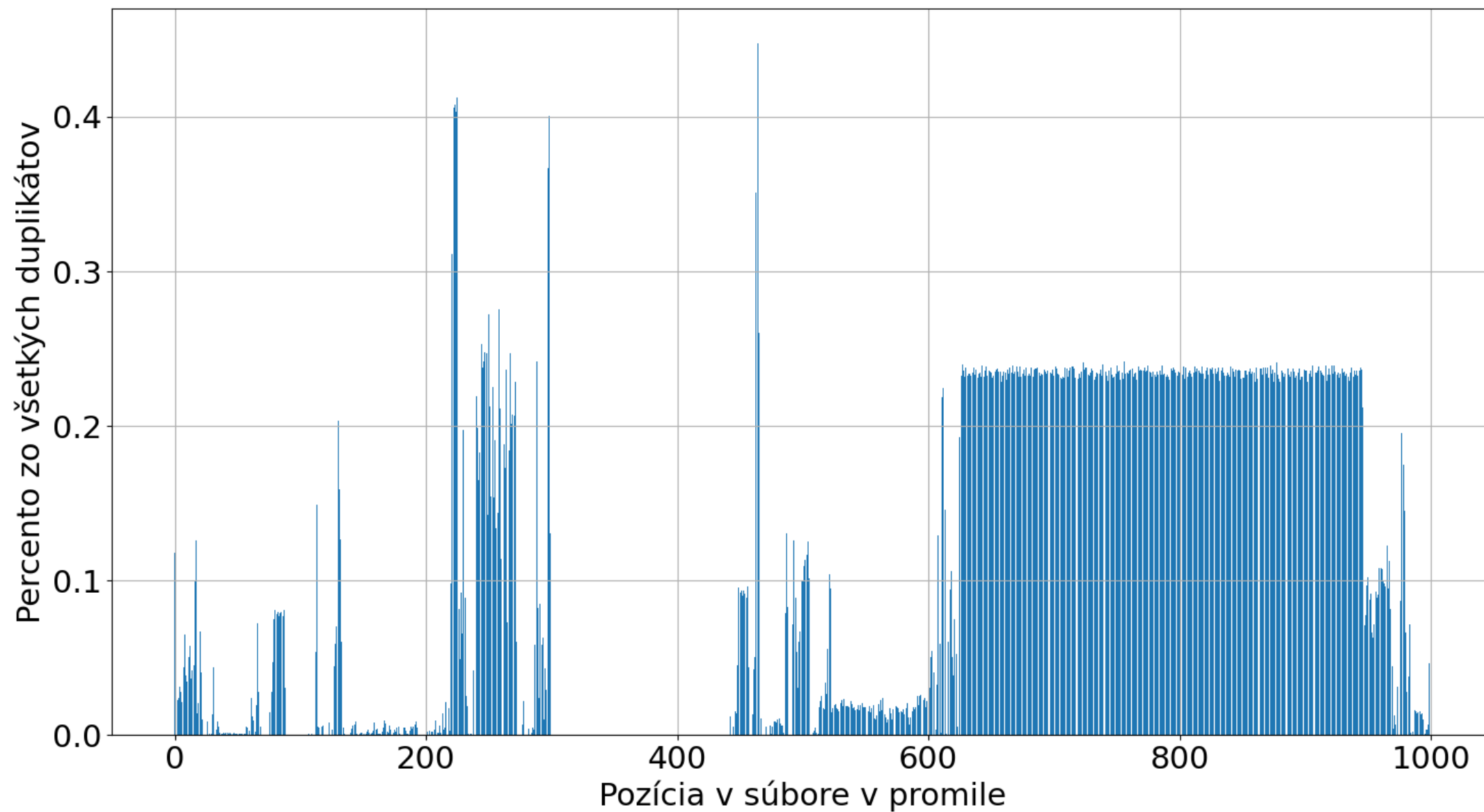
PDF



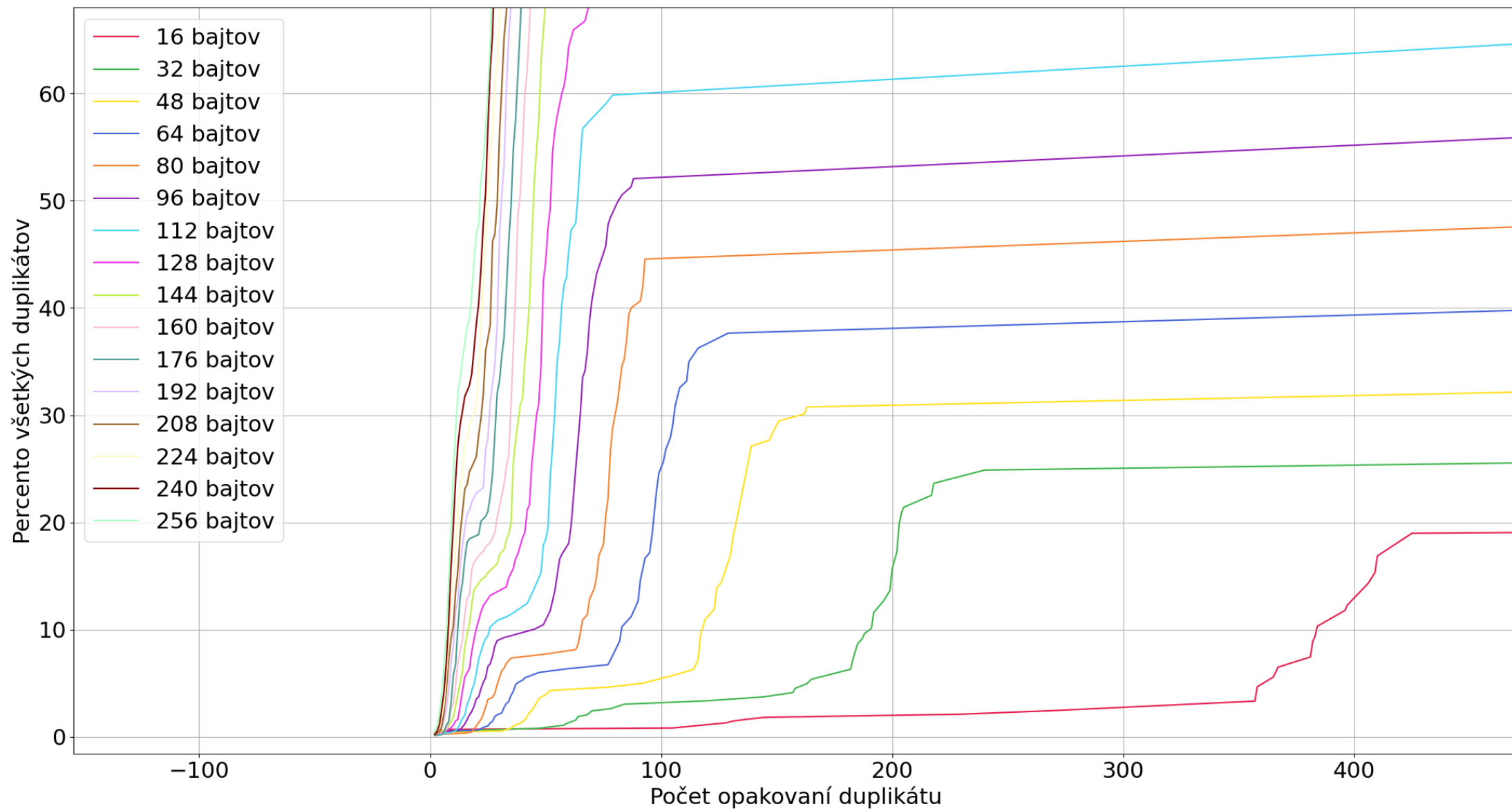
HTML – veľkosť okna 32 bajtov



TXT – veľkosť okna 32 bajtov



AVI



	16	32	48	64	80	96	112	128	256	512
avi	0,46%	0,44%	0,43%	0,42%	0,4%	0,39%	0,38%	0,36%	0,25%	0,1%
doc	27,43%	22,3%	18,9%	17,92%	15,06%	14,65%	13,58%	13,98%	10,53%	6,79%
html	41,76%	24,97%	17,79%	13,49%	10,76%	8,90%	6,35%	6,49%	3,07%	2,02%
jpg	2,36%	1,97%	1,75%	1,55%	1,45%	1,4%	1,23%	1,26%	0,93%	0,7%
mov	0,02%	0,02%	0,02%	0,02%	0,02%	0,01%	0,01%	0,01%	0,01%	0%
mp4	0,53%	0,52%	0,51%	0,49%	0,48%	0,47%	0,45%	0,44%	0,32%	0,18%
pdf	11,65%	7,6%	6,13%	5,21%	4,82%	4,49%	4,31%	4,07%	3,35%	2,75%
ppt	10,9%	7,6%	5,96%	5,04%	4,24%	4,09%	3,49%	3,45%	2,57%	2,02%
ps	41,58%	27,99%	23,34%	20,32%	18,36%	17,49%	16,18%	15,52%	11,66%	7,7%
txt	33,16%	19,87%	15%	12,91%	11,31%	9,88%	8,56%	7,3%	2,15%	0,27%
xls	19,37%	12,5%	10,12%	8,92%	7,63%	6,96%	6,07%	5,76%	1,93%	1,29%

Virtuálne stroje

- Po minimalizácii bolo 30% dát tvorených nulovými bajtami
- Debugfs
 - Ncheck, ickcheck
 - Posunutie -> názov súboru
 - Nulové bajty
 - Nepriradené bloky
 - Bajt 255
 - Nepriradené bloky a ovládače zariadení (mbn)
 - Tilda
 - Mikrokód (ucode)

Možné rozšírenia práce

- Využiť dáta na vylepšenia
 - Nástroja Dedupnetgo
 - Súborových systémov
 - Komprimačných algoritmov
- Preskúmať ďalšie vlastnosti duplikátov
- Analyzovať premenlivú veľkosť bloku

Výsledky

- Duplikáty
 - Najviac redundancie v nekomprimovaných dátach
 - Html a txt súbory
 - Početnosti duplikátov – exponenciálne rozdelenie
- Veľkosť bloku
 - Čo najmenšia
 - Problém – režijné náklady
- Vzdialenosti medzi duplikátmi
 - Väčšina do 1 promile súboru

Porovnanie s inými riešeniami

- Rsync prenáša rozdiely medzi rôznymi verziami dát
 - Dedupnetgo staršiu verziu dát ignoruje
 - Nové dáta prenáša celé (alebo komprimované)
 - Dedupnetgo nové dáta priebežne deduplikuje
- ZFS a BTRFS (Duperemove)
 - Oproti Dedupnetgo neprenášajú dáta cez sieť
 - Deduplikovanie po väčších blokoch
 - 4kB a viac

Najmenší blok

- Veľkosť prenesených dát
 - Nastavenia prenosu 10 bajtov
 - **Nový blok dát – prenesené celé + 1 bajt**
 - **Duplikát 8 + 1 bajtov**
 - Posledný blok dát – prenesené celé + 1 + 8 bajtov
- Prenos môže byť spravidla efektívny pri bloku väčšom ako 10 bajtov