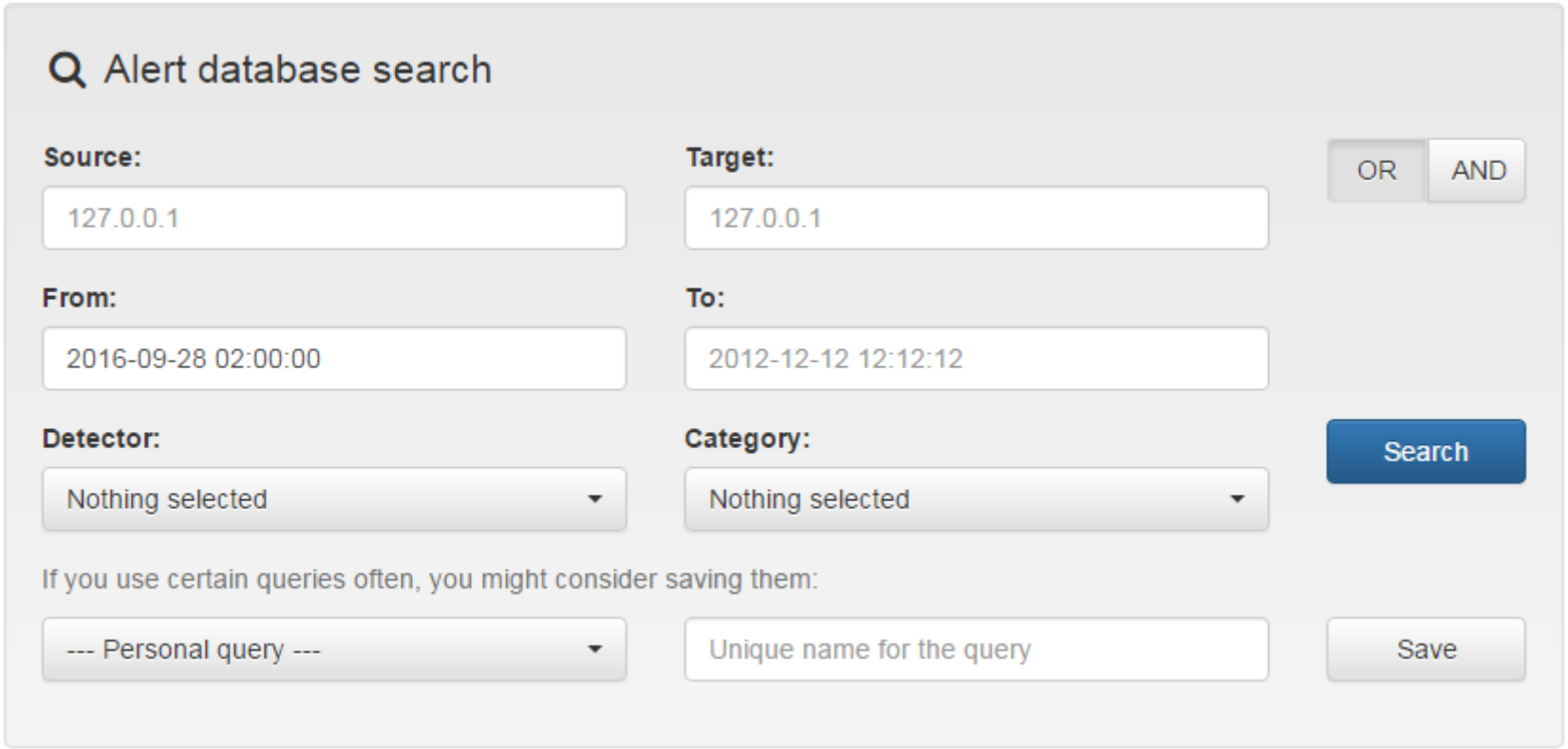
1996–2016
CESNET

Výkon databázového subsystému

12.10.2016 Radko Krkoš

Mentat

- Modulárny distribuovaný SIEM (Security Information and Event Management)
- Príjem, ukladanie, analýza, spracovanie a reakcie na *veľké množstvo* bezpečnostných udalostí



Mentat – DB subsystém

- IDEA → JSON → NoSQL → MongoDB
- Mentat-HUB:
  - MongoDB 3.2.10,
  - WiredTiger (Snappy),
  - ~70M udalostí,
  - Predtým MongoDB 2.6.x, MMAPv1.
- Premenlivá doba odpovede na dotaz bez zjavnej príčiny.

Projekt merania výkonu DB

- Ciele:
  - Modelovanie typickej záťaže databázy systémom Mentat,
  - Zistenie výkonových charakteristík rôznych DB systémov pri takejto záťaži,
  - **Výber vhodnejšej databázy?**
  - Sekundárne: drobné vylepšenia aktuálne nasadeného systému.

Testovacie scenáre

1. Zápis dát a indexácia
   1.1. Vplyv fragmentácie na zápis dát
   1.2. Vplyv fragmentácie na zápis dát (s indexom)
   1.3. Doba zápisu a indexácie
   1.4. Doba zápisu pri vytvorených indexoch
2. Kompletné čítanie
   2.1. Studené čítanie (COLD)
   2.2. Teplé čítanie (HOT)
   2.3. Meranie vplyvu COUNT()
   2.4. Meranie vplyvu opačného radenia
   2.5. Škálovanie v závislosti na veľkosti zoznamu IP adries
3. Dávkové čítanie
   3.1. Dávkové studené čítanie poslednej stránky pomocou SKIP() + LIMIT()
   3.2. Dávkové studené čítanie poslednej stránky v opačnom radení pomocou SKIP() + LIMIT()
   3.3. Dávkové teplé čítanie pomocou SKIP() + LIMIT() od začiatku
   3.4. Dávkové teplé čítanie pomocou SKIP() + LIMIT() od konca

Testovacie dotazy

Označenie	Filter*
MENTAT	Udalosti za obdobie
HAWAT1	Udalosti kde figurovala daná IP adresa
HAWAT2	Udalosti z rozsahu adries <IP_{min},IP_{max}>
HAWAT3	Udalosti zo zadanej kategórie
HAWAT4	Udalosti získané od daného detektoru
HAWAT5	Udalosti získané od daného detektoru a spadajúce do danej kategórie
HAWAT6	Udalosti kde figurovala daná IP adresa a spadajúce do danej kategórie
HAWAT7	Udalosti z rozsahu adries <Ip_{min},Ip_{max}> a spadajúce do danej kategórie
OTHER1	Udalosti kde figurovala IP adresa zo zoznamu
OTHER2	Udalosti kde figurovala daná IP adresa a bol použitý zadaný port transportnej vrstvy

* Všetky dotazy sú časovo ohraničené a výsledky sú okrem OTHER1 zoradené chronologicky

Testovací nástroj

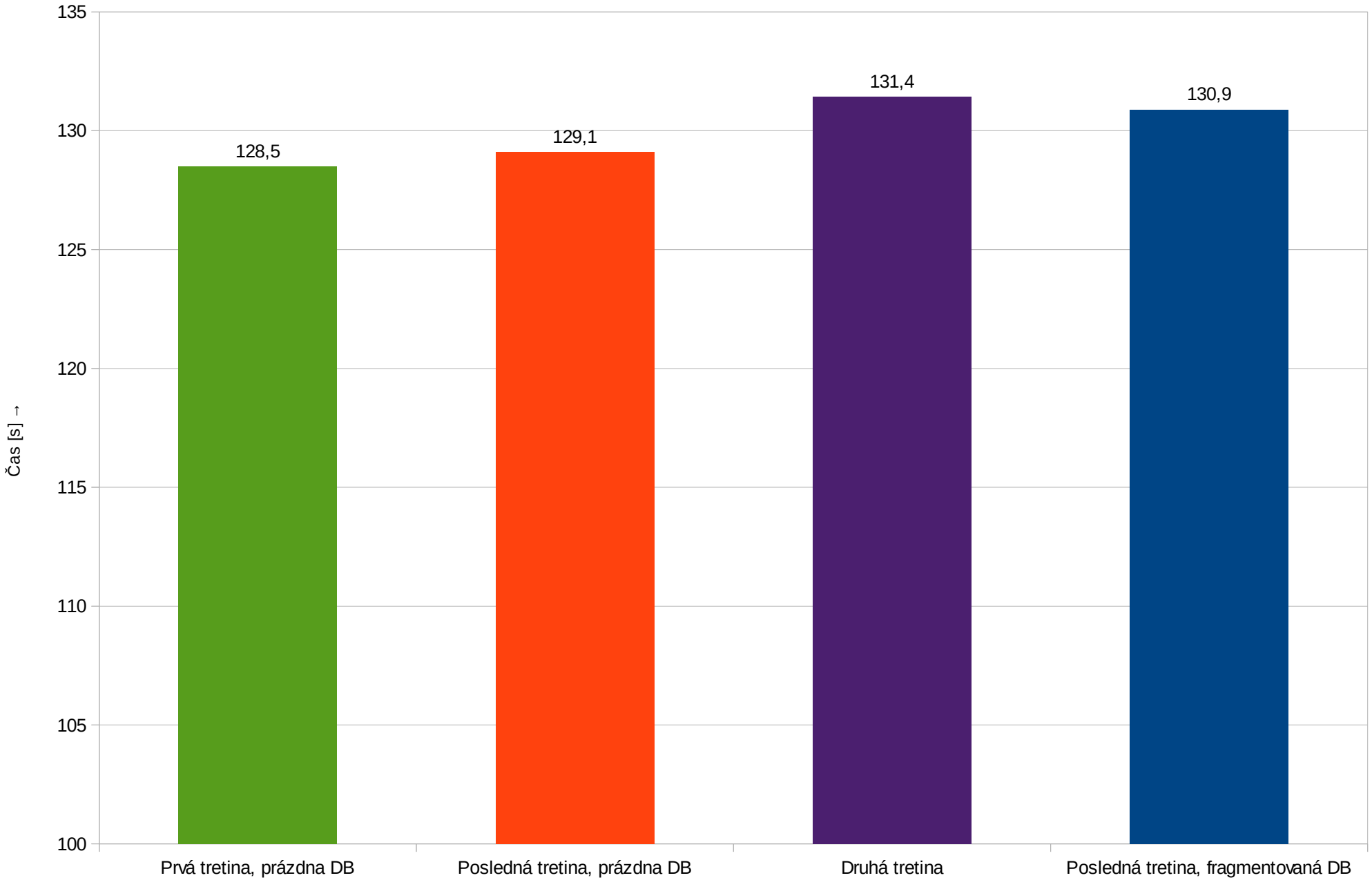
- Python3,
- Spoločný testplán pre všetky DB systémy,
- Jednoduché pridanie nového typu DB,
- Žiadne vynútené použitie indexu (hint).

```
def ts2_1(self):
    """TS2.1: Cold read"""
    print(self.ts2_1.__doc__)
    self.dbinst.init_ts2_1()
    params = {"start": "2016-06-06T12:00:00", "end": "2016-06-12T12:00:00", "ip": "195.113.252.33", "net_min": "208.100.26.0",
              "net_max": "208.100.26.255", "category": "Recon.Scanning", "node_name": "cz.cesnet.hoststats"}
    queries = self.dbinst.prepare_queries(("MENTAT", "HAWAT"), params)
    for query in queries:
        for run in range(1, self.repeat):
            self.dbinst.coldstart()
            self.dbinst.run_simple_query(query, run)
```

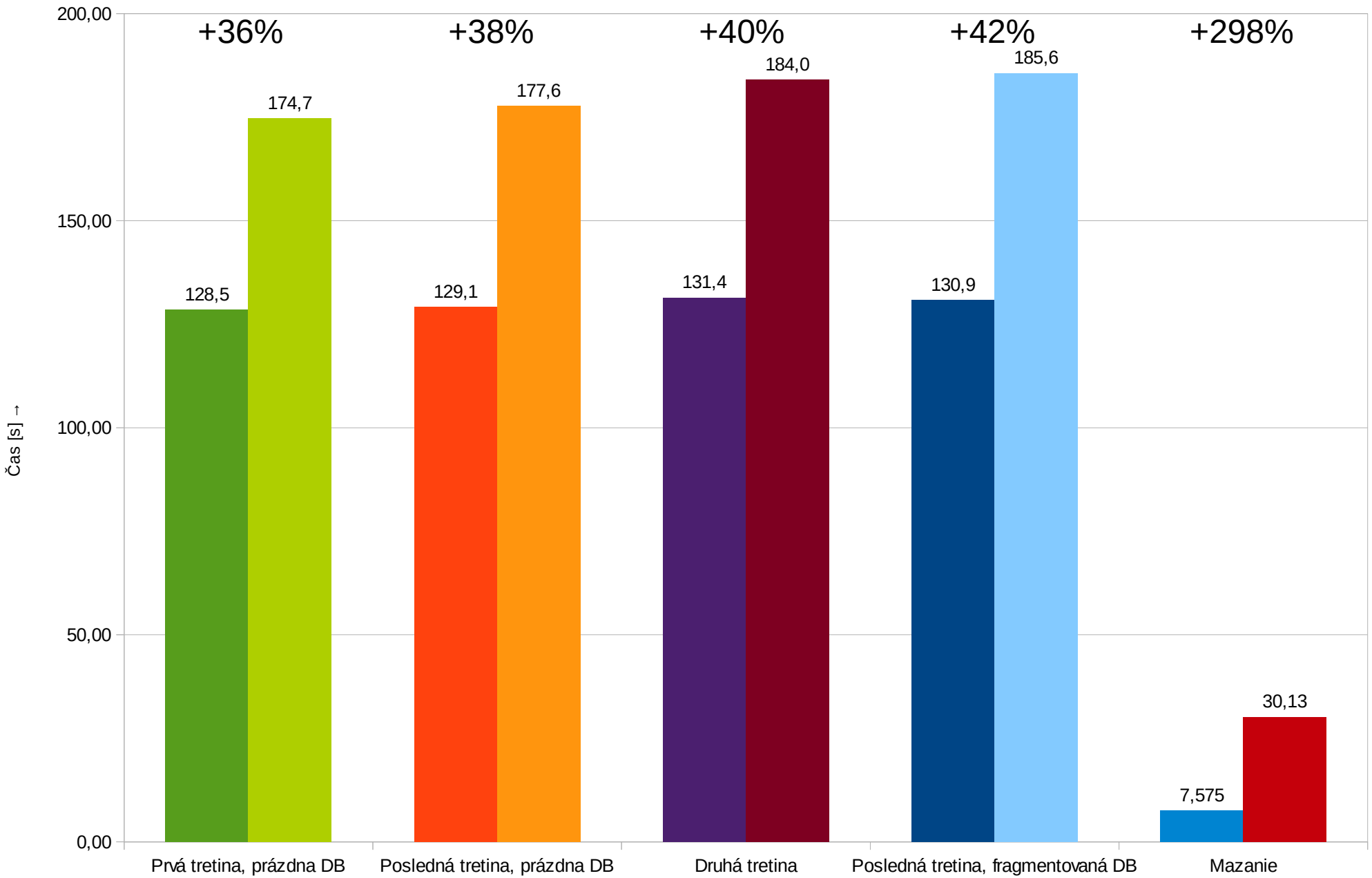
Realizácia merania

- Testovací systém:
  - warden-dev.cesnet.cz
  - 2x AMD Opteron 4184 (6C@2,8GHz),
  - 32GB RAM (NUMA),
  - MongoDB 3.2.9.
- Testovacie dáta:
  - Zápis: 1 deň, 1 149 054 záznamov,
  - Čítanie: 1 týždeň, 8 043 378 záznamov.
- Priemer z 9 meraní.

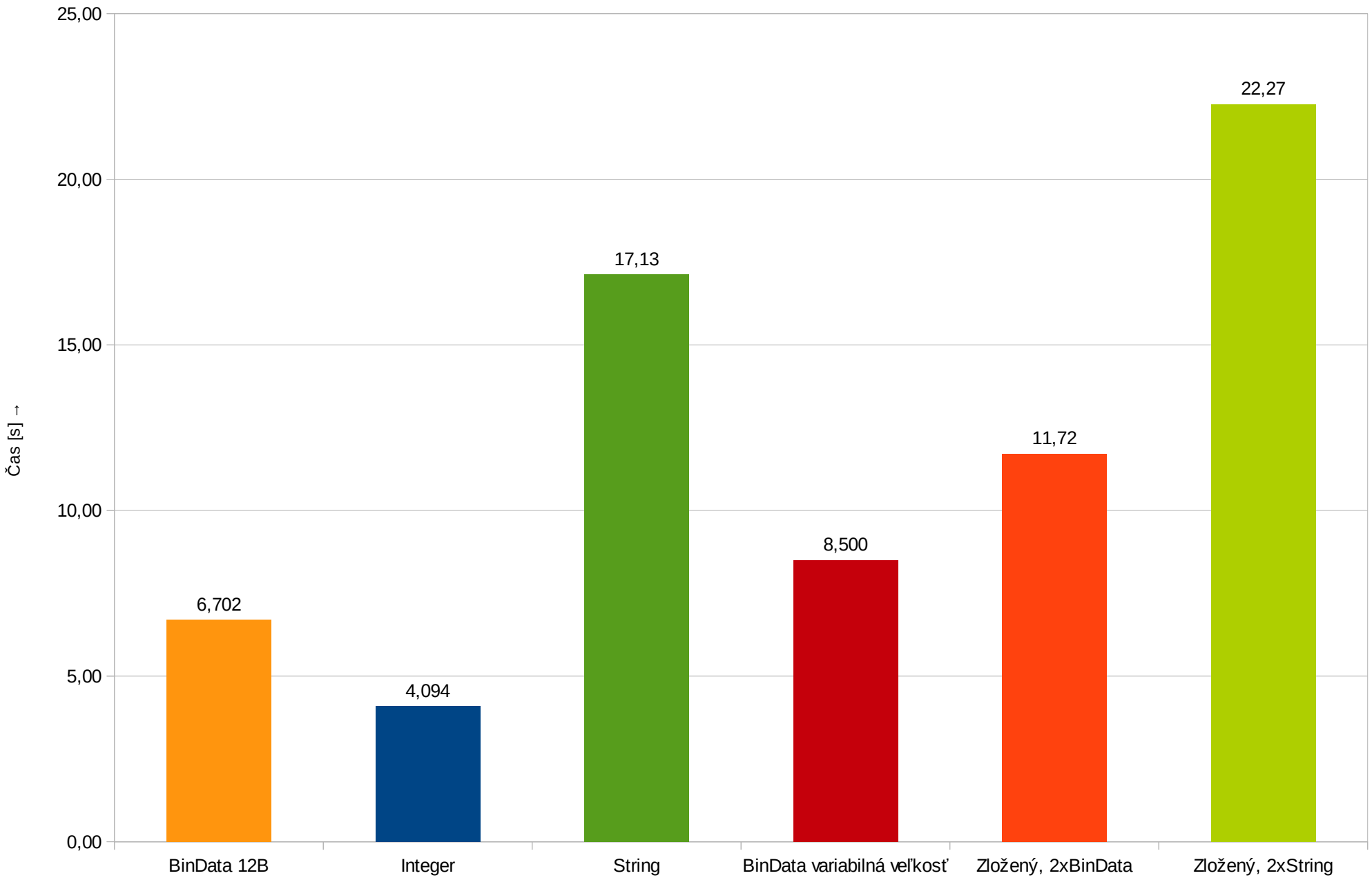
TS1.1: Vplyv fragmentácie na zápis dát



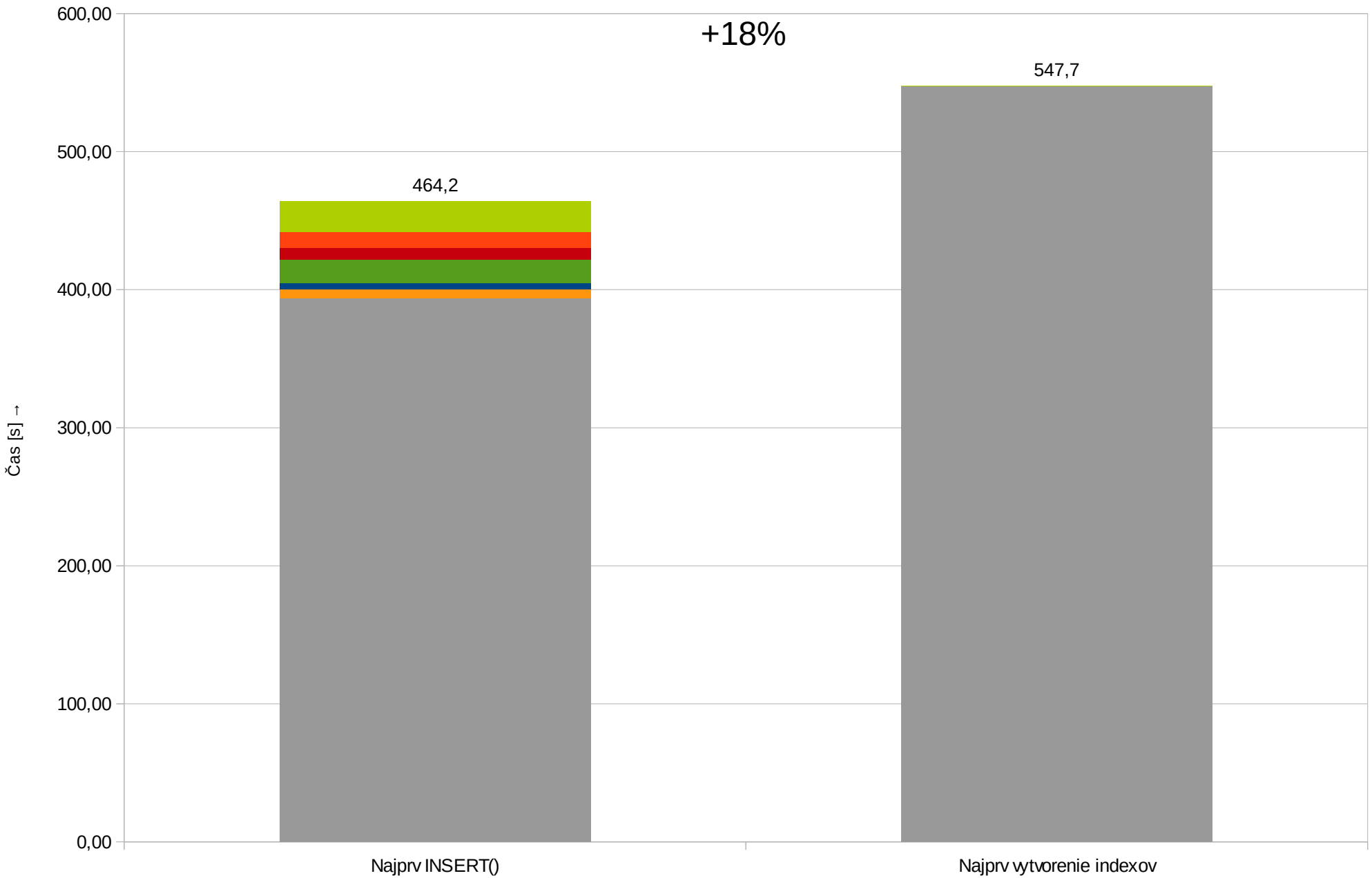
TS1.2: Vplyv indexácie na zápis dát



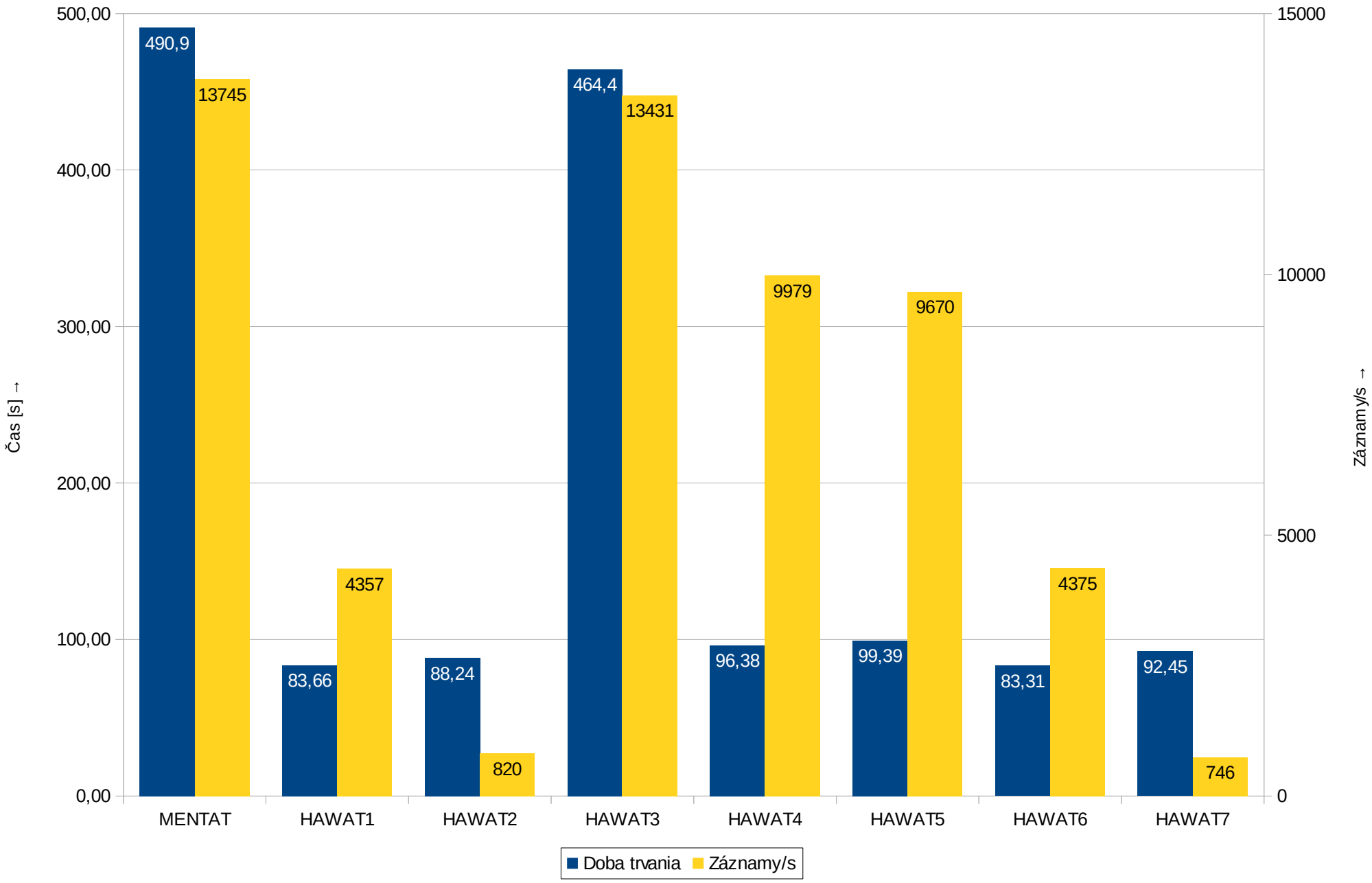
TS1.3: Doba vytvorenia indexov



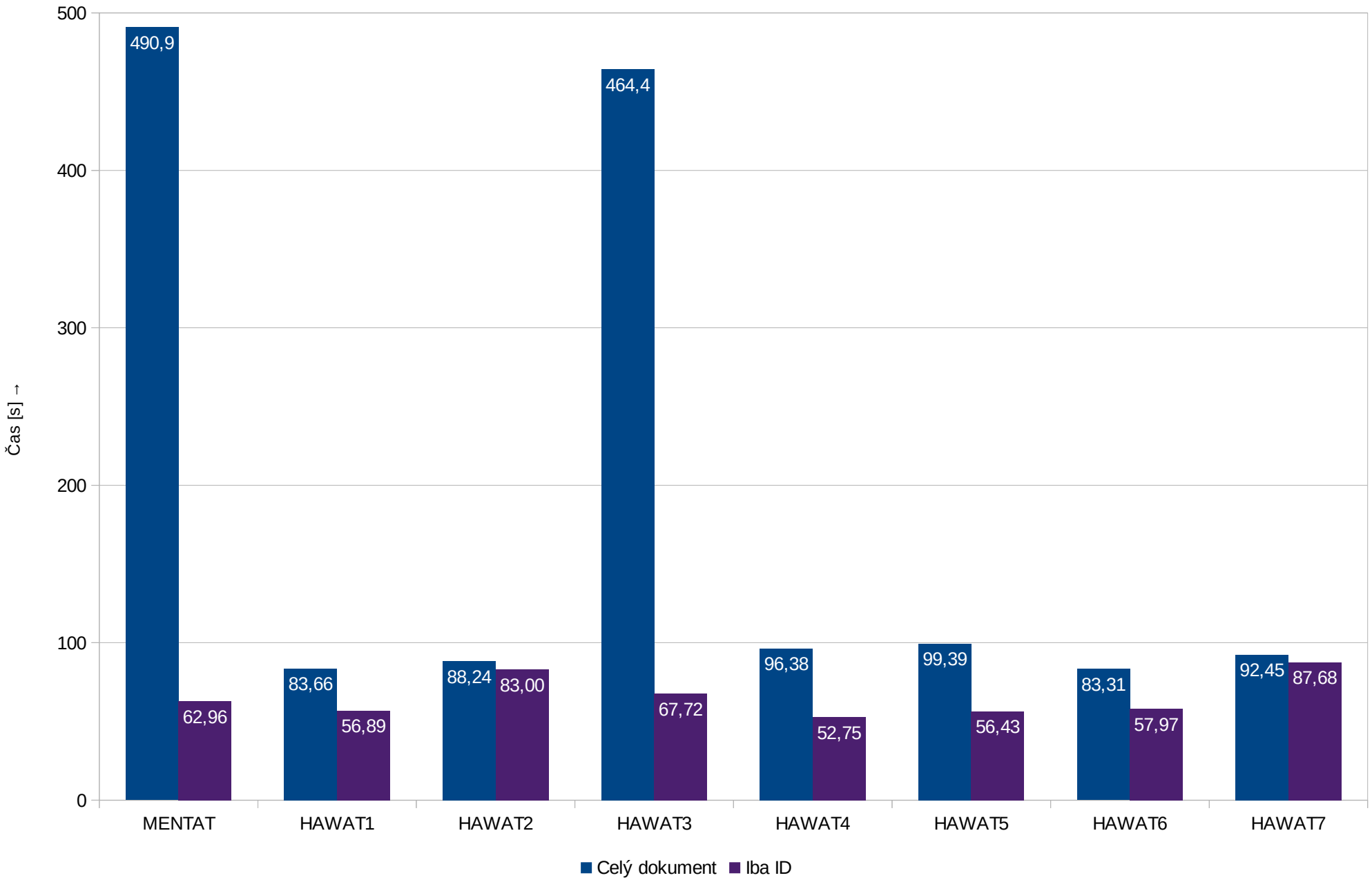
TS1.4: Doba zápisu dát - indexy



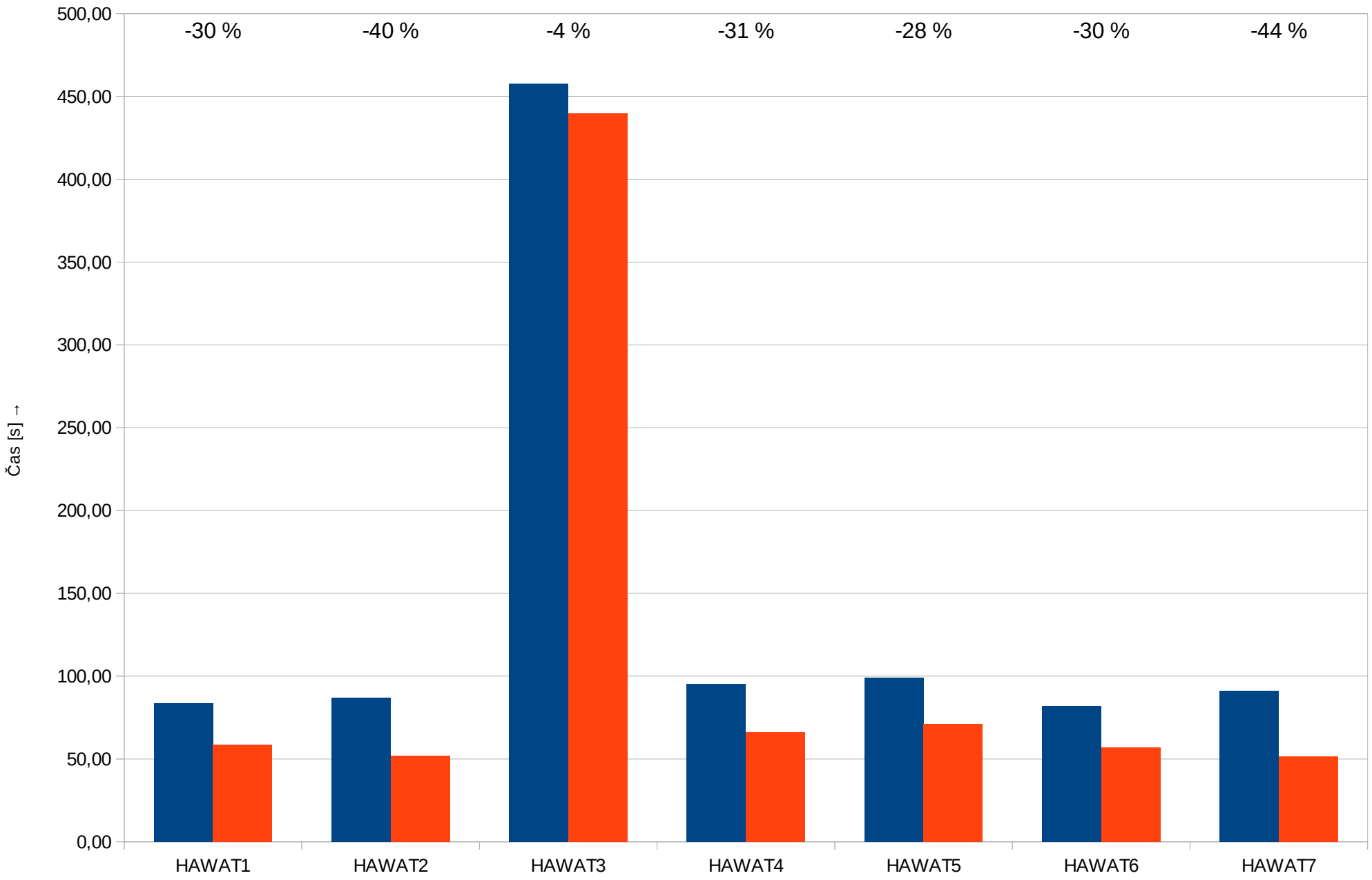
TS2.1: Čítanie za studena



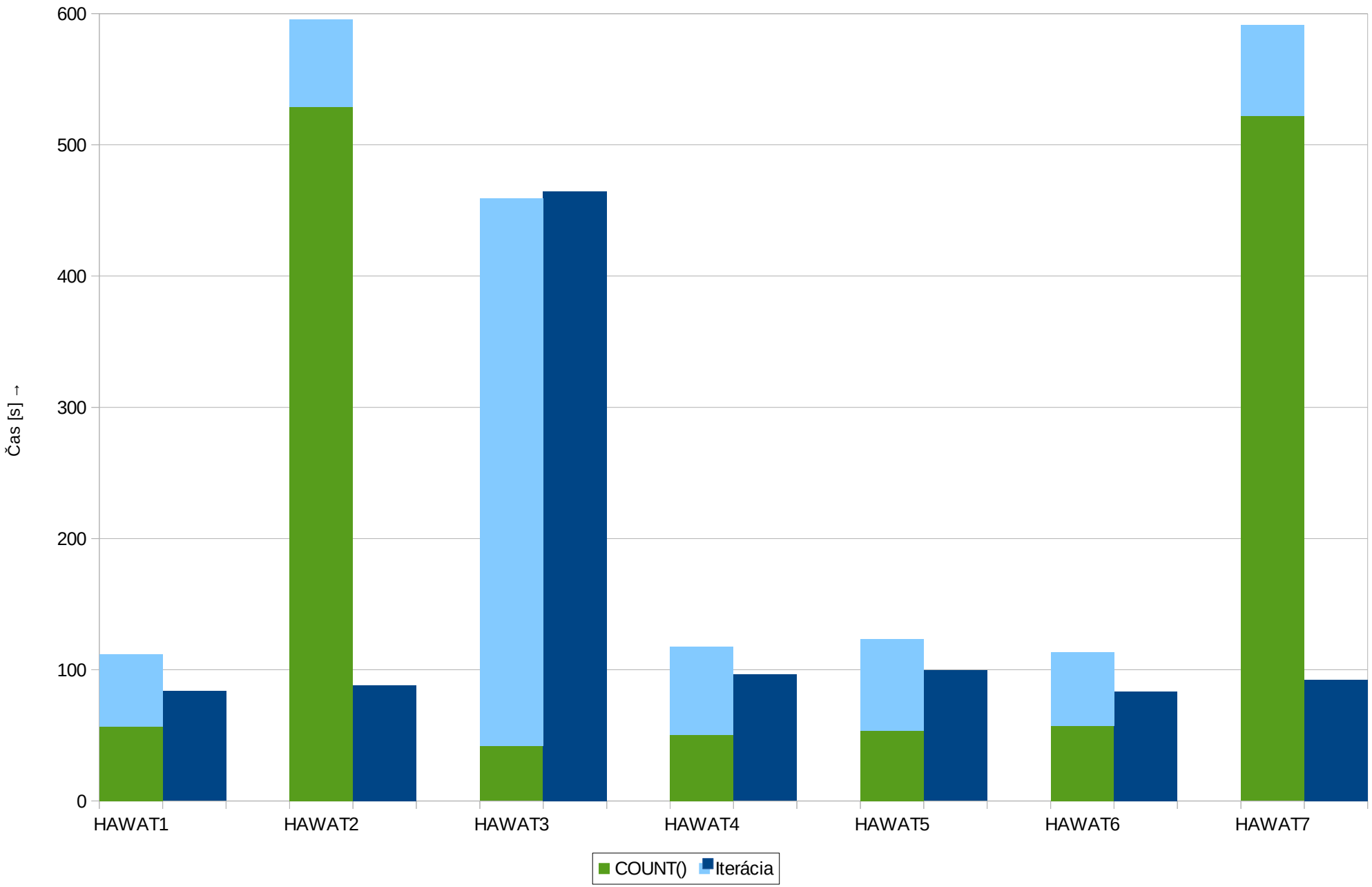
TS2.1: Vplyv veľkosti dokumentu



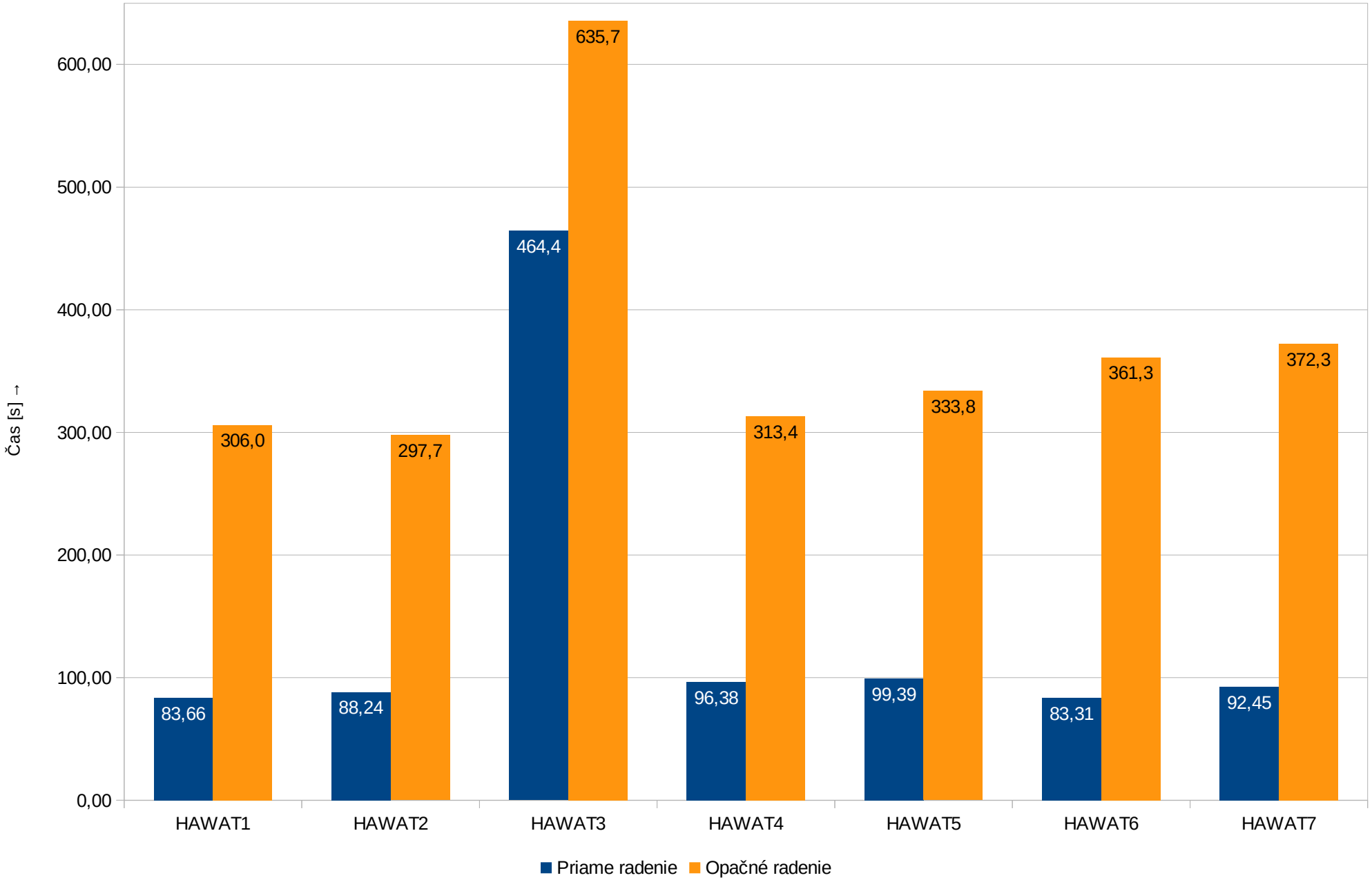
TS2.2: Čítanie za tepla



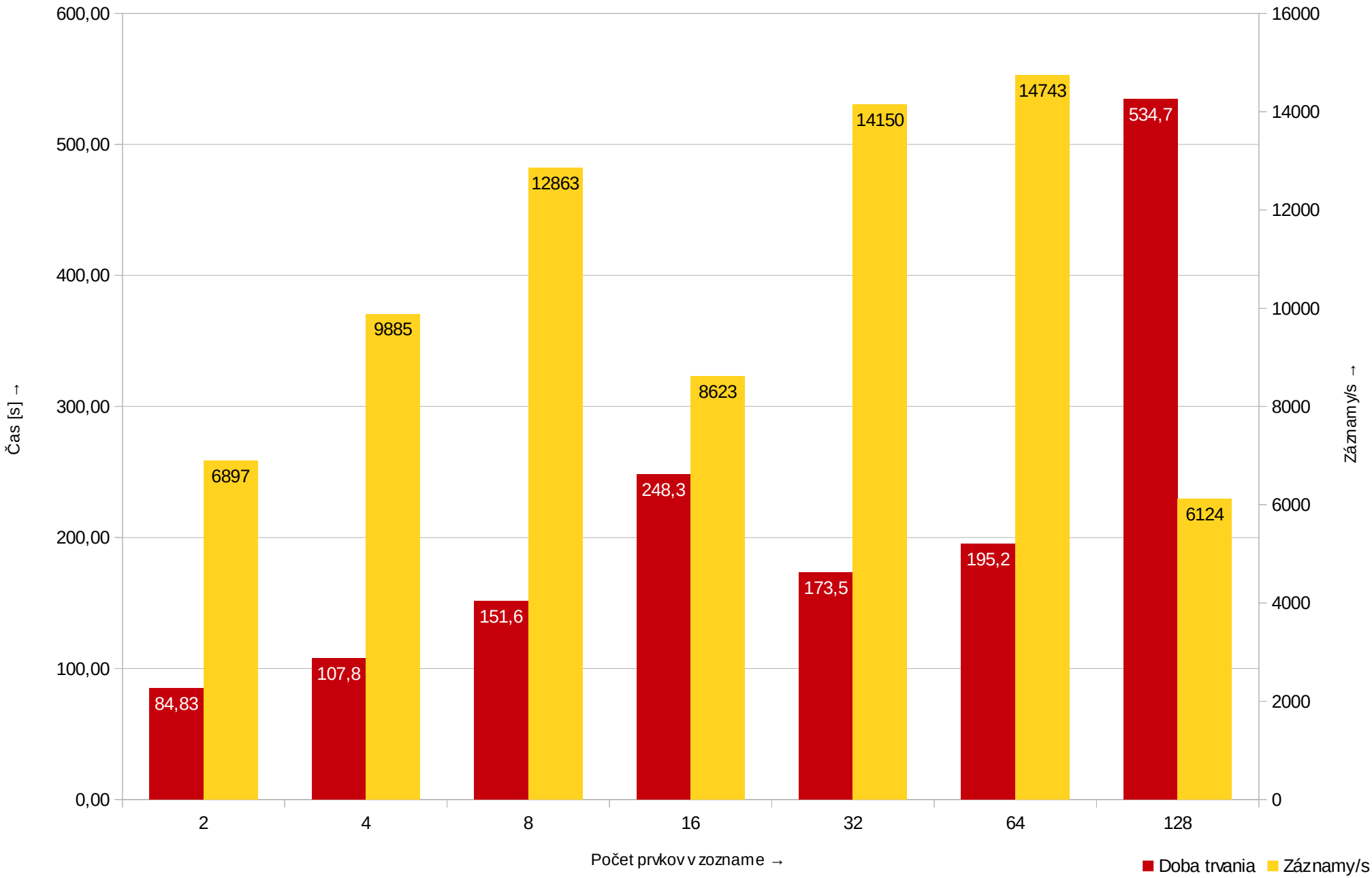
TS2.3: Vplyv COUNT()



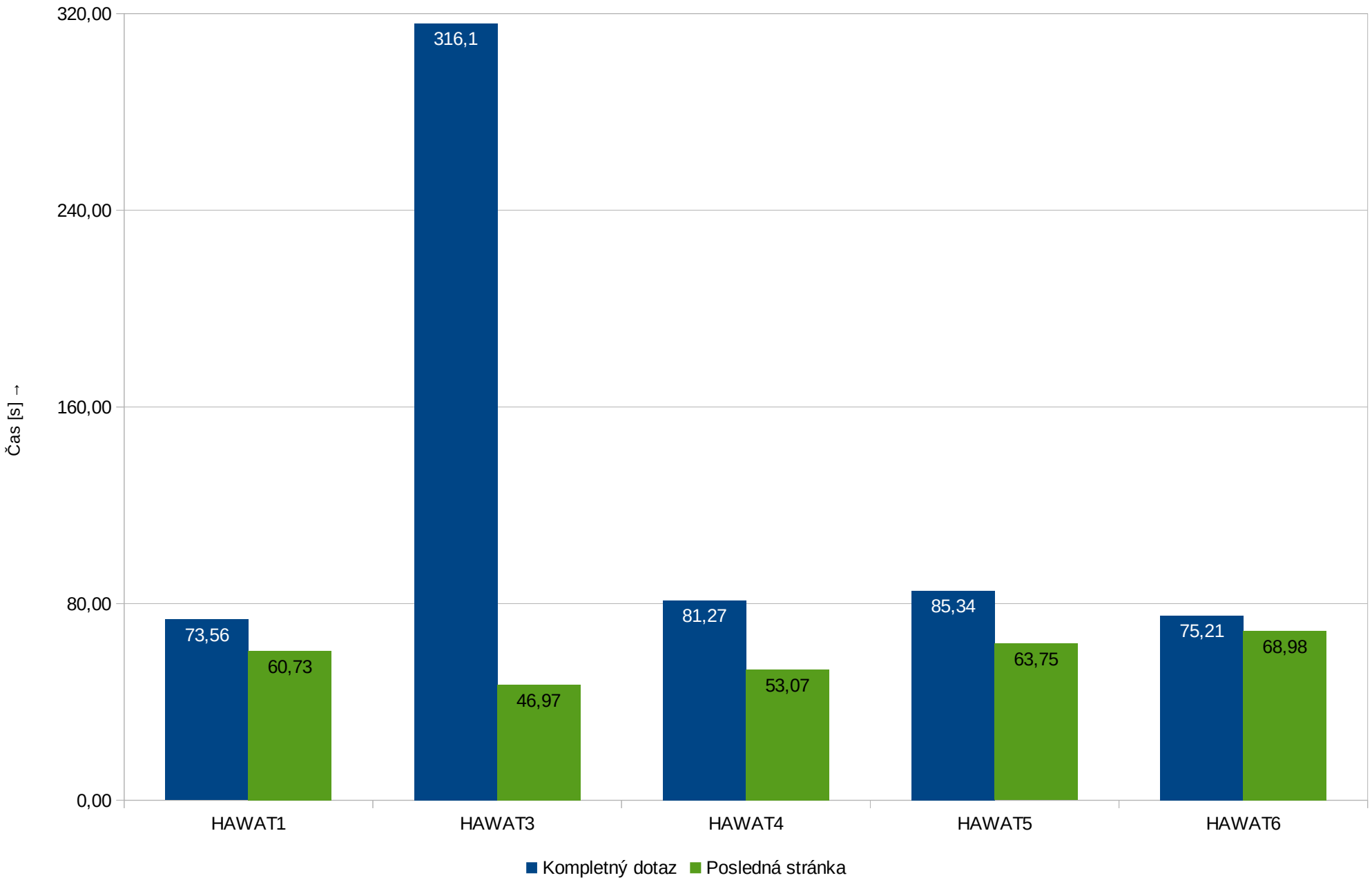
TS2.4: Vplyv opačného radenia



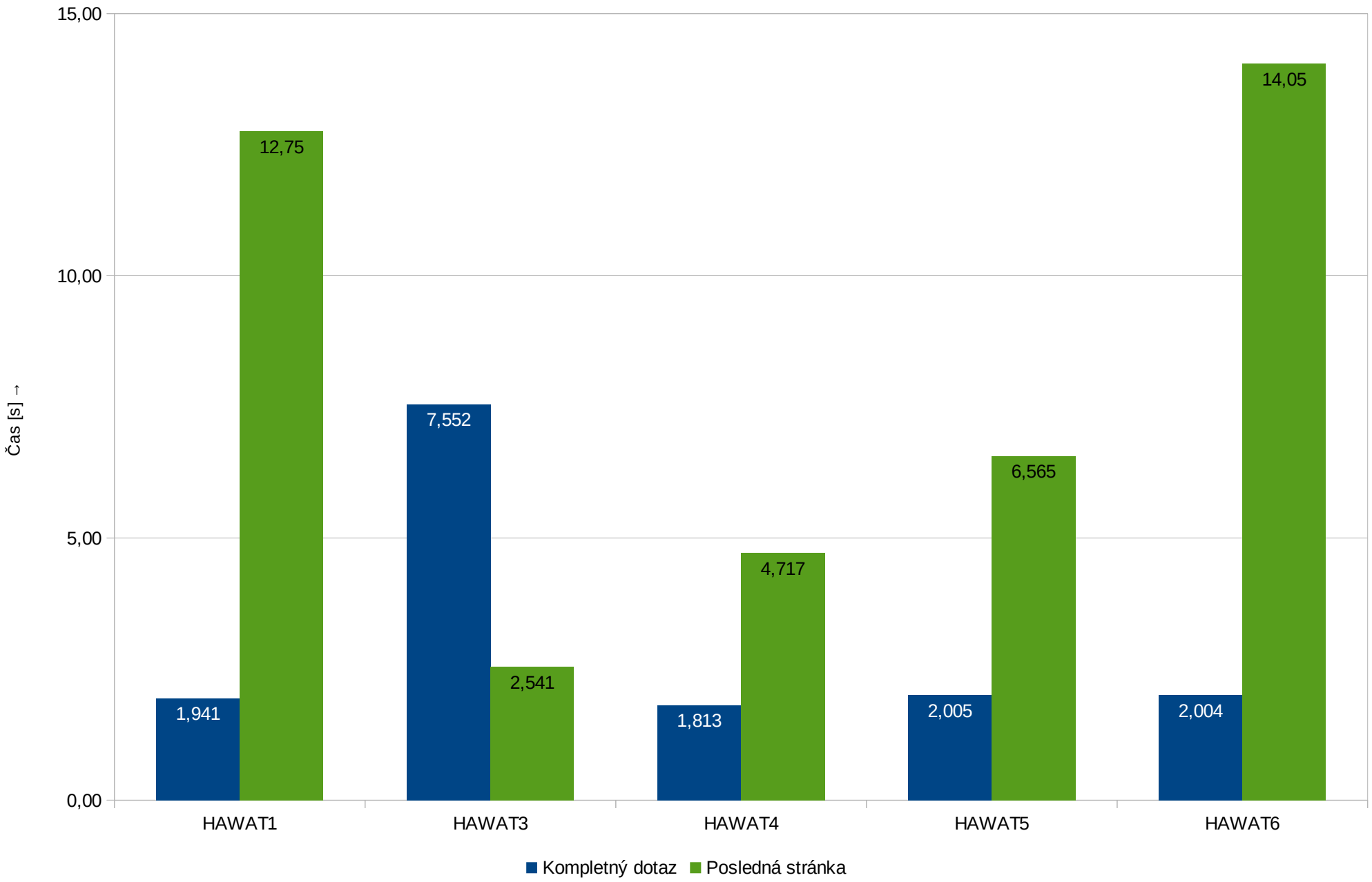
TS2.5: Škálovanie v závislosti na veľkosti zoznamu



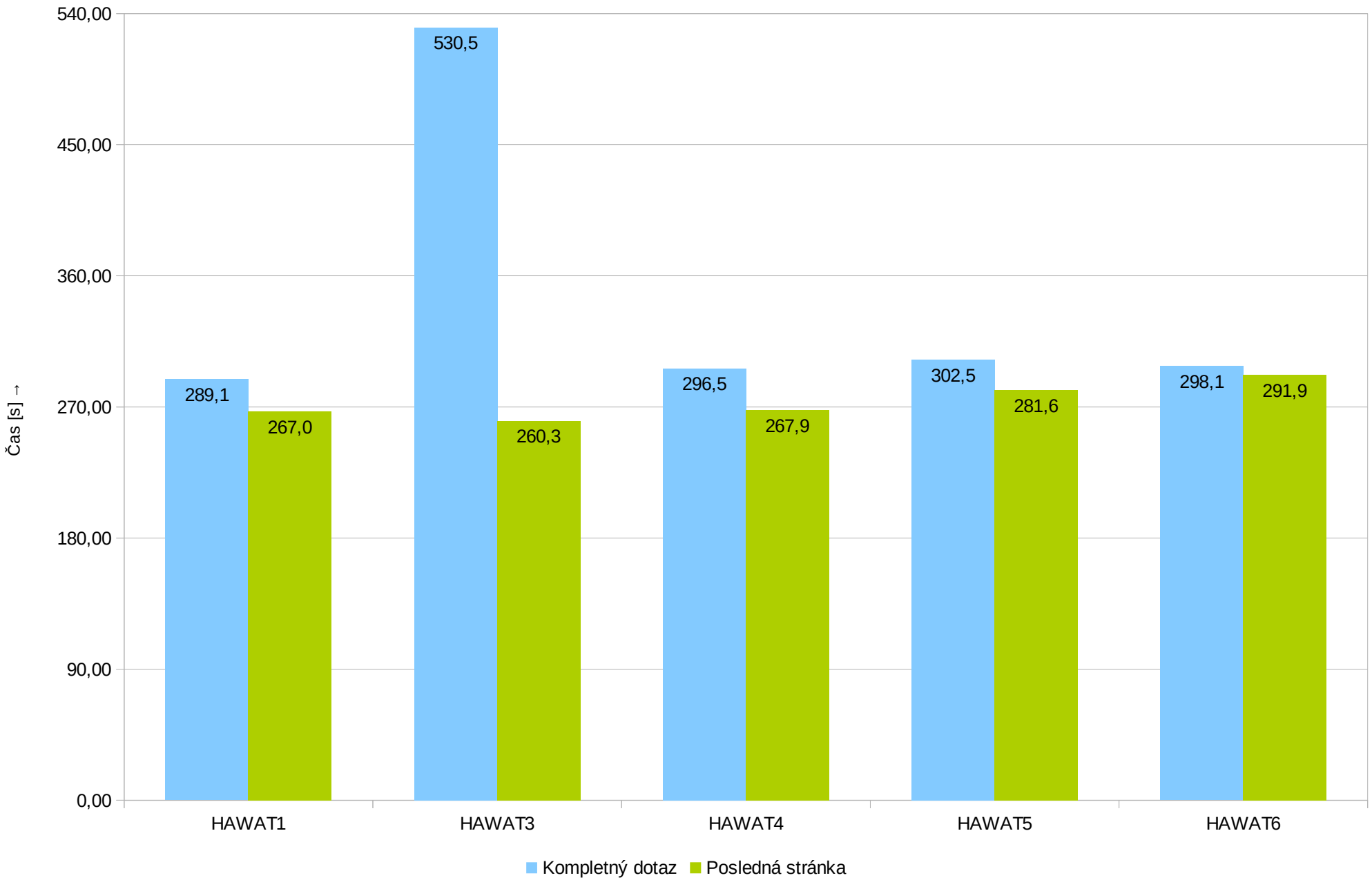
TS3.1: Dávkové čítanie (1M)



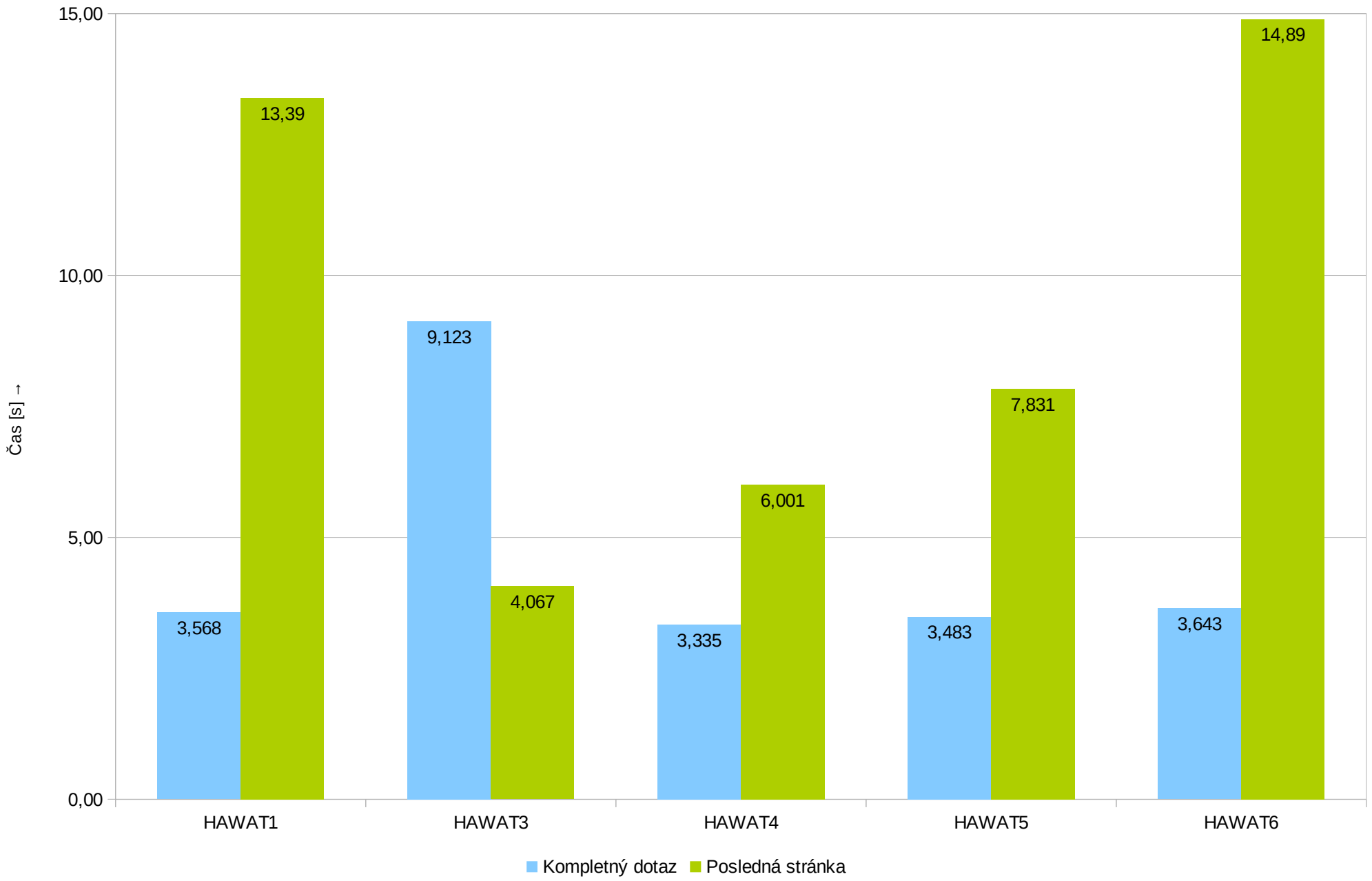
TS3.1: Dávkové čítanie (10k)



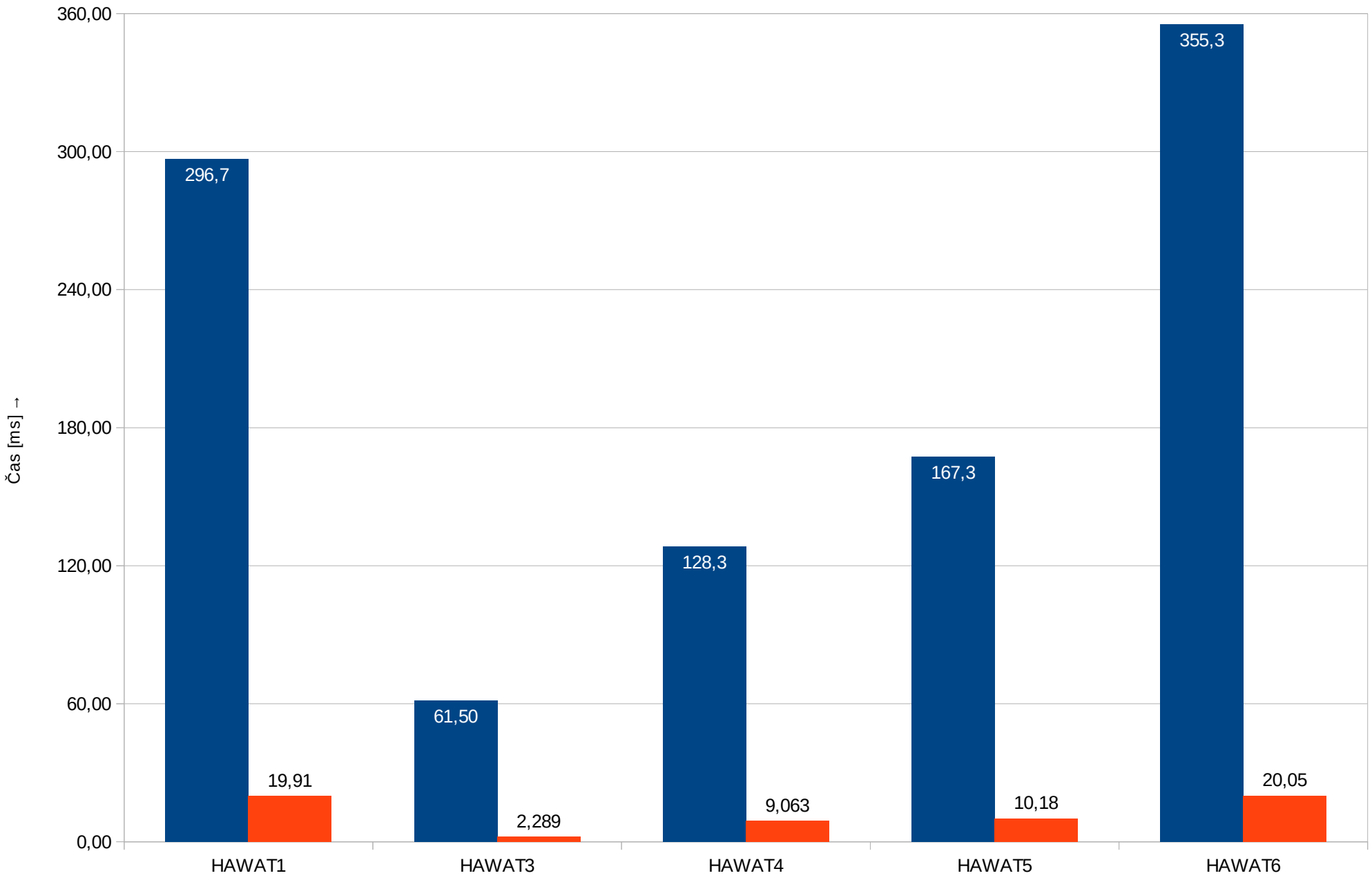
TS3.2: Dávkové čítanie s opačným radením (1M)



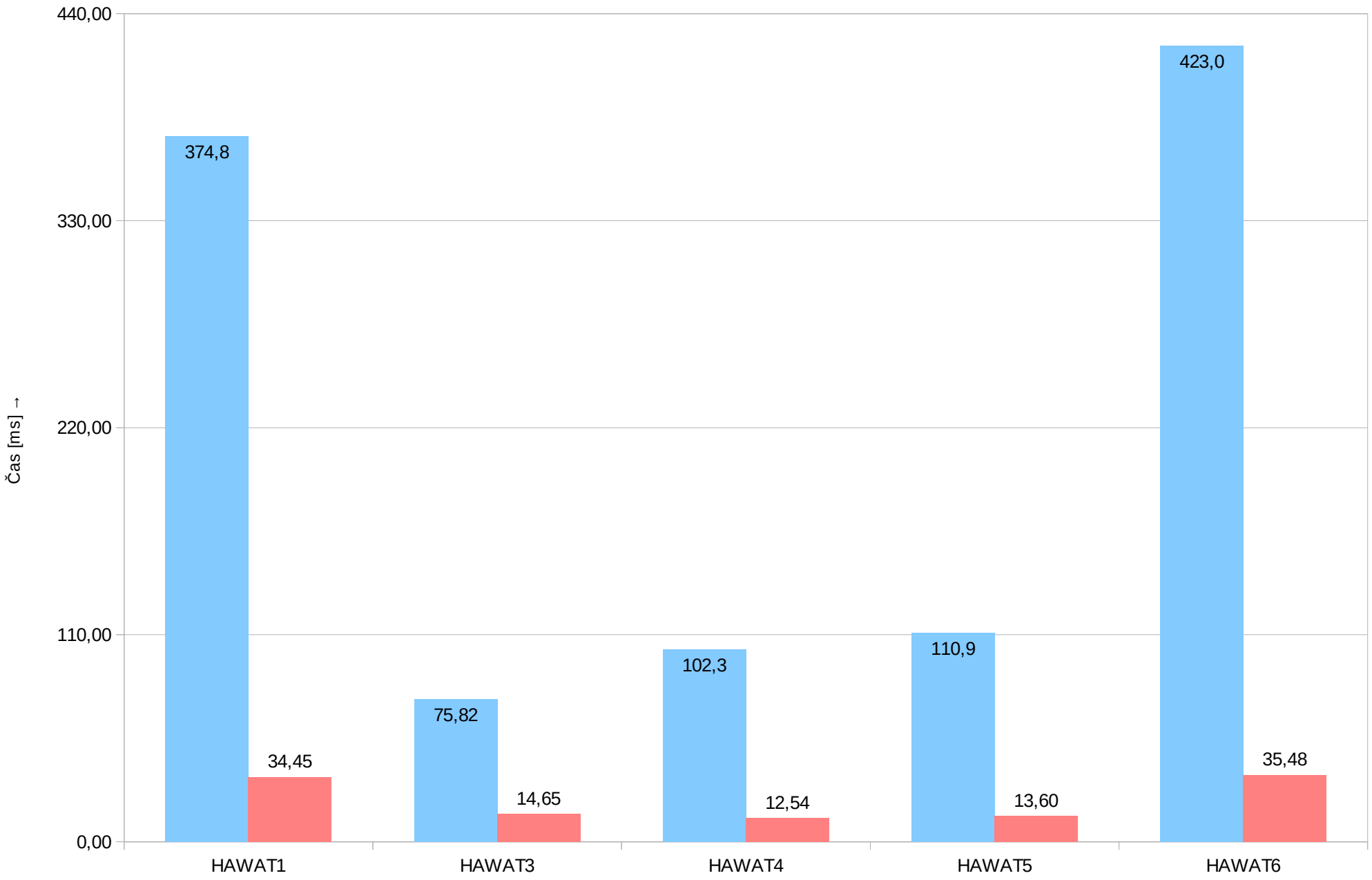
TS3.2: Dávkové čítanie s opačným radením (10k)



TS3.3: Dávkové čítanie za tepla



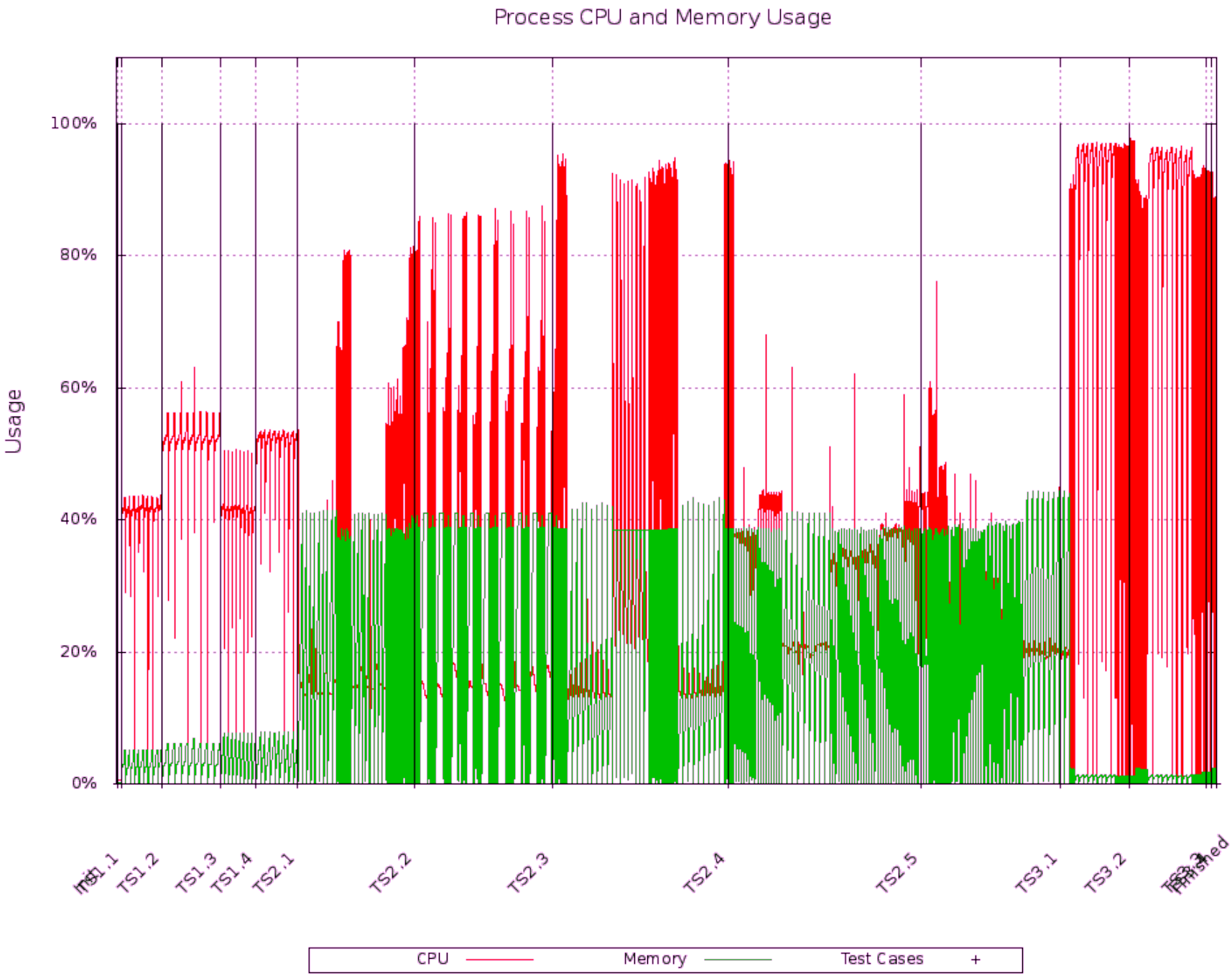
TS3.4: Dávkové čítanie za tepla s opačným radením



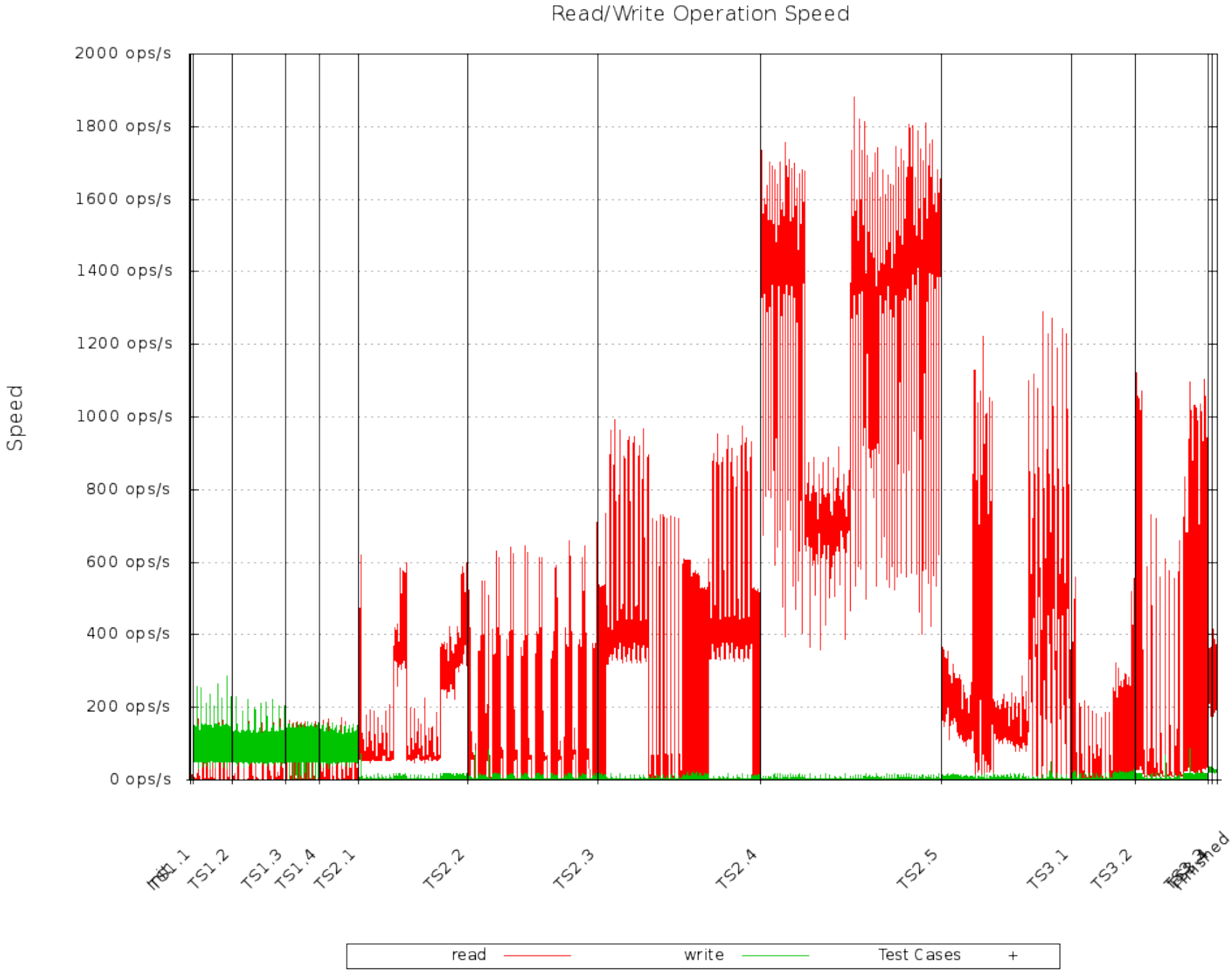
Zhrnutie výsledkov - MongoDB

- Fragmentácia databázového úložiska nie je vo WiredTiger problém,
- Vkladanie dát je výhodnejšie bez indexov, mazanie ešte výraznejšie,
- Pre analytické dotazy sú výhodou vhodné zložené indexy pokrývajúce celú podmienku,
- Je dobré sa vyhnúť použitiu COUNT(), špeciálne ak nemáme optimálny zložený index,
- Je dobré sa vyhnúť opačnému radeniu,
- Pri použití zoznamu ako filtračnej podmienky je vhodné premerať výkon pre konkrétne pole a dĺžku zoznamu,
- Dávkové čítanie funguje obstojne pre dotazy vracajúce veľký počet výsledkov, pozor na malé dotazy,
- Nadväzujúci dotaz s mierne upravenou podmienkou je vykonaný rýchlo.

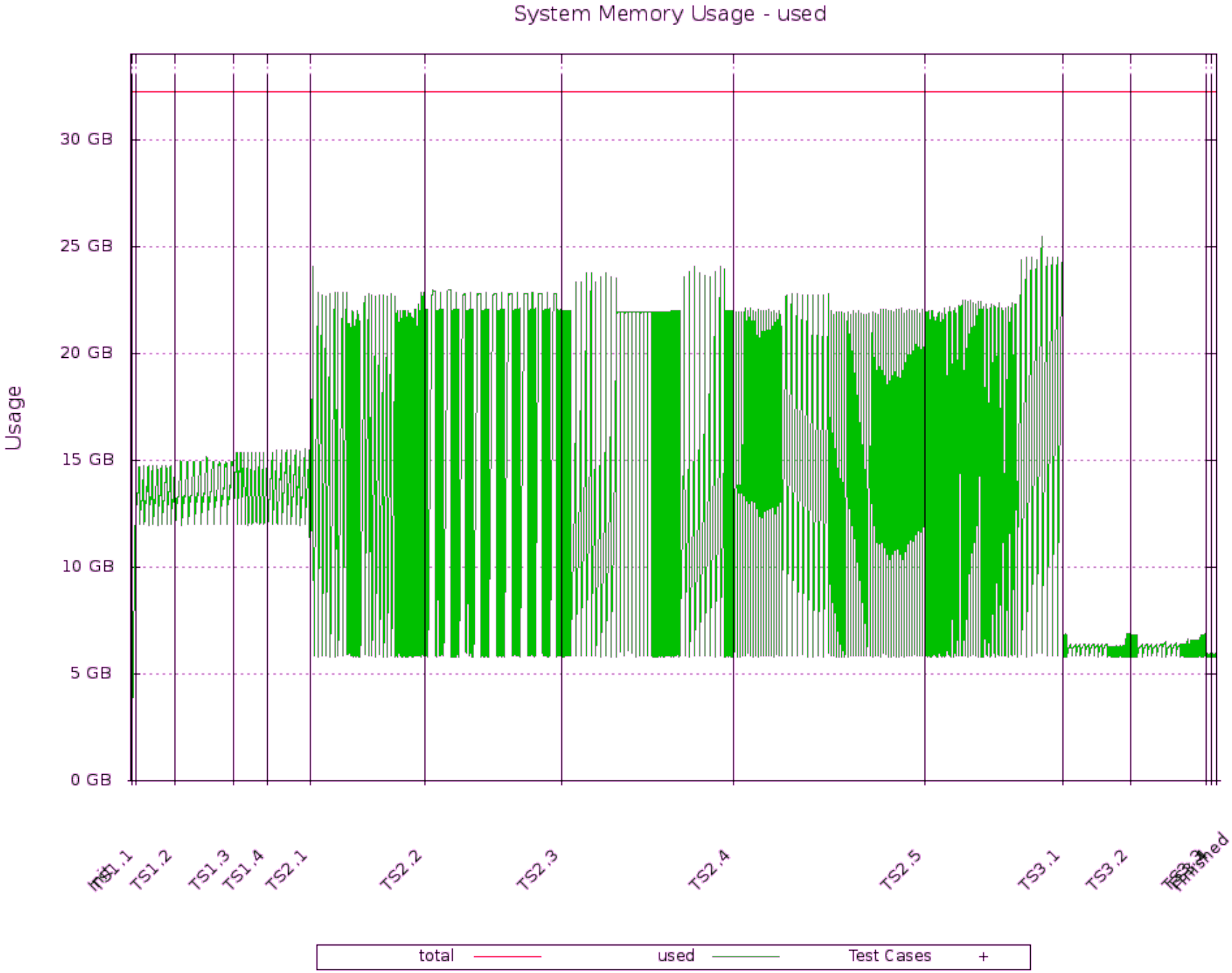
Využitie CPU a RAM procesom mongod



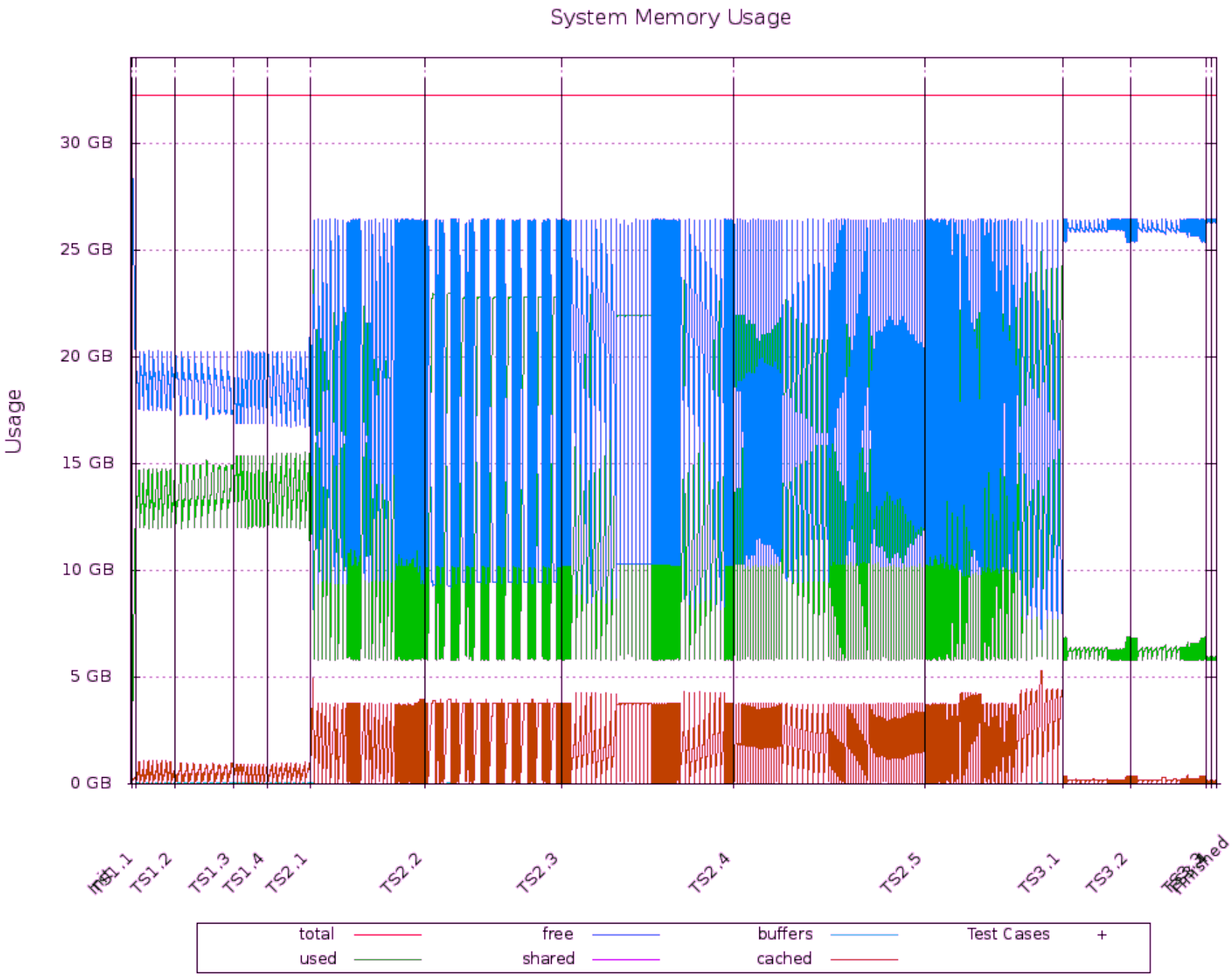
Čítanie a zápis - IOPS



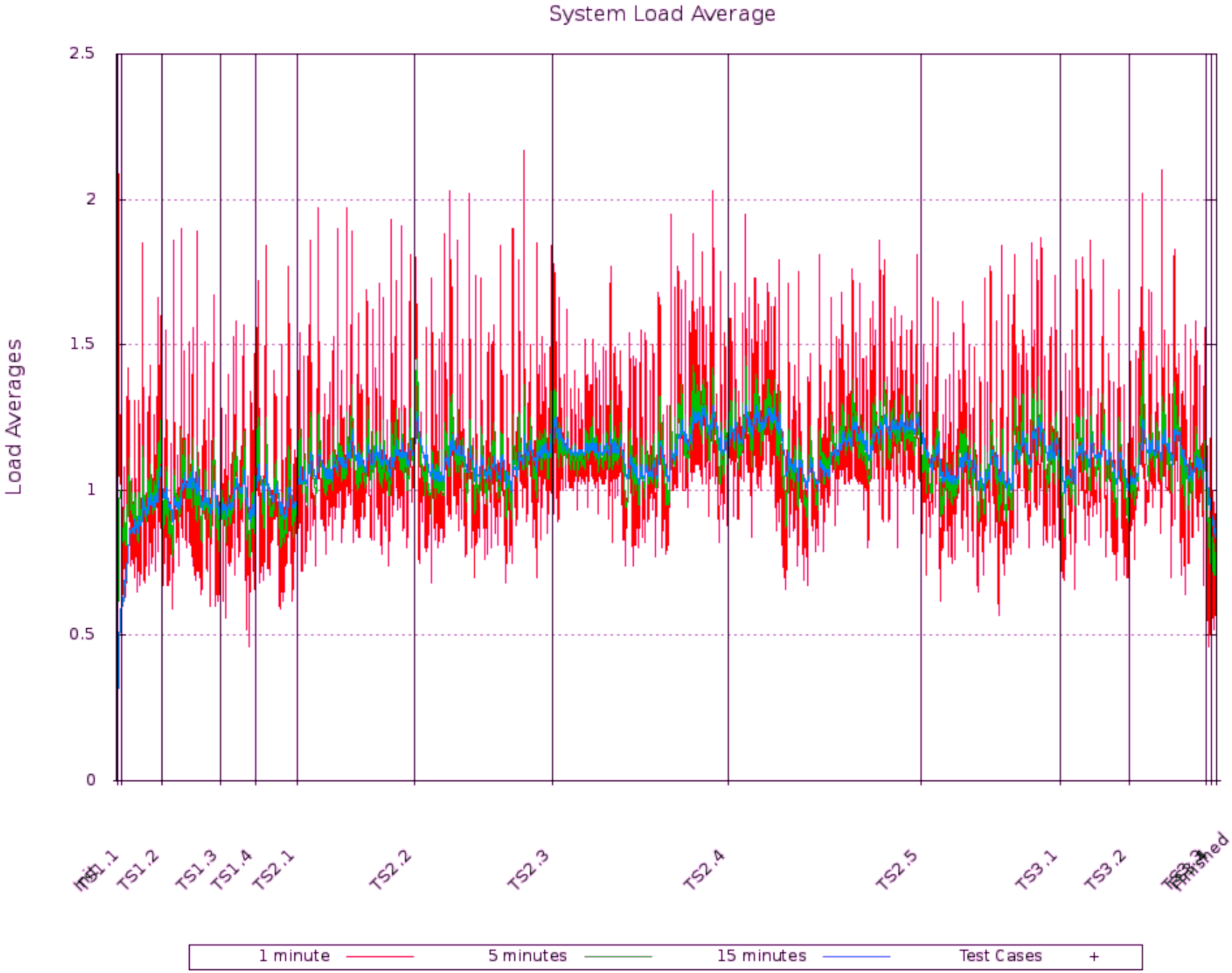
Celkové využitie RAM



Disková cache



Load Average



Poučenie z krízového vývoja

- JSON nie je vhodný dotazovací jazyk,
- Dokumentácia MongoDB je fragmentovaná a neúplná, často vynecháva dôležité detaily, užitočné informácie bývajú roztrúsené v poznámkach,
- Podobne ako dokumentácia je na tom Changelog,
- V prípade práce s poľom pomocou relačných operátorov je potrebné použiť $elemMatch,
- Extrémnu premenlivosť doby behu dotazov sa nepodarilo reprodukovať,
- Najdôležitejšie sú vhodné indexy.

Diskový subsystém

- HW RAID:
  - SATA/SAS SSD,
  - RAID 0, RAID 10,
  - Vhodný radič (PERC H710**P**, H730**P** – FastPath),
  - Striping size,
- NVMe (Intel P3700, Samsung SM1715),
- SW RAID – pozor na škálovanie vzhľadom k potrebnému počtu front na disk,
- LVM – negatívny dopad na výkon,
- Ideálne vyhradený disk/pole len pre DB.

Súborový systém

- ext4 vs BTRFS, ZFS, ...
- Zarovnanie vzhľadom k mazacím blokom SSD,
- Vypnúť journal, alebo použiť nobarrier (BBU),
- Voľby:
  - noatime/relatime: posledný prístup k súboru,
  - nodiratime/reldiratime: k adresárom,
  - data=writeback (BBU),
  - discard (TRIM).
- wm.swappiness: <0,100>, malé hodnoty – preferencia zahodenia diskovej cache, veľké hodnoty – swapovanie programovej pamäte, 1 – minimálny swap bez úplného vypnutia.

IO plánovač

- CFQ (Completely Fair Queuing) – spravodlivé delenie prostriedkov, často default, vhodné pre desktop, nie pre DB, nie pre SSD,
- noop – FIFO fronta, best-effort, minimálny jitter,
- deadline – v podstate noop zabraňujúci vyhladoveniu, lepšie škáluje s počtom front,
- (anticipatory) → CFQ,
- Nastaviteľný pre blokové zariadenia zvlášť,
- Voľba medzi noop a deadline, noop vyhráva pri malom počte front.

Ostatné

- THP (Transparent Huge Pages) – pre SW využívajúci libc malloc() funguje väčšinou dobre, pre jemalloc() a TCMalloc() negatívny dopad,
- Sieťové rozhranie:
  - Jumbo frames – vhodné najmä pre zálohy,
  - Vypnutie zbierania metrík.
- NUMA interleaving,
- Počet procesorov vs. taktovacia frekvencia,
- SMT (Simultaneous multithreading).

Pokračovanie

- Ďalšie DBMS:
  1. **PostgreSQL**,
  2. Elasticsearch,
  3. Couchbase,
- Parametrizácia testov,
- Porovnanie s novšími verziami MongoDB (performance project), pymongo.

1996–2016
CESNET

Otázky?

Radko Krkoš
krkos@cesnet.cz

https://homeproj.cesnet.cz/projects/mentat/wiki/AnalysisDatabaseTesting