UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu

Redis

Lukáš Trkan, 1. ročník

V Hradci Králové
dne 12.05.2025

Obsah

Úvod

Tato práce se konkrétně databází Redis a použitím v režimu clusteru. Zaměřuje se na praktické nasazení a konfiguraci Redis Clusteru v prostředí Dockeru a Docker Compose. Cílem je nejen vytvořit funkční řešení s důrazem na dostupnost a škálovatelnost, ale také ukázat praktické schopnosti Redisu při práci s daty.

Součástí práce je popis architektury clusteru, konfigurace shardingu, replikace, perzistence dat i zabezpečení pomocí TLS a ACL. Významnou roli zde hraje modul RediSearch, který výrazně rozšiřuje možnosti Redisu o fulltextové vyhledávání, filtrování a agregace. Dotazy jsou demonstrovány na datech z platformy Steam. Práce se kromě vlastního řešení se věnuje také případovým studiím nasazení Redisu ve firmách jako BioCatch, Axis Bank nebo Deutsche Börse, které popisují jeho použití ve velkém měřítku.

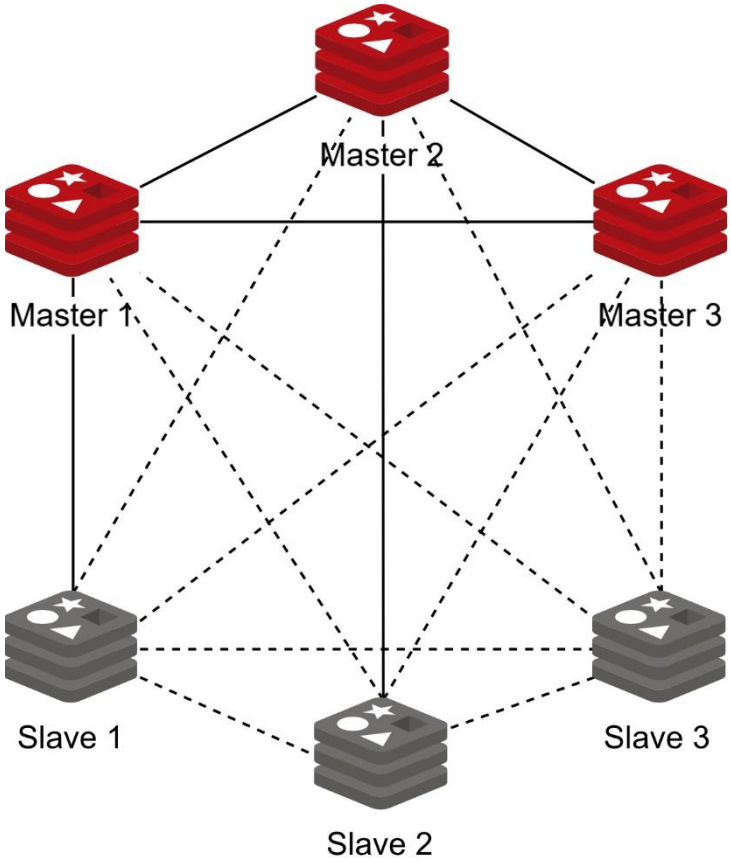
Z hlediska technického postupu bylo řešení zpočátku založeno na vlastnoručně kompilovaném modulu RediSearch, protože dostupné verze nepodporovaly open source cluster. Po vydání open source verze Redis 8 následovala kompilace celého Redis serveru včetně potřebných modulů, z důvodu absence oficiálního image. Ten je nyní ale už veřejně dostupný a finální řešení tak může vycházet oficiálního image Redis 8.

Práce se nezabývá produkčním nasazením, spíše poskytuje základy k použití Clusteru vhodné k dalšímu prozkoumávání funkcí a případů použití.

1 Architektura

1.1 Schéma a popis architektury

Architektura Redis Clusteru je tvořena třemi master uzly, z nichž každý má přiřazenou jednu repliku (slave uzel). Každému uzlu je přidělena část z celkových 16384 hash slotů, přičemž jednotlivé klíče jsou podle výpočtu kontrolního součtu klíče (nebo hash tagu) ukládány na příslušný master a automaticky replikovány i na jeho slave. V případě výpadku některého z master uzlů dojde k automatickému povýšení jeho repliky na nového mastera. Tato konfigurace odpovídá doporučení podle oficiální dokumentace Redis Clusteru.



1.2 Specifika konfigurace

```
# Konfigurace portů
# Vypnutí nešifrované komunikace a povolení šifrované na portu 6379
port 0
tls-port 6379

# Nastavení certifikátů, povolení pro komunikaci v clusteru, s klientem a replikaci
tls-cert-file /certs/redis.crt
tls-key-file /certs/redis.key
tls-ca-cert-file /certs/ca.crt
tls-auth-clients yes
tls-cluster yes
tls-replication yes

# Povolení clusteru, nastavení dostupnosti i ve fail stavu
cluster-enabled yes
cluster-config-file nodes.conf
cluster-node-timeout 5000
cluster-allow-reads-when-down yes
cluster-require-full-coverage no

# Povolení AOF souboru
appendonly yes

# Nastavení sítě – povolení všech IP (řeší docker, FW hosta), povolení dotazovaní ze
všech IP
protected-mode no
bind 0.0.0.0

#Spustit na popředí – potřebné pro docker
daemonize no

# Nastavení uživatelských účtů
masteruser cluster-sync
masterauth syncpassword
aclfile /access-rights.acl

# Načtení modulů
loadmodule /usr/local/lib/redis/modules/redisbloom.so
loadmodule /usr/local/lib/redis/modules/redisearch.so
loadmodule /usr/local/lib/redis/modules/rejson.so
loadmodule /usr/local/lib/redis/modules/redistimeseries.so
```

redis.conf

1.2.1 CAP teorém

Moje řešení splňuje garanci AP, tedy dostupnost (Availability) a odolnost vůči síťovému rozdělení (Partition Tolerance). Cluster je nakonfigurován tak, aby zůstal dostupný i při výpadku uzlů a nevyžadoval plné pokrytí všech hash slotů clusteru. Tedy i v případě kdy

dojde k výpadku nebo odpojení shardu (masteru i jeho slavu), tak bude možné se dotazovat na data na jiných shardech. Konzistence je v tomto modelu částečně obětována – může dojít k dočasným nesrovnalostem mezi jednotlivými replikami což je ale pro zamýšlené použití (pouze read přístup ze strany klientů/zákazníků) dostačující. Klíčová je pro mě vysoká dostupnost, nízká latence a schopnost clusteru fungovat i v degradovaném režimu.

1.2.2 Cluster

V řešení je použit pouze jeden cluster, protože Redis nepodporuje propojení více clusterů dohromady. Každý cluster funguje samostatně a není možné je mezi sebou přímo propojovat. Jeden cluster ale pro moje potřeby plně stačí – umožňuje rozdělení dat mezi více uzlů, replikaci i vysokou dostupnost. Nasazení více clusterů by navíc zbytečně zvýšilo složitost celého řešení na případné aplikační úrovni, aniž by přineslo reálný přínos.

1.2.3 Uzly

Uzel je v kontextu práce každý samostatný Redis server, ze kterých je následně vytvořen cluster. V řešení používám 6 uzlů – tři jako master a tři jako jejich repliky. Tento počet a tato konfigurace je doporučena v dokumentaci pro Redis Cluster.

1.2.4 Sharding

Sharding je technika horizontálního škálování databáze, při které se data rozdělují napříč několika samostatnými databázovými uzly (shardy). Každý shard obsahuje pouze část celkových dat a zpracovává jen podmnožinu dotazů, čímž dochází ke zvýšení výkonu a škálovatelnosti systému.

V Redis Clusteru sharding využívá hash sloty pro rozdělení klíčů mezi jednotlivé uzly. Celkem se pracuje s 16384 hash sloty, kdy hash slot je určen výpočtem pomocí kontrolního součtu celého klíče nebo tagu (část klíče v {}). Každý shard má stanoveno, jaký rozsah slotů obsluhuje, což lze konfigurovat ručně nebo nechat rozdělit automaticky při vytváření clusteru. Zde je použitu automatické dělení mezi 3 shardy, každý shard tedy obsluhuje přibližně třetinu hash slotů. Počet shardů vychází z doporučení uvedeném v dokumentaci.

1.2.5 Replikace

Replikace udržování kopie dat z master nody na dalších nodech. Kopírování dat probíhá prakticky v reálném čase. V Redis Clusteru je ve výchozím stavu zapnutá asynchronní replikace, která sice snižuje latenci, ale potvrzování replikovaných dat probíhá periodicky takže může při výpadku masteru vzniknout částečná ztráta dat. Tomuto lze přecházet použitím synchronní replikace, ovšem za cenu vyšší latence. Zde jsou použity tři replikace – každá pro jeden master node. Synchronní ukládání dat nemá cenu řešit, vzhledem k tomu že se jedná pouze o cache. Počet replikací vychází z doporučení v cluster dokumentaci.

1.2.6 Perzistence dat

Perzistence dat je schopnost trvalého uložení dat, která tak zůstanou zachována i po vypnutí databáze. Prakticky se jedná o uložení dat na disk. V projektu mají perzistentní úložiště všechny nody, po restartu slave node tak nebude vždy nutné kopírovat veškerá data z příslušného masteru.

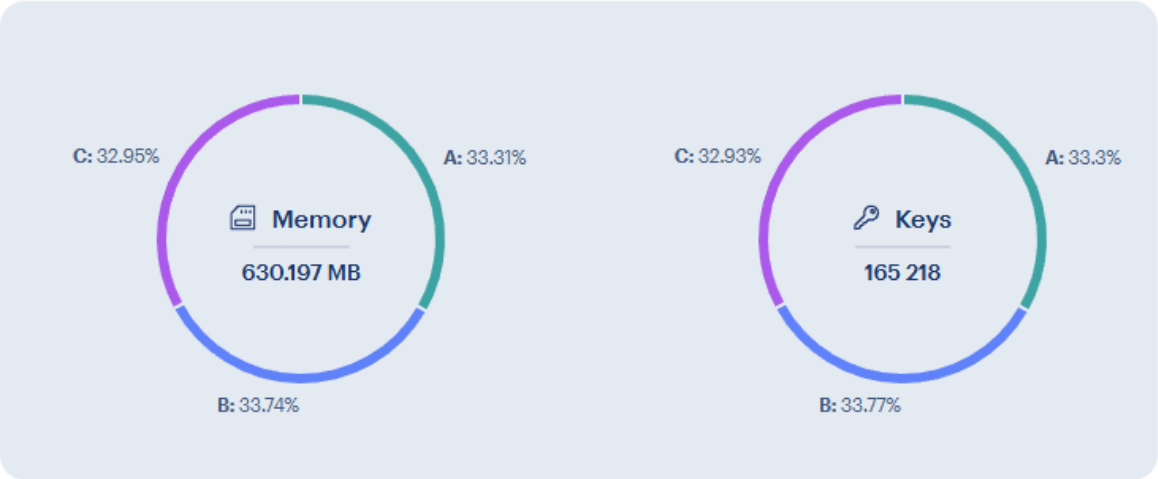
Redis je primárně in-memory databáze, což znamená že jsou data načtena v operační paměti RAM (primární paměť), ze které po restartu ale nenávratně zmizí. Tato konfigurace může být v některých dostatečná například při použití Redisu jako jednoduchou cache nebo jako message queue. Ve výchozím nastavení Redis pravidelně ukládá na disk (sekundární paměť) RDB soubory (Redis Database Backup), což je snapshot celé databáze v binární formě. V případě výpadku ale hrozí ztráta dat od posledního snapshotu. Existuje proto druhý typ ukládání dat ve formě AOF (Append Only File), což je seznam všech příkazů manipulujících s daty, takže při provedení všech příkazů ze souboru data vždy dostaneme do stejné podoby.

V projektu se data ukládají jak do RDB, tak do AOF souborů.

1.2.7 Distribuce dat

Cluster je složen ze 3 shardů kdy ke každému master shardu existuje jeho slave s replikou dat. Data jsou na clusteru rozdělena pomocí hash tagů v podobě hodnoty pole ID, což zajistí že související data jsou u sebe na jednom shardu a je možné s nimi pracovat například i pomocí Lua scriptů nebo jiných Redis modulů nepodporující cluster. Zápis i čtení probíhá z jednotlivých master shardů. Použitím příkazu READONLY je ale možné povolit i čtení potencionálně neaktuálních dat z replik. Čtení z replik sice může snížit zátěž na masteru, ale je nutné vzít v potaz, zda bude nebo nebude vadit případná neaktuálnost dat – v případě špatné hodnoty nastavení to vadit bude více než v případě neaktuálního popisku produktu.

V práci se spoléhá na automatické rozdělení dat v Redisu, který data rozděluje poměrně rovnoměrně. Rovnoměrnost rozdělování může rozbít nesprávné použití hash tagů – například otagováním na základě neměnné části klíče ({student}:1023). Díky tomu, že v práci jsou použity tagy na proměnlivou část klíčů (ID) tak jsou i tato data rovnoměrně rozdělena. Konkrétně je 165218 klíčů rozděleno na 55016 + 55797 + 54405, na každém shardu je tedy přibližně 33 % dat. Resharding zde není nutný provádět, nutný by byl pouze v případě přidání nebo odebrání shardu, a to konkrétně příkazem „redis-cli --cluster reshard“.



3 Primary nodes ↑		Total Keys
A	172.28.0.11:6379	55 016
B	172.28.0.12:6379	**55 797**
C	172.28.0.14:6379	54 405

1.2.8 Zabezpečení

Veškerá komunikace v clusteru i s klientem vyžaduje šifrování pomocí TLS aby se zamezilo možnosti odposlechů. V práci je použit self-signed certifikát, v produkčním prostředí by bylo ideální použít certifikát vydaný nějakou z certifikačních autorit (Let's Encrypt, Cloudflare). Při připojování se ke clusteru je potřeba u sebe mít certifikát ověřovací autority a příslušný pár klíčů.

Dalším krokem v bezpečnosti je použití ACL, díky kterému lze specifikovat přístupová práva jednotlivým uživatelům. Uživatelé jsou v řešení celkem 4. Prvním z nich je výchozí uživatel, sloužící ke konfiguraci a má přidělena všechna práva. Dalším uživatelem je „cluster-sync“ sloužící pro synchronizaci napříč clusterem. Jako poslední existují uživatelé „cache-client“ a „cache-service“, kdy první má pouze oprávnění pro čtení (pro klientskou aplikaci) a druhý má přístup i pro zápis (service plnící Redis daty).

2 Funkční řešení

2.1 Struktura

```
├── .github/
│   └── workflows/
│       └── docker.yml                  # Předpis pro build docker imagů
├── Data/
│   ├── D1.ipynb                        # Práce se steam.csv
│   ├── D2.ipynb                        # Práce se steamspy_tag_data.csv
│   ├── D3.ipynb                        # Práce se steam_requirements_data.csv
│   ├── steam.csv                       # Dataset Steam aplikací
│   ├── steam_requirements_data.csv     # Dataset požadavků Steam aplikací
│   └── steamspy_tag_data.csv           # Dataset tagů Steam aplikací
├── Dotazy/
│   └── dotazy.md                       # Soubor s dotazy a jejich popisem
└── Funkcni_reseni/
    ├── certs/                          # Složka pro uložení certifikátů
    ├── common-services.yml             # Předpis services pro docker compose
    ├── compose.yml                     # Hlavní compose
    ├── config/
    │   ├── access-rights.acl           # Konfigurace uživatelů a práv
    │   └── redis.conf                  # Konfigurace Redis
    ├── delete-data.sh                  # Pomocný script pro reset databáze
    ├── docker/
    │   ├── init.Dockerfile             # Dockerfile pro inicializační script
    │   ├── rcproxy.Dockerfile          # Dockerfile pro Redis Cluster Proxy
    │   ├── redis.Dockerfile            # Dockerfile pro Redis 8
    │   └── stack.Dockerfile            # Dockerfile pro Redis 7 s OSS RediSearch
    ├── init/
    │   ├── init-cluster.sh             # Script pro inicializaci clusteru
    │   ├── load_data.py                # Script pro načtení dat do databáze
    │   ├── redis_client.py             # Script pro vytvoření Redis klienta
    │   └── requirements.txt            # Seznam požadavků pro Python scripty
    └── start.sh                        # Script pro spuštění řešení
```

2.1.1 Docker compose

Compose je složen ze 2 částí – ze samotného compose.yml a z common-services.yml. V common-services.yml je připravený předpis společných parametrů pro node Redisu který je následně rozšířen v samotném compose.yml.

```
services:
  redis-base:
    image: ghcr.io/lukastrkan/redis8
    volumes:
      - ./config/redis.conf:/redis.conf:ro
      - ./certs:/certs:ro
      - ./config/access-rights.acl:/access-rights.acl:ro
    healthcheck:
      test: [ "CMD", "redis-cli", "--tls", "--cert", "/certs/redis.crt", "--key",
"/certs/redis.key","--cacert", "/certs/ca.crt", "-a", "strongpassword", "ping" ]
      interval: 5s
      timeout: 3s
      retries: 5
      start_period: 10s
```

common-services.yml

Předpis tedy obsahuje definici základní redis node, která vychází z mého image hostovaného na GitHubu, aktuálně už lze ale použít oficiální image který ale v době přípravy ještě nebyl k dispozici. Další konfigurací jsou mounty složek a souborů, společných pro všechny nody – konfigurace pro redis, vygenerované certifikáty a soubor s definicí uživatelů a přístupových práv. Jako poslední je nadefinována kontrola stavu běžícího kontejneru pomocí příkazu redis-cli a ping.

```
services:
  redis-init:
    image: ghcr.io/lukastrkan/redis8-init
    depends_on:
      redis-1:
        condition: service_healthy
      … obdobně pro každý node
    volumes:
      - ./data/cluster:/cluster
      - ./certs:/certs:ro
    networks:
      redis-net:
        ipv4_address: 172.28.0.10
  redis-insight:
    image: redis/redisinsight:latest
    container_name: redis-insight
    ports:
      - "5540:5540"
    restart: unless-stopped
    user: root
    volumes:
      - ./data/redis-insight:/data
      - ./certs:/certs:ro
    environment:
      - RI_REDIS_HOST=172.28.0.11
      - RI_REDIS_PORT=6379
      - RI_REDIS_ALIAS=Redis Cluster Master
      - RI_REDIS_PASSWORD=strongpassword
      - RI_REDIS_TLS=true
      - RI_REDIS_TLS_CA_PATH=/certs/ca.crt
      - RI_REDIS_TLS_CERT_PATH=/certs/redis.crt
      - RI_REDIS_TLS_KEY_PATH=/certs/redis.key
    networks:
      redis-net:
        ipv4_address: 172.28.0.100
    depends_on:
      redis-init:
        condition: service_completed_successfully
  redis-1:
    extends:
      file: common-services.yml
      service: redis-base
    volumes:
      - ./data/redis-1/data:/data
    ports:
      - 6371:6379
      - 16371:16379
    networks:
      redis-net:
        ipv4_address: 172.28.0.11
```

```
...obdobně pro zbytek nodů
networks:
  redis-net:
    driver: bridge
    ipam:
      config:
        - subnet: 172.28.0.0/16
```

compose.yml

Samotný compose obsahuje definici inicializačního kontejneru používající image z GitHubu a slouží k automatickému nastavení celého clusteru. Díky konfiguraci depends_on obsahující všechny redis nody se init script spustí až po jejich nastartování. Dále jsou nastaveny mounty k certifikátům a ke složce pro uložení příznaku vytvoření clusteru. Dále je nastavena IP adresa uvnitř Docker sítě. Inicializační scripty jsou součásti image definovaného v docker/init.Dockerfile.

Další službou je webové rozhraní Redis Insight využívající oficiální image. Služba ven mapuje port 5540 na kterém je ono rozhraní dostupné. Kontejner je spuštěn pod root uživatelem, má mount pro datovou složku obsahující nastavení aplikace a na složku s certifikáty. Připojení na cluster je nadefinováno pomocí environment proměnných, obsahujících například adresu, port nebo cesty k certifikátům. Služba se díky použití depends_on a service_completed_successfully spustí až po inicializaci clusteru.

Compose samozřejmě obsahuje i jednotlivé nody redisu (1-6), které extendují službu redis-base z common-services.yml. U každého kontejneru tak stačí nadefinovat jen mount kam bude daný kontejner ukládat data, mapping portů a nastavení IP adresy v Docker síti.

Jako poslední je nadefinovaná síť a její IP rozsah. Tako možnost není povinná, zde je použita jen aby umožnila lepší přehlednost díky manuální definici IP adres uvnitř clusteru.

2.2 Instalace

Pro spuštění řešení stačí ve složce „Funkcni_reseni“ spustit script „start.sh“. Po úspěšném nastartování je na portu 5540 dostupné webové UI Redis Insight.

```
cd Funkcni_reseni
bash start.sh
```

Script při prvním spuštění vygeneruje nové certifikáty potřebné pro spuštění a následně spustí kontejnery pomocí docker compose. Nejprve se spustí všechny instance Redisu, následně se po jejich úspěšném načtení spustí i inicializační script. Inicializační script při prvním spustí konfigurační příkazy pro vytvoření clusteru, načte do databáze data a vytvoří indexy pro modul RediSearch.

3 Případy užití a případové studie

Jeden z hlavních účelů, pro které se Redis používá, je in-memory ukládání dat ve formátu klíč–hodnota, zejména jako cache uchovávající často používané informace. Tím se snižuje odezva systému a zátěž na hardwarové nebo softwarové prostředky, jako jsou API či jiné databáze. Redis se dále často využívá jako úložiště pro uživatelské relace.

Dalším běžným případem použití je asynchronní zpracování úloh – ať už prostřednictvím front úkolů, nebo předávání zpráv. Fronty lze realizovat pomocí datového typu list, přičemž aplikace musí sama zajišťovat přidávání a odebírání položek. Redis však podporuje také vestavěné zprávy pomocí příkazů PUBLISH a SUBSCRIBE, které umožňují jednoduchou komunikaci mezi komponentami systému.

Díky podpoře modulů umožňuje Redis provádět i pokročilejší operace nad daty. V této práci je využit například modul RediSearch, který umožňuje fulltextové vyhledávání, řazení, agregace, autocomplete nebo práci s geografickými daty. Dalším modulem je RedisJSON, který přidává plnohodnotnou podporu pro práci s JSON formátem, a RedisTimeSeries, jenž je určen pro ukládání a analýzu časových řad. Za zmínku stojí také dříve podporovaný modul RedisGears, který přidával podporu pro skriptování v JavaScriptu. RedisGears je kompatibilní s Redis 8, avšak v režimu clusteru již nepodporuje autentizaci pomocí hesla.

Pro projekt jsem si Redis zvolil pro jeho rychlost, a protože se jedná o nejrozšířenější řešení v oblasti cachování s podporou pro prakticky všechny platformy. Vzhledem k zamýšlenému použití je výhodou i vysoká dostupnost v případě provozu v clusteru, konzistence dat je v tomto případě vedlejší. Rozšiřitelnost pomocí modulů je další silnou stránkou. Konkrétně šlo o modul RediSearch umožňující filtrování dat nebo jejich agregaci, což jsou funkce více než vhodné zobrazení katalogu. Dalším důvodem pro zvolení Redisu byl i fakt, že se jedná o technologii, kterou používáme v práci. Redis v kombinaci s RediSearch je také pravděpodobně nejlepší fulltextové vyhledávání v moment absence ElasticSearch v možných zadáních.

Přímou alternativu Redisu Valkey jsem nezvolil, protože se jedná o poměrně nové řešení a nemá taková ekosystém modulů – konkrétně neexistuje podpora pro Fulltextové vyhledávání. Alternativa Redisu v podobě Dragonfly by byla mnohem lepší volba. Cassandra fultextové vyhledávání nepodporuje vůbec, proto nemělo cenu o ní dál uvažovat. MongoDB sice má podporu fulltextového vyhledávání, ale pracuje s daty na disku a je proto pomalejší. Teoreticky by se ale jednalo o vhodnou alternativu.

3.1 Případová studie: BioCatch

BioCatch je zaměřený na detekci podvodného chování prostřednictvím behaviorální biometriky – analýzy způsobu, jakým uživatelé interagují se zařízeními – například jak drží telefon nebo pohybují kurzorem. Systém zpracovává více než 5 miliard transakcí měsíčně a obsluhuje přes 70 milionů uživatelů, což klade vysoké nároky na rychlost a spolehlivost zpracování dat v reálném čase. Původní technologický stack nebyl schopen takové zatížení efektivně zvládat, a proto se rozhodlo pro přechod na Redis, konkrétně ve formě managed služby Redis Cloud. Redis se stal jediným prvkem s perzistencí a ukládá různé typy dat, včetně behaviorálních metadat, uživatelských a podvodových profilů, geolokačních informací a systémových konfigurací. V daném čase Redis spravuje přibližně 3 TB dat a 200 milionů klíčů, které využívají různé mikroslužby tvořící aplikaci jako celek. Nasazení Redisu vedlo k výraznému zlepšení provozu– Redis poskytuje nízkou latenci (méně než 1 ms i při vysokém zatížení) a vysokou dostupnost bez výpadků. Díky oddělení perzistentní části od aplikační je architektura flexibilnější a odolnější vůči chybám. BioCatch tak získal stabilní platformu, která zvládá nároky na rychlé zpracování dat a současně umožňuje další škálování dle potřeb.

3.2 Případová studie: Axis Bank

Axis Bank čelila problému s poskytováním aktuálních informací v mobilní aplikaci, zejména v případech, kdy zákazníci prováděli změny na pobočce. Tyto změny se do aplikace nepropisovaly v reálném čase kvůli omezení infrastruktury, což vedlo ke stížnostem na nepřesná nebo opožděná data a zhoršení UX. Za účelem vyřešení této situace banka nasadila nový systém využívající Redis pro poskytování aktuálních údajů o účtech více než šesti milionům uživatelů denně. Pomocí synchronizačních nástrojů se replikují klíčová data do tradiční databáze a důležitá data se přes Redis Data Integration (RDI) zpřístupňují v reálném čase pro mobilní aplikaci. Redis zde slouží jako rychlá přístupová vrstva, která umožňuje čtení dat bez zpoždění, a to i při vysoké zátěži systému. Tento přístup vedl k 76% snížení odezvy. Díky nasazení Redis se podařilo výrazně zlepšit spolehlivost a aktuálnost informací poskytovaných v mobilním bankovnictví, eliminovat nutnost manuálních zásahů a zaručit, že zákazníci mají přístup k přesným údajům o svých účtech bez ohledu na to, zda byly změny provedeny online nebo offline.

3.3 Případová studie: Deutsche Börse

Deutsche Börse Group je mezinárodní burzovní organizace, která při poskytování reportovacích služeb svým zákazníkům a regulatorním orgánům potřebuje zajistit rychlé a spolehlivé zpracování velkých objemů dat. Hlavním cílem bylo vybudovat systém, který zvládne vysokou propustnost a nízkou odezvu, což je ve finančním sektoru naprosto nezbytné. Společnost zvolila technologii Redis jako klíčový nástroj pro realizaci tohoto řešení. Redis zde funguje jako inteligentní cache pro datový sklad, do kterého se sbírají obchodní kotace a transakce prostřednictvím nástroje Kafka. Redis tato data následně agreguje, čistí a připravuje pro uložení v databázi Oracle, kde jsou dále zpracovávána pro tvorbu regulačních výstupů. Použití Redisu přineslo významné zlepšení výkonu – umožňuje zpracování dat v reálném čase, škálování podle potřeby a zajišťuje nepřetržitou dostupnost systému. Tím Deutsche Börse dosáhla optimalizace svého aplikačního frameworku, přičemž systém je nyní připraven zvládnout i očekávaný nárůst objemu dat. Redis tak hraje zásadní roli v infrastruktuře, která podporuje nejen rychlé zpracování, ale i spolehlivé a včasné plnění legislativních povinností vůči regulátorům a klientům.

4 Výhody a nevýhody

4.1 Výhody Redis

4.1.1 Rychlost
Přístup k datům je velice rychlý díky tomu že se jedná primárně o in-memory databázi.

4.1.2 Univerzálnost
Redis použít na ukládání téměř čehokoliv, převod dat ale musí provést aplikace. Kromě ukládání dat lze Redis používat i jako message broker.

4.1.3 Snadná integrace
Podpora Redisu existuje na většinu platforem, od webových aplikací po IoT zařízení.

4.1.4 Jednoduchost
Redis má velice jednoduché dotazování, primitivní dotazy se skládají pouze z několika klíčových slov.

4.2 Nevýhody Redisu

4.2.1 Paměťové nároky
Veškerá data jsou načtena v paměti, při větším množství dat lze narazit na nedostatek operační paměti a je nutné škáloval –přidáním RAM nebo rozšířením clusteru.

4.2.2 Jednoduchost
Dotazování neumožňuje některé konstrukce jako joiny. Tato logika je potřeba implementovat na aplikační úrovni.

4.2.3 Připojení ke clusteru
V případě použití clusteru je nutné, aby klientská aplikace podporovala cluster připojení a měla přístup ke všem nodám. Případně je potřeba nasadit proxy, která cluster vystavuje jako jeden node.

4.3 Výhody řešení

4.3.1 Bezpečnost
Cluster je zabezpečen pomocí šifrované komunikace i pomocí ACL řídící oprávnění jednotlivých uživatelů.

4.3.2 Vysoká dostupnost
Cluster má povolené čtení i v případě fail stavu clusteru. Uživatelům se tak dostanou alespoň některá data.

4.3.3 Fulltextové vyhledávání
Použitím modulu RediSearch je rozšířena funkčnost samotného Redisu o fulltextové vyhledávání nebo agregace.

4.4 Nevýhody řešení

4.4.1 Nekonzistence dat

Díky konfiguraci upřednostňující dostupnost je ohrožena konzistence dat, pro tento use case to ovšem není podstatný problém.

4.4.2 Náročnost na paměť

Využití RediSearch a jeho indexů zvyšuje paměťové nároky.

5 Další specifika

Za specifické lze považovat použití vlastních docker imagů. Z počátku tvorby projektu je ponechaný dockerfile pro Redis Stack 7 s OSS verzí RediSearch, to samé platí pro build Redisu 8. V době tvorby ještě nebyla verze 8 oficiálně dostupná jako hotový image. Dockerfily doplňuje připravený image pro Redis Cluster Proxy a inicializační image obsahující scripty a závislosti potřebné pro prvotní spuštění. Image jsou automaticky publikované pomocí GitHub Actions do GitHub Container Registry.

Samotný cluster je jinak implementovaný podle doporučení v dokumentaci.

6 Data

6.1 Hry

Dataset obsahující základní informace o hrách na platformě Steam.

6.1.1 Základní informace

- Soubor: steam.csv
- Zdroj: https://www.kaggle.com/datasets/nikdavis/steam-store-games
- Počet záznamů: 27075
- Počet sloupců: 18

6.1.2 Sloupce

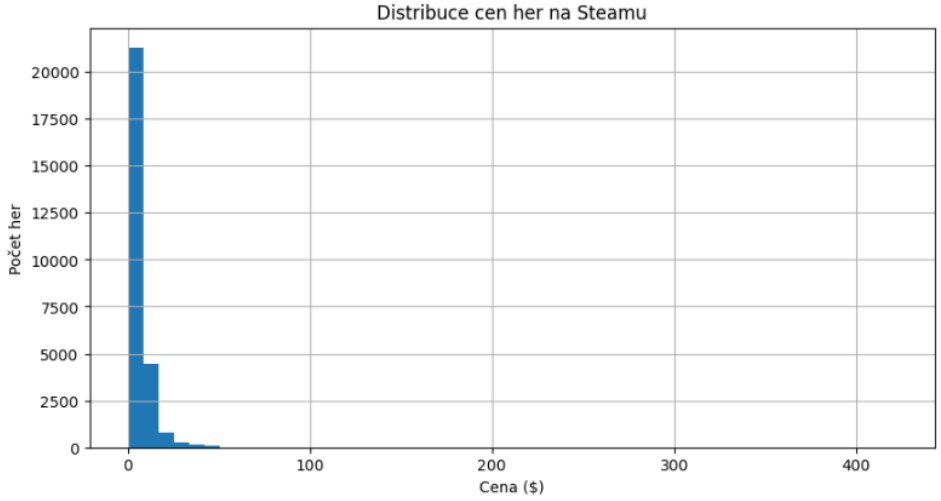
Název	Popis
Appid	ID aplikace
name	Název aplikace
release_date	Datum vydání ve formátu YYYY-MM-DD
english	Podpora angličtiny
developer	Vývojář
publisher	Vydavatel
platforms	Podporované platformy
required_age	Minimální věk
categories	Kategorie
genres	Žánry
steamspy_tags	Tagy z platformy SteamSpy
achievements	Počet achievementů
positive_ratings	Počet pozitivních hodnocení
negative_ratings	Počet negativních hodnocení
average_playtime	Průměrně odehraný čas
median_playtime	Medián odehraného času
owners	Rozsah počtu vlastníků
price	Cena

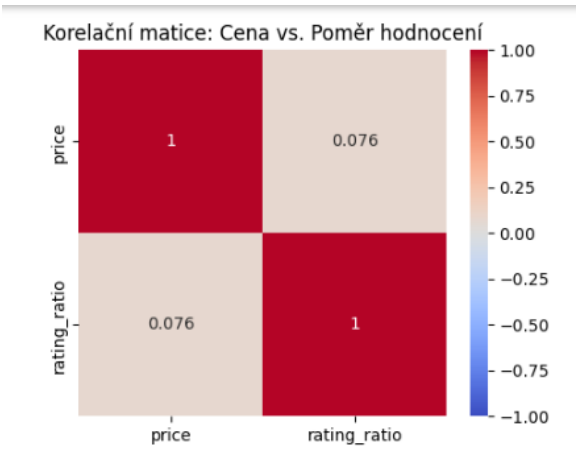
6.1.3 Grafy

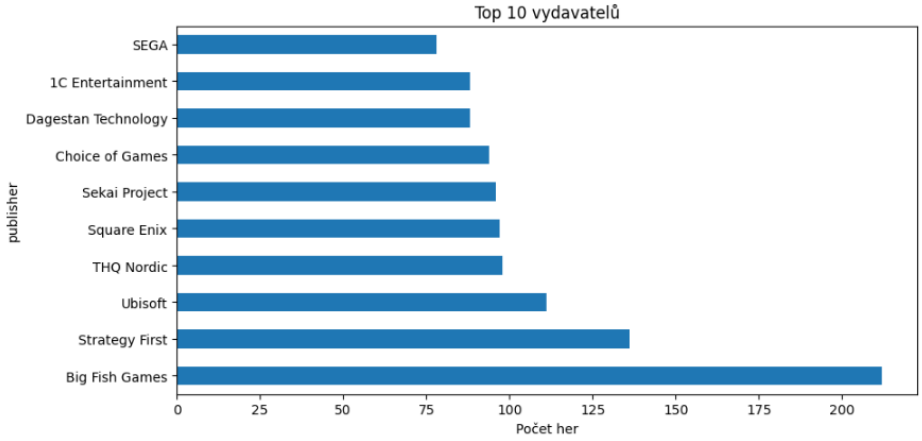
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27075 entries, 0 to 27074
Data columns (total 24 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   appid             27075 non-null  int64
 1   name              27075 non-null  object
 2   release_date      27075 non-null  object
 3   english           27075 non-null  int64
 4   developer         27074 non-null  object
 5   publisher         27061 non-null  object
 6   platforms         27075 non-null  object
 7   required_age      27075 non-null  int64
 8   categories        27075 non-null  object
 9   genres            27075 non-null  object
 10  steamspy_tags     27075 non-null  object
 11  achievements      27075 non-null  int64
 12  positive_ratings  27075 non-null  int64
 13  negative_ratings  27075 non-null  int64
 14  average_playtime  27075 non-null  int64
 15  median_playtime   27075 non-null  int64
 16  owners            27075 non-null  object
 17  price             27075 non-null  float64
 18  total_ratings     27075 non-null  int64
 19  rating_ratio      27075 non-null  float64
 20  windows           27075 non-null  bool
 21  mac               27075 non-null  bool
 22  linux             27075 non-null  bool
 23  negative_ratio    27075 non-null  float64
dtypes: bool(3), float64(3), int64(9), object(9)
memory usage: 3.5+ MB
```

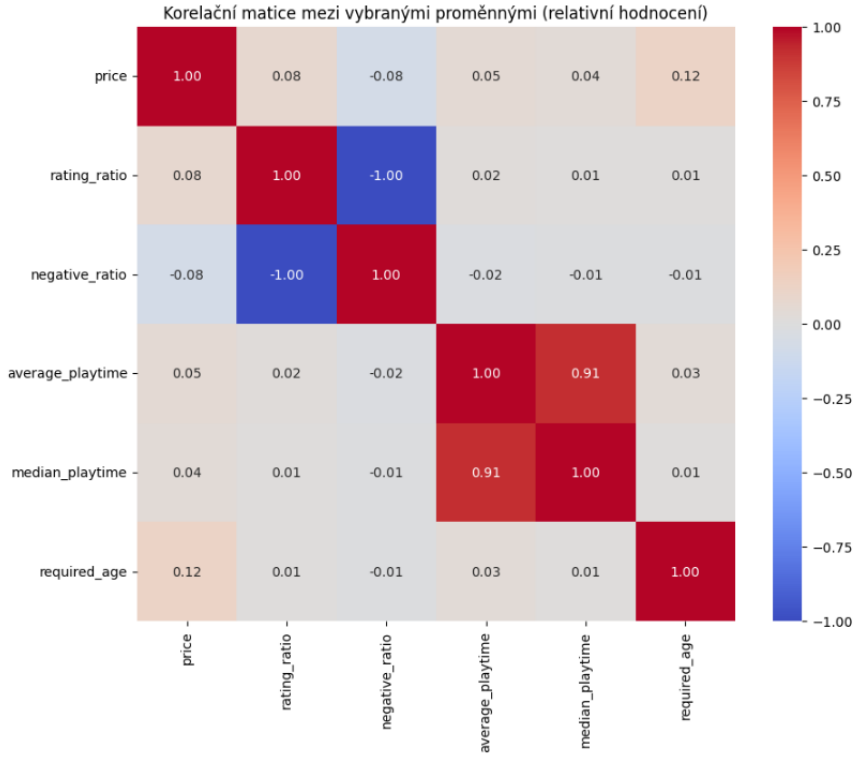
```
##Počet prázdných hodnot
steam_df.isnull().sum().sort_values(ascending=False)
```

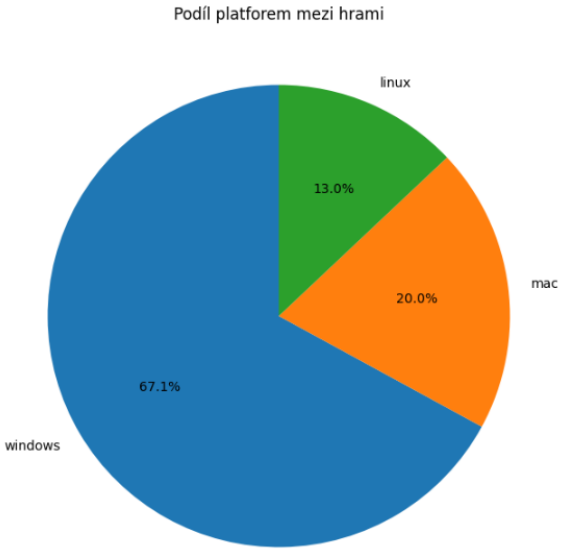
```
publisher           14
developer            1
appid                0
negative_ratings     0
linux                0
mac                  0
windows              0
rating_ratio         0
total_ratings        0
price                0
owners               0
median_playtime      0
average_playtime     0
positive_ratings     0
name                 0
achievements         0
steamspy_tags        0
genres               0
categories           0
required_age         0
platforms            0
english              0
release_date         0
negative_ratio       0
dtype: int64
```

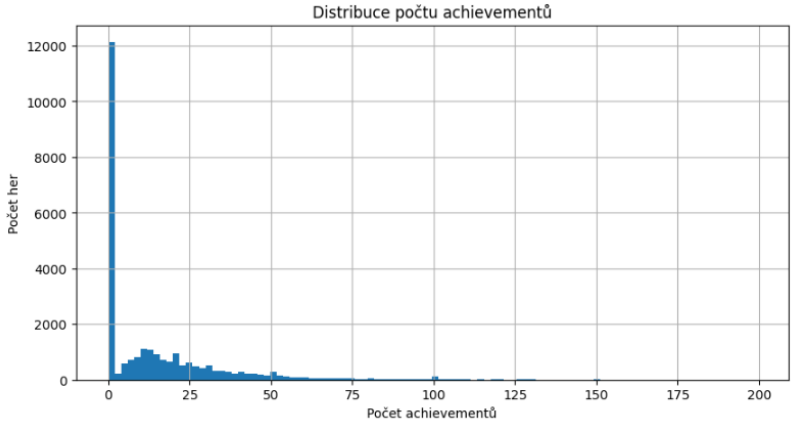





Top 10 vydavatelů
SEGA
1C Entertainment
Dagestan Technology
Choice of Games
Sekai Project
Square Enix
THQ Nordic
Ubisoft
Strategy First
Big Fish Games
publisher
0
25
50
75
100
125
150
175
200
Počet her


Korelační matice mezi vybranými proměnnými (relativní hodnocení)
price
rating_ratio
negative_ratio
average_playtime
median_playtime
required_age
1.00
0.08
-0.08
0.05
0.04
0.12
0.08
1.00
-1.00
0.02
0.01
0.01
-0.08
-1.00
1.00
-0.02
-0.01
-0.01
0.05
0.02
-0.02
1.00
0.91
0.03
0.04
0.01
-0.01
0.91
1.00
0.01
0.12
0.01
-0.01
0.03
0.01
1.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00





6.2 Tagy her

6.2.1 Základní informace

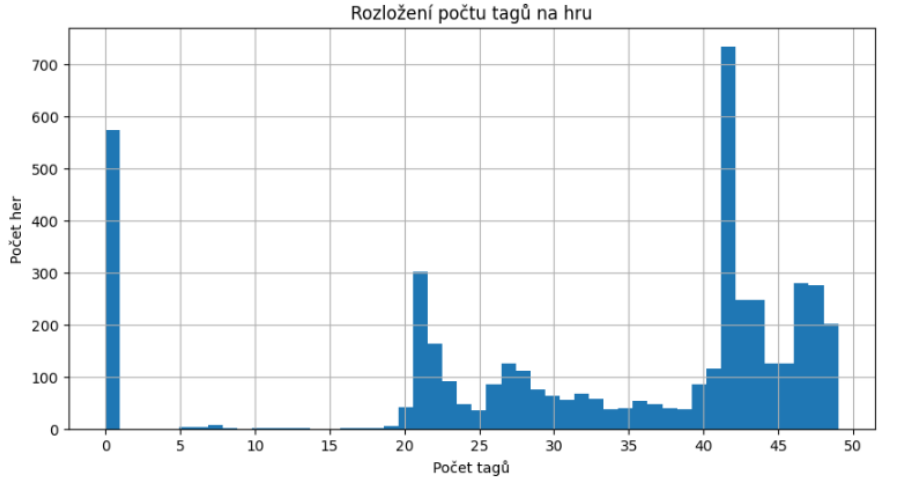
- Soubor: steamspy_tag_data.csv
- Zdroj: https://www.kaggle.com/datasets/nikdavis/steam-store-games
- Počet záznamů: 29022
- Počet sloupců: 372

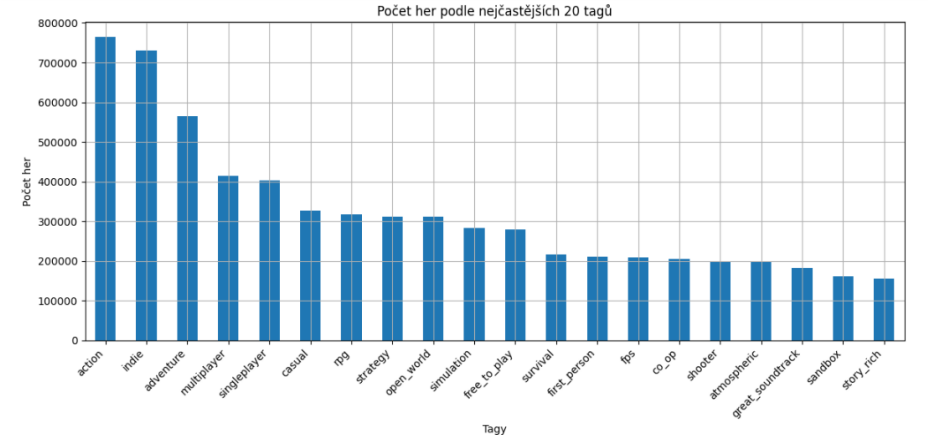
6.2.2 Sloupce

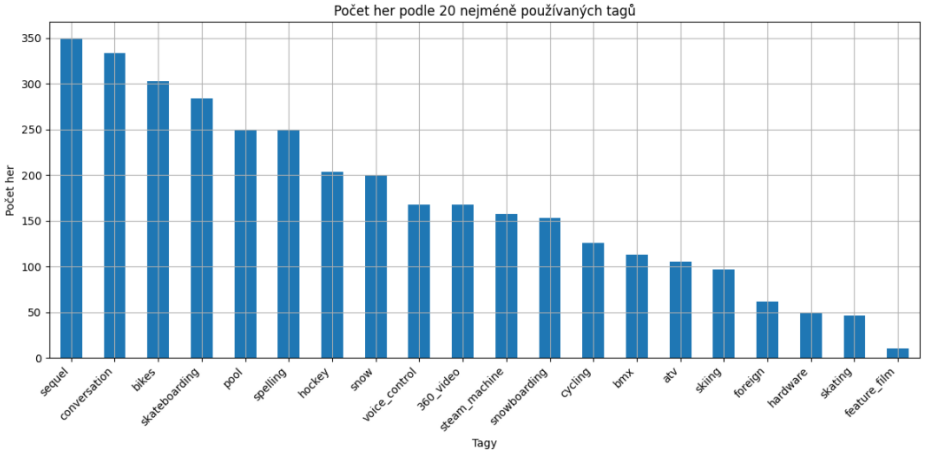
Název	Popis
appid	ID aplikace
1980s	Hra s tématem 80. let
…	…

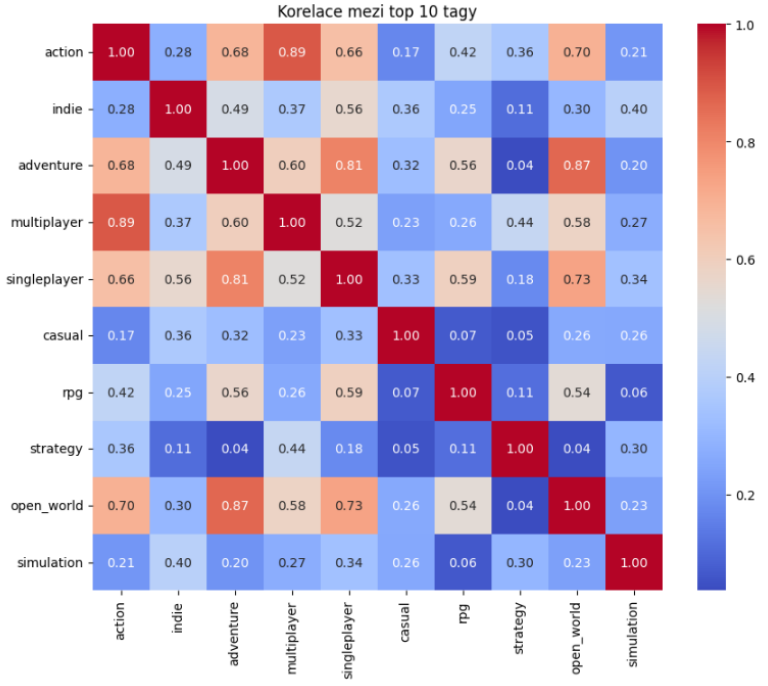
6.2.3 Grafy

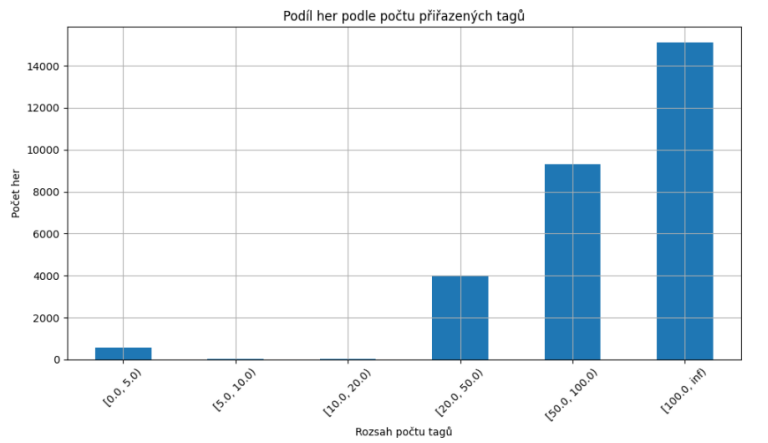
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29022 entries, 0 to 29021
Columns: 373 entries, appid to num_tags
dtypes: int64(373)
memory usage: 82.6 MB
```











6.3 HW požadavky

6.3.1 Základní informace

- Soubor: steam_requirements_data.csv
- Zdroj: https://www.kaggle.com/datasets/nikdavis/steam-store-games
- Počet záznamů: 27319
- Počet sloupců: 6

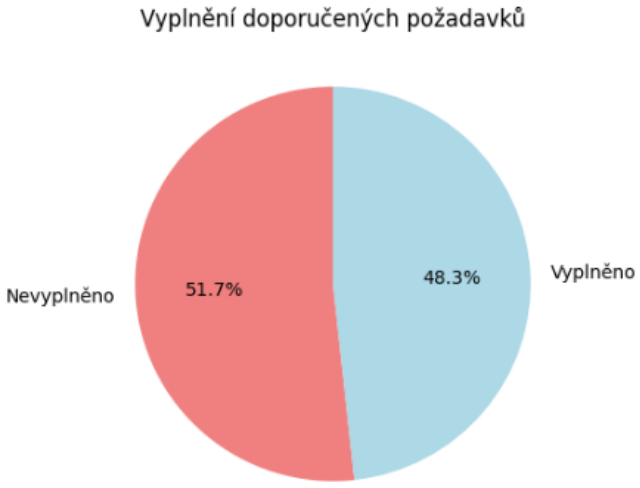
6.3.2 Sloupce

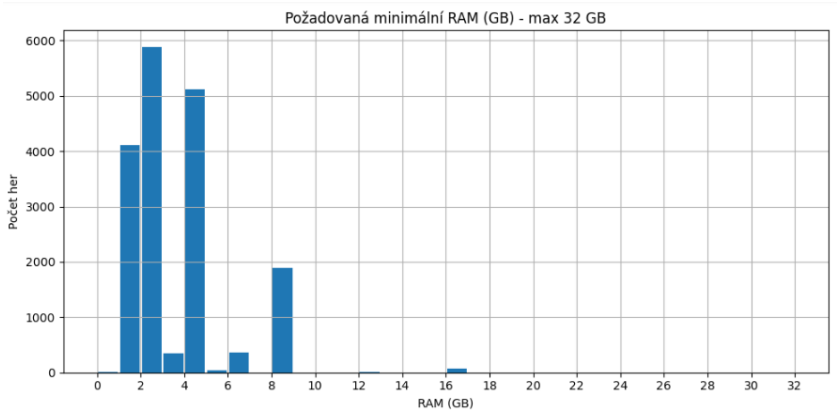
Název	Popis
steam_appid	ID aplikace
pc_requirements	Minimální požadavky pro Windows
mac_requirements	Minimální požadavky pro MAC
linux_requirements	Minimální požadavky pro Linux
minimum	Minimální HW požadavky
recommended	Doporučené HW požadavky

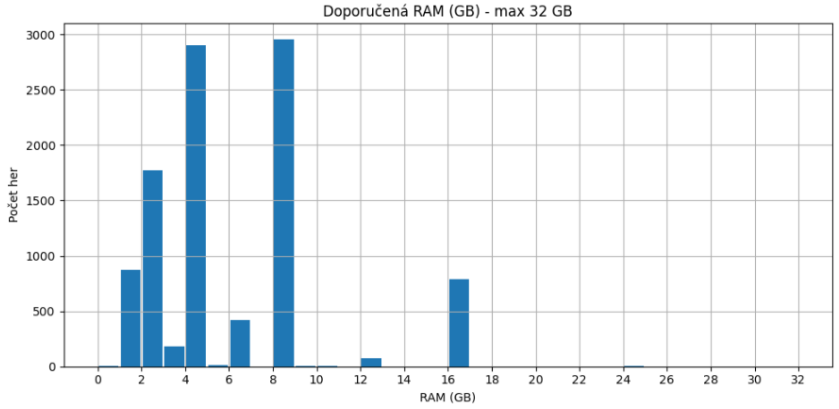
6.3.3 Grafy

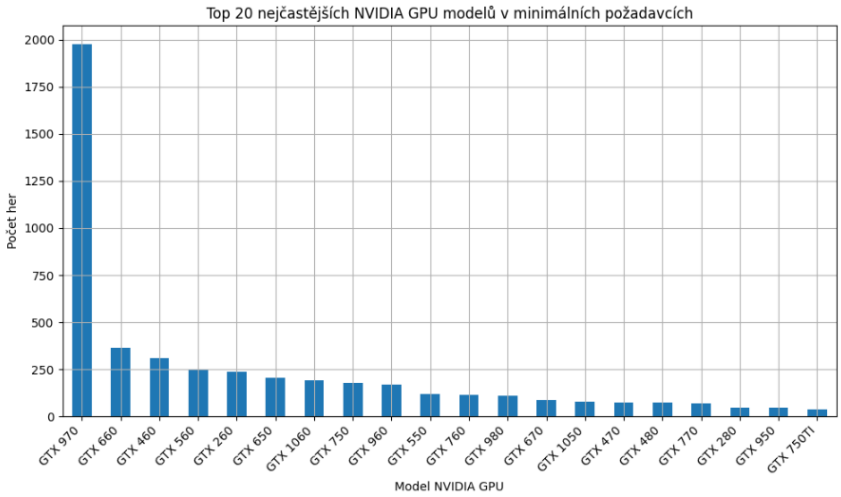
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27319 entries, 0 to 27318
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   steam_appid         27319 non-null  int64
 1   pc_requirements     27319 non-null  object
 2   mac_requirements    27319 non-null  object
 3   linux_requirements  27319 non-null  object
 4   minimum             27314 non-null  object
 5   recommended         14134 non-null  object
dtypes: int64(1), object(5)
memory usage: 747.1+ KB
```

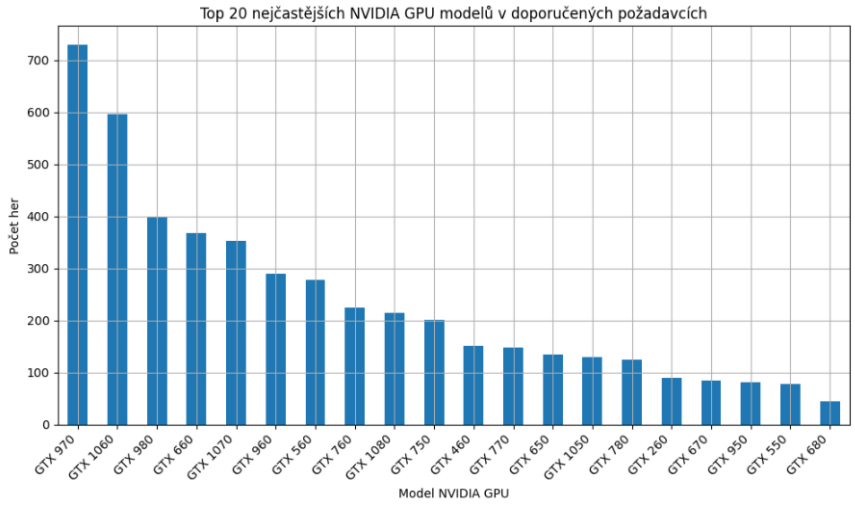
```
steam_appid               0
pc_requirements           0
mac_requirements          0
linux_requirements        0
minimum                   5
recommended           13185
dtype: int64
```

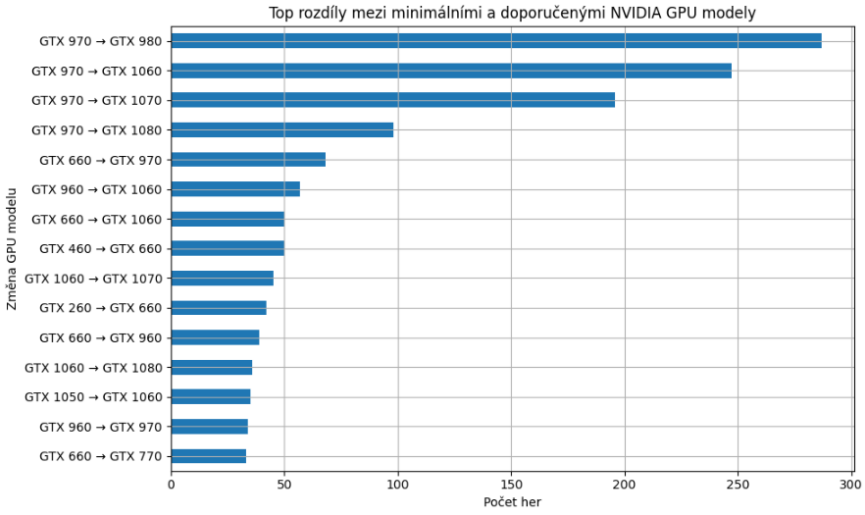





Doporučená RAM (GB) - max 32 GB
Počet her
RAM (GB)
0
500
1000
1500
2000
2500
3000
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32


Top 20 nejčastějších NVIDIA GPU modelů v minimálních požadavcích
Počet her
Model NVIDIA GPU
0
250
500
750
1000
1250
1500
1750
2000
GTX 970
GTX 660
GTX 460
GTX 560
GTX 260
GTX 650
GTX 1060
GTX 750
GTX 960
GTX 550
GTX 760
GTX 980
GTX 670
GTX 1050
GTX 470
GTX 480
GTX 770
GTX 280
GTX 950
GTX 750TI


Top 20 nejčastějších NVIDIA GPU modelů v doporučených požadavcích
Počet her
0
100
200
300
400
500
600
700
GTX 970
GTX 1060
GTX 980
GTX 660
GTX 1070
GTX 960
GTX 560
GTX 760
GTX 1080
GTX 750
GTX 460
GTX 770
GTX 650
GTX 1050
GTX 780
GTX 260
GTX 670
GTX 950
GTX 550
GTX 680
Model NVIDIA GPU


Top rozdíly mezi minimálními a doporučenými NVIDIA GPU modely
Změna GPU modelu
GTX 970 → GTX 980
GTX 970 → GTX 1060
GTX 970 → GTX 1070
GTX 970 → GTX 1080
GTX 660 → GTX 970
GTX 960 → GTX 1060
GTX 660 → GTX 1060
GTX 460 → GTX 660
GTX 1060 → GTX 1070
GTX 260 → GTX 660
GTX 660 → GTX 960
GTX 1060 → GTX 1080
GTX 1050 → GTX 1060
GTX 960 → GTX 970
GTX 660 → GTX 770
0
50
100
150
200
250
300
Počet her

7 Dotazy

7.1 Práce s daty

7.1.1 Insert

```
HSET steam:{2000000} appid 2000000 name "UHK Simulator" release_date
"2025-01-01" developer "ltrk.dev" publisher "UHK" platforms
"windows,linux" price 6.9
```

Vytvoření nové hashmapy pod klíčem `steam:{2000000}`. Ukládají se následující informace o hře:

- **appid**: 2000000
- **name**: "UHK Simulator"
- **release_date**: "2025-01-01
- **developer**: "ltrk.dev
- **publisher**: "UHK
- **platforms**: "windows,linux"
- **price**: 6.9

7.1.2 Update hodnoty

```
HSET steam:{2000000} name "Simulator UHK"
```

Aktualizace/přidání pole name v již existující hashmapě pod klíčem steam:{2000000}. Hodnota name se změní na "Simulator UHK".

7.1.3 Smazání hodnoty

```
HDEL steam:{2000000} price
```

Z hashmapy s klíčem steam:{2000000} se odstraní pole price.

7.1.4 Přejmenování klíče

```
RENAME steam:{2000000} steam:{2000001}
```

Změní název klíče ze steam:{2000000} na steam:{2000001}.

7.1.5 Nastavení expirace

```
EXPIRE steam:{2000000}_copy 60
```

Nastaví expiraci klíče steam:{2000000}_copy na **60 sekund**. Po uplynutí této doby bude klíč i s jeho obsahem automaticky odstraněn z databáze.

7.1.6 Kopírování klíče

```
COPY steam:{2000000} steam:{2000000}:backup
```

Vytvoří kopii klíče steam:{2000000} pod názvem steam:{2000000}:backup.

7.2 Indexy (RediSearch)

7.2.1 Vytvoření indexu

```
FT.CREATE idx_games ON HASH PREFIX 1 "steam:{" SCHEMA name TEXT
SORTABLE developer TEXT SORTABLE price NUMERIC SORTABLE genres TEXT
```

Vytvoření **fulltextového indexu** idx_games nad hash záznamy začínajícími prefixem steam:{. Index bude fungovat nad těmito poli v každé hash:

- name jako **textové pole**, které lze řadit
- developer jako **textové pole**, které lze řadit
- price jako **číselné pole**, které lze řadit
- genres jako **textové pole** bez možnosti řazení

7.2.2 Výpis existujících indexů

```
FT._LIST
```

Výpis všech existujících indexů.

7.2.3 Zobrazení detailů indexu

```
FT.INFO idx_games
```

Vrátí nastavení konkrétního indexu.

IDENTIFIER	ATTRIBUTE	TYPE	WEIGHT	SORTABLE
name	name	TEXT	1	×
price	price	NUMERIC		✓
genres	genres	TEXT	1	×
platforms	platforms	TEXT	1	×
steamspy_tags	steamspy_tags	TEXT	1	×
release_date	release_date	TEXT	1	✓
developer	developer	TEXT	1	✓
positive_ratings	positive_ratings	NUMERIC		✓
negative_ratings	negative_ratings	NUMERIC		✓
average_playtime	average_playtime	NUMERIC		✓

7.2.4 Smazání indexu

```
FT.DROPINDEX idx_games
```

Odstraní index idx_games.

7.2.5 Smazání indexu a dat

```
FT.DROPINDEX idx_games DD
```

Odstraní index idx_games a smaže i hodnoty klíče kterých se index týkal.

7.2.6 Úprava indexu

```
FT.ALTER 'idx_games' SCHEMA ADD release_date TEXT SORTABLE
```

Do existujícího indexu idx_games se přidá nové textové řaditelné pole release_date.

7.3 Filtrování (RediSearch)

7.3.1 Fulltextové vyhledávání v textu a řazení

```
FT.SEARCH idx_games "@name: simulator" SORTBY price DESC
```

Vyhledá v indexu idx_games všechny hry, kde pole name obsahuje slovo „simulator“. Výsledky budou seřazeny sestupně podle ceny.

Matched: 358

Doc	price	release_date	achievements	categories	appid	name	owners	publisher
steam:{553520}	49.99	2019-01-07	23	Single-player;Multi-player;Co-op;Steam Achiev...	553520	Trainz Railroad Simulator 2019	0-20000	N3V Games
steam:{434030}	44.99	2017-11-20	0	Single-player;Partial Controller Support	434030	Aerofly FS 2 Flight Simulator	20000-50000	IPACS
steam:{278550}	44.99	2015-10-21	0	Single-player;Steam Trading Cards;Includes le...	278550	A-Train 9 V4.0 : Japan Rail Simulator	0-20000	Degica
steam:{24010}	34.99	2015-09-17	1165	Single-player;Steam Achievements;Full control...	24010	Train Simulator 2019	1000000-2000000	Dovetail Games - Trains
steam:{370350}	32.99	2018-09-14	0	Single-player;VR Support;Steam Workshop;Par...	370350	LOTUS-Simulator	0-20000	Oriolus Software GmbH
steam:{794180}	30.99	2018-11-29	47	Single-player;Steam Achievements;Full control...	794180	PRO FISHING SIMULATOR	0-20000	Bigben Interactive
steam:{787860}	29.99	2018-11-19	23	Single-player;Multi-player;Co-op;Cross-Platfor...	787860	Farming Simulator 19	500000-1000000	Focus Home Interactive
steam:{585080}	29.99	2017-06-09	30	Single-player;Online Co-op;Steam Achievemen...	585080	Hunting Simulator	0-20000	Bigben Interactive

7.3.2 Fulltext vyhledávání a filtrování podle rozsahu

```
FT.SEARCH 'idx_games' '@genres:(simulator) @price:[0, 10]'
```

Vyhledá v indexu idx_games všechny hry, které splňují následující podmínky:

- V poli genres se nachází slovo „simulator“
- Hodnota pole price je v rozsahu 0–10

Matched: 3897

Doc	appid	name	release_date	english	developer	publisher	platforms	required_age
steam:{497720}	497720	Funfair Ride Simulator 3	2016-08-04	1	Pixelsplit Simulations	Pixelsplit Simulations	windows;mac;linux	0
steam:{815240}	815240	Forest Harvester Simulator	2018-03-19	1	Simulators Live	Simulators Live	windows;mac	0
steam:{502430}	502430	Roadworks - The Simulation	2016-11-30	1	VIS-Games	United Independent Entertainment GmbH	windows	0
steam:{785660}	785660	Subway Simulator	2018-03-16	1	Simulators Live	Simulators Live	windows;mac	0
steam:{323780}	323780	Fireworks Simulator	2014-11-26	1	Reality Twist GmbH	rondomedia GmbH	windows	0
steam:{889240}	889240	Loot Box Simulator	2018-07-27	1	Clickbait Studios	Clickbait Studios	windows	0
steam:{262870}	262870	Recovery Search & Rescue Simulation	2014-01-09	1	Excalibur Publishing	Excalibur Publishing	windows	0
steam:{273940}	273940	Warehouse and Logistics Simulator	2014-02-25	1	app2fun	United Independent Entertainment GmbH	windows	0

7.3.3 Fulltext vyhledávání s negací

```
FT.SEARCH 'idx_games' '+@platforms:mac @genres:(racing)
-@platforms:linux ' RETURN 3 name platforms genres
```

Vyhledá v indexu idx_games všechny hry, které:

- Obsahují „mac“ v poli platforms

- Mají v poli genres slovo „racing“
- Nevyskytují se na platformě „linux“

Matched: 123

Doc	name	platforms	genres
steam:{780160}	All Stars Racing Cup	windows;mac	Casual;Racing;Early Access
steam:{698620}	Screamer 2	windows;mac	Racing;Simulation;Sports
steam:{317080}	MotorSport Revolution	windows;mac	Racing;Simulation;Sports
steam:{324750}	StuntMANIA Reloaded	windows;mac	Adventure;Casual;Racing
steam:{448610}	Draw Rider	windows;mac	Action;Indie;Racing
steam:{311820}	RC Mini Racers	windows;mac	Action;Indie;Racing
steam:{451990}	Cranks and Goggles	windows;mac	Casual;Indie;Racing;Sports
steam:{487330}	Totally Unbalanced	windows;mac	Action;Adventure;Casual;Indie;Racing

7.3.4 Vyhledávání s OR + negace

```
FT.SEARCH 'idx_games' '@platforms:(mac|linux) -@platforms:(windows) '
RETURN 4 name platforms genres price
```

Vyhledá všechny hry, které:

- Podporují platformu mac NEBO linux
- Nepodporují platformu windows

Z výsledku zobrazí pouze hodnoty z polí name, platforms, genres a price.

Matched: 5

Doc	name	platforms	genres	price
steam:{594550}	Arma: Cold War Assault Mac/Linux	mac;linux	Violent;Action;Simulation;Strategy	3.49
steam:{805260}	PICNIC	linux	Indie	0
steam:{214630}	Call of Duty: Black Ops - Mac Edition	mac	Action	15.49
steam:{569050}	Paul Pixel - The Awakening	mac	Adventure;Indie	2.89
steam:{694180}	MobileZombie	mac	Adventure;Casual;Free to Play;Indie	0

7.3.5 Profilování vyhledávání

```
FT.PROFILE 'idx_games' SEARCH QUERY '@platforms:(mac|linux)
-@platforms:(windows)'
```

Slouží k ladění dotazu – zobrazuje rozpis toho jak Redis vyhodnocuje a optimalizuje hledání.

7.3.6 Vyhledávání s AND, řazením a limitací počtu výsledků

```
FT.SEARCH 'idx_games' '@steamspy_tags: VR @steamspy_tags: horror
@price:[0, 30]' SORTBY release_date DESC LIMIT 0 10 RETURN 5 name
release_date platforms steamspy_tags price
```

Vyhledá hry, které:

- Pole steamspy_tags obsahuje VR a horror
- Cena je od 0 do 30

Výsledky budou seřazené podle data vydání a vyhledá se pouze prvních 10. Výstup bude obsahovat hodnoty z 5 polí.

Matched: 13

Doc	release_date	name	platforms	steamspy_tags	price
steam:{1000370}	2019-01-03	SurReal Subway	windows	Indie;Horror;VR	2.89
steam:{972740}	2018-11-14	KOBOLD: Chapter I	windows	Adventure;VR;Horror	7.19
steam:{913100}	2018-10-09	RED: Lucid Nightmare	windows	Indie;Psychological Horror;VR	10.29
steam:{513270}	2017-12-21	Lockdown: Stand Alone	windows	Action;Horror;VR	7.99
steam:{700170}	2017-10-13	Deserving Life	windows	Free to Play;Horror;VR	0
steam:{707580}	2017-10-03	AFFECTED: The Manor	windows	Indie;VR;Horror	5.99
steam:{503580}	2017-09-14	Duck Season	windows	Action;Horror;VR	14.99
steam:{614710}	2017-05-02	Seance: The Unquiet (Demo 1)	windows	Indie;Horror;VR	0

7.4 Agregační funkce (RediSearch)

7.4.1 Počet vydaných her a jejich hodnocení jednotlivých vývojářů

```
FT.AGGREGATE idx_games "*"
  GROUPBY 1 @developer
  REDUCE COUNT 0 AS game_count
  REDUCE SUM 1 @positive_ratings AS total_positive
  REDUCE SUM 1 @negative_ratings AS total_negative
  APPLY "(@total_positive / (@total_positive + @total_negative)) * 100"
AS positive_ratio
  FILTER "@game_count > 0"
  SORTBY 2 @game_count DESC
```

Zjistí hodnocení her podle vývojáře

- Seskupí podle vývojáře
- Vypočítá počet her pro vývojáře
- Sečte pozitivní hodnocení
- Sečte negativní hodnocení
- Spočítá procentuální hodnocení
- Zobrazí pouze vývojáře s nějakou hrou
- Seřadí podle počtu her

Matched: 17018

developer	game_count	total_positive	total_negative	positive_ratio
choice of games	94	4804	663	87.8726906896
koei tecmo games co., ltd.	72	40005	17331	69.7729175387
ripknot systems	62	811	541	59.9852071006
laush dmitriy sergeevich	51	506	273	64.9550706033
nikita \"ghost_rus\"	50	1034	585	63.8665843113
dexion games	45	236	567	29.3897882939
rewindapp	43	1498	710	67.8442028986
hosted games	42	1544	225	87.2809496891

7.4.2 Počet vydaných her jednotlivých vývojářů za každý rok a jejich průměrná cena

```
FT.AGGREGATE idx_games "*"
  APPLY "substr(@release_date, 0, 4)" AS release_year
  GROUPBY 2 @release_year @developer
  REDUCE COUNT 0 AS game_count
  REDUCE AVG 1 @price AS avg_price
  SORTBY 4 @release_year ASC @game_count DESC
```

Zjistí statistky vydaných her podle roku a vývojáře

- Získá rok vydání z datumu
- Seskupí podle roku a vývojáře
- Spočítá hry
- Vypočítá průměrnou cenu
- Seřadí vzestupně podle roku a sestupně podle ceny

Matched: 21691

release_year	developer	game_count	avg_price
1997	stainless games ltd	1	5.99
1998	valve	1	7.19
1999	gearbox software	1	3.99
1999	valve	1	3.99
2000	valve	2	5.59
2001	spiderweb software	1	0
2001	valve	1	3.99
2001	gearbox software	1	3.99

7.4.3 Průměrný herní čas podle vývojáře

```
FT.AGGREGATE idx_games "*"
  GROUPBY 1 @developer
  REDUCE AVG 1 @average_playtime AS avg_playtime_minutes
  APPLY "@avg_playtime_minutes / 60" AS avg_playtime_hours
  SORTBY 2 @avg_playtime_hours DESC
```

Zjistí průměrnou dobu hraní podle vývojáře

- Seskupí hry podle vývojáře
- Vypočítá průměr v minutách
- Převede průměr na hodiny
- Seřadí sestupně podle hodin

Matched: 17018

developer	avg_playtime_minutes	avg_playtime_hours
manuel pazos;daniel celem\xc3\xadn	190625	3177.08333333
sebastian krzyszkowiak;konrad burandt;pawe\xc5\x82 radej	95242	1587.36666667
portalarium	54618	910.3
laminar research	44169	736.15
\xe6\x89\x98\xe6\x9b\xb4\xe7\x9a\x84\xe4\xbf\xae\xe7\xbd\x97	43632	727.2
screeps	38805	646.75
construct studio	36029	600.483333333
wonderstruck	27857	464.283333333
delphine software	27375	456.25

7.4.4 Počet her podle rozsahu počtu majitelů

```
FT.AGGREGATE idx_games "*" GROUPBY 1 @owners REDUCE COUNT 0 AS
total_games SORTBY 2 @total_games DESC
```

Ukáže, kolik her spadá do jednotlivých rozsahů počtu majitelů

- Seskupí podle počtu zakoupení
- Spočítá počet výskytů ve skupině
- Setřídí sestupně podle počtu her

Matched: 13

owners	total_games
0-20000	18596
20000-50000	3059
50000-100000	1695
100000-200000	1386
200000-500000	1272
500000-1000000	513
1000000-2000000	288
2000000-5000000	193
5000000-10000000	46

7.4.5 Nejvyšší cena hry podle vývojáře

```
FT.AGGREGATE idx_games "*"
  GROUPBY 1 @developer
  REDUCE MAX 1 @price AS max_price
  SORTBY 2 @max_price DESC
```

Pro každého vývojáře dohledá jeho nejdražší hru

- Seskupí hry podle vývojáře
- Najde maximum
- Seřadí sestupně

Matched: 17018

developer	max_price
suomen kuljetusturva oy	421.99
yoyo games ltd.	303.99
sidefx	209.99
pixel wonder	154.99
apeirogon games	154.99
first_ukrainian	154.99
capt. mccay soft	154.99
capt mccay soft	154.99
3dflow srl	154.99

7.4.6 Profilování agregace

```
FT.PROFILE  'idx_games'  AGGREGATE QUERY "*"
  GROUPBY 1 @developer
  REDUCE MAX 1 @price AS max_price
  SORTBY 2 @max_price DESC
```

Profilování AGGREGATE dotazu. Vrátí informace o tom, jak dlouho dotaz trval, nebo jak dlouho trvala jednotlivá část dotazu.

7.5 Cluster příkazy

7.5.1 Informace o clusteru

```
CLUSTER INFO
```

Zobrazí informace o clusteru jako celku. Signalizuje například status cluster (ok/fail), případně i chybějící jednotlivé sloty.

CLUSTER INFO

```
cluster_state:ok
cluster_slots_assigned:16384
cluster_slots_ok:16384
cluster_slots_pfail:0
cluster_slots_fail:0
cluster_known_nodes:6
cluster_size:3
cluster_current_epoch:6
cluster_my_epoch:1
cluster_stats_messages_ping_sent:34854
cluster_stats_messages_pong_sent:34798
cluster_stats_messages_sent:69652
cluster_stats_messages_ping_received:34793
cluster_stats_messages_pong_received:34854
cluster_stats_messages_meet_received:5
cluster_stats_messages_received:69652
total_cluster_links_buffer_limit_exceeded:0
```

7.5.2 Uzly clusteru

```
CLUSTER NODES
```

Vypíše seznam nodů v clusteru, u každého uvede, zda se jedná o master nebo slave a jaký je jeho status.

```
CLUSTER NODES

10888277ef65d1a59495f60a702b11b925c61774 172.28.0.12:6379@16379 myself,master - 0 0 2 connected 5461-10922
4190de5fe00421376a25e2674fd532e51cd088eb 172.28.0.11:6379@16379 master - 0 1747755430505 1 connected 0-5460
7e10af2729f39199674f862beb98fcd53e215497 172.28.0.14:6379@16379 slave 467ce99e7459a68313f9170d07193156547829cb 0 1747755430000 3 connected
6ff7ca8dc6222d07d446a0349d095f27a60a9fac 172.28.0.15:6379@16379 slave 4190de5fe00421376a25e2674fd532e51cd088eb 0 1747755430303 1 connected
467ce99e7459a68313f9170d07193156547829cb 172.28.0.13:6379@16379 master - 0 1747755431310 3 connected 10923-16383
ded1d75dedc258d03dd75700ddd4699754512eb7 172.28.0.16:6379@16379 slave 10888277ef65d1a59495f60a702b11b925c61774 0 1747755430000 2 connected
```

7.5.3 Identifikace slotu klíče

CLUSTER KEYSLOT steam:{1000370}

Dotaz vrátí příslušný slot, do kterého klíč náleží.

```
CLUSTER KEYSLOT steam:{1000370}

(integer) 6339
```

7.5.4 Zobrazení běžící konfigurace

CONFIG GET cluster*

Příkaz vypíše aktuální nastavení clusteru.

```
CONFIG GET cluster*

1) "cluster-announce-hostname"
2) ""
3) "cluster-config-file"
4) "nodes.conf"
5) "cluster-require-full-coverage"
6) "no"
7) "cluster-replica-validity-factor"
8) "10"
9) "cluster-announce-ip"
10) ""
11) "cluster-replica-no-failover"
12) "no"
13) "cluster-allow-replica-migration"
14) "yes"
15) "cluster-slave-validity-factor"
16) "10"
17) "cluster-slave-no-failover"
18) "no"
19) "cluster-enabled"
20) "yes"
21) "cluster-link-sendbuf-limit"
22) "0"
23) "cluster-announce-port"
24) "0"
25) "cluster-preferred-endpoint-type"
26) "ip"
27) "cluster-announce-human-nodename"
28) ""
29) "cluster-migration-barrier"
30) "1"
31) "cluster-announce-bus-port"
32) "0"
33) "cluster-announce-tls-port"
34) "0"
35) "cluster-port"
36) "0"
37) "cluster-allow-pubsubshard-when-down"
38) "yes"
39) "cluster-node-timeout"
40) "5000"
41) "cluster-allow-reads-when-down"
42) "yes"
43) "cluster-compatibility-sample-ratio"
44) "0"
```

7.5.5 Přesunutí slotu v rámci clusteru

CLUSTER SETSLOT <SLOT> NODE <NODE ID>

Příkaz přiřadí slot jiné node. Vhodné pro vyvažování zátěže nebo přesun dat mezi uzly.

7.5.6 Status replikace

INFO REPLICATION

Zobrazí status replikace aktuální node.

7.6 Simulace výpadku

7.6.1 Výpadek master nody

1. Vypnutí některé master nody – „docker compose stop redis-3“
2. Proběhne detekce výpadku

```
redis-6-1 | 1:S 18 May 2025 19:17:10.590 * Marking node
62fe3c19e3914aa6080fca9e28f699355c249bfb () as failing (quorum reached).
redis-1-1 | 1:M 18 May 2025 19:17:10.590 * Marking node
62fe3c19e3914aa6080fca9e28f699355c249bfb () as failing (quorum reached).
redis-5-1 | 1:S 18 May 2025 19:17:10.591 * FAIL message received from
8e5eeed88769f64cc0da48c0d36edff2a55fe7bc () about
62fe3c19e3914aa6080fca9e28f699355c249bfb ()
redis-2-1 | 1:M 18 May 2025 19:17:10.591 * FAIL message received from
8e5eeed88769f64cc0da48c0d36edff2a55fe7bc () about
62fe3c19e3914aa6080fca9e28f699355c249bfb ()
redis-4-1 | 1:S 18 May 2025 19:17:10.592 * FAIL message received from
8e5eeed88769f64cc0da48c0d36edff2a55fe7bc () about
62fe3c19e3914aa6080fca9e28f699355c249bfb ()
```

3. Proběhne volba nového masteru

```
redis-4-1 | 1:S 18 May 2025 19:17:10.604 * Start of election delayed for 744
milliseconds (rank #0, offset 168482310).
redis-4-1 | 1:S 18 May 2025 19:17:11.409 * Starting a failover election for epoch
7.
redis-1-1 | 1:M 18 May 2025 19:17:11.410 * Failover auth granted to
e026d0960457a20f7a6f457e3ded15edef92acbc () for epoch 7
redis-2-1 | 1:M 18 May 2025 19:17:11.410 * Failover auth granted to
e026d0960457a20f7a6f457e3ded15edef92acbc () for epoch 7
redis-4-1 | 1:S 18 May 2025 19:17:11.411 * Failover election won: I'm the new
master.
```

4. Cluster obnoví funkční stav

7.6.2 Výpadek shardů

1. Vypnutí posledního node clusteru – docker compose stop redis-4
2. Detekce výpadku

```
redis-2-1 | 1:M 18 May 2025 19:25:59.541 * Marking node
e026d0960457a20f7a6f457e3ded15edef92acbc () as failing (quorum reached).
redis-6-1 | 1:S 18 May 2025 19:25:59.542 * FAIL message received from
8317c13f4a52ca8b7f87dc164ec6d42873573d12 () about
e026d0960457a20f7a6f457e3ded15edef92acbc ()
redis-1-1 | 1:M 18 May 2025 19:25:59.542 * Marking node
e026d0960457a20f7a6f457e3ded15edef92acbc () as failing (quorum reached).
redis-5-1 | 1:S 18 May 2025 19:25:59.542 * FAIL message received from
8317c13f4a52ca8b7f87dc164ec6d42873573d12 () about
e026d0960457a20f7a6f457e3ded15edef92acbc ()
```

3. Neúspěšné pokusy o připojení nového masteru

```
redis-1-1 | [0x721349c0a180 172.28.0.14:6379 Connecting]Error on connect:
redis-1-1 | [0x721349c0a180 172.28.0.14:6379 Connecting]Switching state to
Connecting
redis-2-1 | [0x73c553c0a180 172.28.0.14:6379 Connecting]Error on connect:
redis-2-1 | [0x73c553c0a180 172.28.0.14:6379 Connecting]Switching state to
Connecting
```

4. Načtení hodnot z ostatních shardů (HGETALL steam:{1000370}) proběhne, cluster zůstal částečně dostupný

```
1) "appid"
2) "1000370"
3) "name"
4) "SurReal Subway"
5) "release_date"
6) "2019-01-03"
7) "english"
8) "1"
9) "developer"
10) "LFiO Studio"
11) "publisher"
12) "LFiO Studio"
13) "platforms"
```

Závěr

Tato práce demonstruje nasazení Redis Clusteru v prostředí Dockeru. Podařilo se vytvořit funkční a zabezpečené řešení, které využívá všechny klíčové vlastnosti Redis Clusteru – vysokou dostupnost, horizontální škálování, replikaci dat a zabezpečení pomocí TLS a ACL. Důkaz byl kladen i na přehlednou definici služeb pro docker compose a automatické nasazení a spuštění.

Práce také seznamuje čtenáře také s prací s modulem RediSearch, díky kterému je možné provádět složitější dotazování s využitím fulltextového vyhledávání a možností agregačních funkcí. Použití modulu RediSearch demonstruje praktické použití Redisu mimo běžné použití jako databáze klíč-hodnota.

Díky zařazeným případovým studiím je patrné, že technologie Redis má své pevné místo i v náročných provozech s požadavky na nízkou latenci a vysokou spolehlivost.

Práce se sice nezaměřuje na produkční nasazení, nabízí ale základ pro další rozvoj práce s Redis Clusterem. Části týkající se produkčního nasazení a s tím spojená třeba geografická separace zůstávají otevřené k dalšímu zkoumání. I přes to je dosažené řešení funkční, přehledné a lze ho využít jako výchozí bod pro další experimentování.

Zdroje

ACL | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/operate/oss_and_stack/management/security/acl/

Administration Overview | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/administration/overview/

Aggregation | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/query/aggregation/

Aggregations | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/advanced-concepts/aggregations/

AGPLv3. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/blog/agplv3/

Axis Bank. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/customers/axis-bank/

Basic Constructs | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/basic-constructs/

BioCatch Relies on Redis Enterprise to Protect 70 Million Users with Game-Changing Behavioral Biometrics. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/press/biocatch-relies-redis-enterprise-protect-70-million-users-game-changing-behavioral-biometrics/

BioCatch. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/customers/biocatch/

Cluster vs. Sentinel. [online]. [cit. 20. 5. 2025]. Dostupné z: https://medium.com/@khandelwal.praful/understanding-redis-high-availability-cluster-vs-sentinel-420ecaac3236

Combined | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/query/combined/

Commands — redis-py 5.0.0 documentation. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.readthedocs.io/en/stable/commands.html

Connections — redis-py 5.0.0 documentation. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.readthedocs.io/en/stable/connections.html

Control startup order in Compose. [online]. [cit. 20. 5. 2025]. Dostupné z: https://docs.docker.com/compose/startup-order/

Deutsche Börse. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/customers/deutsche-borse/

Encryption at rest | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/operate/rc/security/encryption-at-rest/

Exact Match | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/query/exact-match

Field and Type Options | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/basic-constructs/field-and-type-options/

Full Text | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/query/full-text/

Indexing | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/indexing/

JSON | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/operate/oss_and_stack/stack-with-enterprise/json/

MongoDB vs Redis: Which Document Database is Right for You? [online]. [cit. 20. 5. 2025]. Dostupné z: https://www.linkedin.com/pulse/mongodb-vs-redis-document-database-yaniv-masler-szuhf

Range | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/query/range/

Redis - Official Image. [online]. [cit. 20. 5. 2025]. Dostupné z: https://hub.docker.com/_/redis

Redis essentials. Birmingham: Packt Publishing, 2015. ISBN 978-1-78439-245-1.

Redis Gears. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/learn/howtos/solutions/caching-architecture/common-caching/redis-gears

Redis persistence | Docs. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/operate/oss_and_stack/management/persistence/

Redis Query Engine | Docs. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/

Redis replication | Docs. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/operate/oss_and_stack/management/replication/

Redis security. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/operate/oss_and_stack/management/security/

Redis TimeSeries | A NoSQL Time Series Database. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/timeseries/

RedisBloom | Bloom & Cuckoo Filtering for Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/probabilistic/

Scale with Redis Cluster. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/operate/oss_and_stack/management/scaling/

Schema Definition | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/basic-constructs/schema-definition/

Security. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/operate/oss_and_stack/management/security/

Sorting | Docs - Redis. [online]. [cit. 20. 5. 2025]. Dostupné z: https://redis.io/docs/latest/develop/interact/search-and-query/advanced-concepts/sorting/

Steam Store Games. [online]. [cit. 20. 5. 2025]. Dostupné z: https://www.kaggle.com/datasets/nikdavis/steam-store-games

Understanding CAP Theorem in Distributed Systems. [online]. [cit. 20. 5. 2025]. Dostupné z: https://devopsideas.com/understanding-cap-theorem-in-distributed-systems/

Nástroje

Visual Studio Code: https://code.visualstudio.com

Proxmox Virtual Environment: https://www.proxmox.com/en/products/proxmox-virtual-environment/overview

Docker: https://www.docker.com

JupyterLab: https://jupyter.org/try-jupyter/lab/

RedisInsight: https://redis.io/insight/

Debian 12 (Bookworm): https://www.debian.org/releases/bookworm/

NumPy: https://numpy.org

pandas: https://pandas.pydata.org

Matplotlib: https://matplotlib.org

Git: https://git-scm.com

Přílohy

- .github
  - Workflows
    - docker.yml – Konfigurace pro Github Actions
- Data
  - D1.ipynb – Jupyter notebook pro steam.csv
  - D2.ipynb – Jupyter notebook pro steamspy_tag_data.csv
  - D3.ipynb – Jupyter notebook pro steam_requirements_data.csv
  - steam.csv – Údaje o hrách
  - steamspy_tag_data.csv – Tagy pro jednotlivé hry
  - steam_requirements_data.csv – Minimální a doporučené požadavky
- Dotazy
  - dotazy.md – Markdown soubor s dotazy
- Funkcni_reseni
  - config
    - access-rights.acl – konfigurace Redis ACL
    - redis.conf – konfigurace Redisu
  - docker
    - init.Dockerfile – Dockerfile init image
    - rcproxy.Dockerfile – Dockerfile Redis Cluster Proxy
    - redis.Dockerfile – Dockerfile Redis 8
    - stack.Dockerfile – Dockerfile pro Redis Stack 7
  - init
    - init-cluster.sh – Inicializační script
    - load_data.py – Script pro načtení počátečních dat
    - redis_client.py – Pomocní script pro získání připojení
    - requirements.txt – Seznam závislostí Python scriptů
  - common-services.yml – Compose pro předpisy services
  - compose.yml – Docker Compose řešení
  - delete-data.sh – Script pro vymazání dat a opětovnou inicializaci
  - start.sh – Script pro spuštění