Studentská Konference Fakulty Mechatroniky, informatiky a mezioborových studií
2. červen 2015, Liberec, Česká republika

Škálovatelný distribuovaný systém pro detekci podobných dokumentů

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Ing. Ondřej Smola
Ing. Jindřich Žďánský, Ph.D.
FM, ITE

Abstract

The main goal of this thesis is to describe the design and the implementation of the distributed system for detection of similar documents. The implemented solution allows near real time search has a persistent history and tunable performance based on number of documents in the history.

Motivace

Detekce podobnosti je důležitá například u dokumentů z mnoha různých zdrojů, u kterých je značná pravděpodobnost, že byl původní originální článek lehce upraven a použit v mnoha jiných zdrojích (regionální noviny, webové články).

Cíl

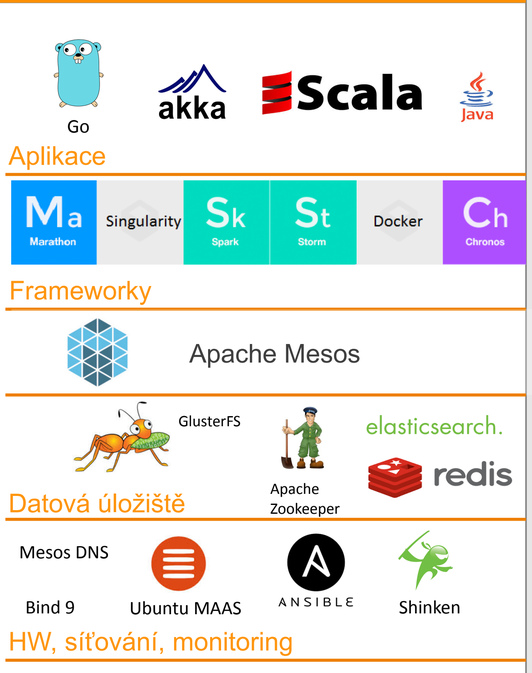
Cílem práce bylo vytvořit online systém pro detekci podobných dokumentů. Mezi hlavní požadavky na tento systém patří jeho vysoká dostupnost v případě výpadku strojů clusteru, perzistentní historie a konfigurovatelná škálovatelnost podle očekávaného množství dokumentů v historii.

Architektura

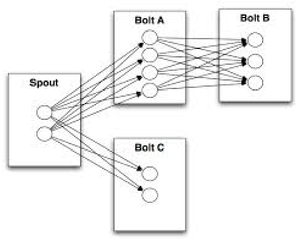
NanoGrid

NanoGrid je distribuovaný systém, vytvořený na Ústavu ITE. Systém je rozdělen na 5 vrstev a jeho aktuální výpočetní kapacita je 44 jader, 170GB paměti a 3TB replikované úložné kapacity. Hlavním úkolem systému NanoGridu je poskytování obecné výpočetní vrstvy, která umožňuje běh celé řady aplikací. Mezi aktuálně podporované distribuované úlohy patří:

- MapReduce úlohy
- Realtime analytické úlohy
- Provoz vysoce dostupných služeb
- Distribuované datové úložiště
- Micro-batching úlohy
- Strojové učení, grafové úlohy
- Distribuované databázové dotazy
- Vlastní implementace plánovačů



Apache Storm



Rámec Apache Storm je využíván pro distribuovanou detekci duplicitních dokumentů. Výpočet je popsán pomocí topologie, která se skládá z komponent, označovaných jako spout a bolt. Spout má za úkol načítat data do topologie a bolt popisuje výpočetní krok. Mezi hlavní výhody rámce Storm patří možnost jednoduchého škálování topologie, odolnost a výkonnost. Hlavním programovacím jazykem je Java.

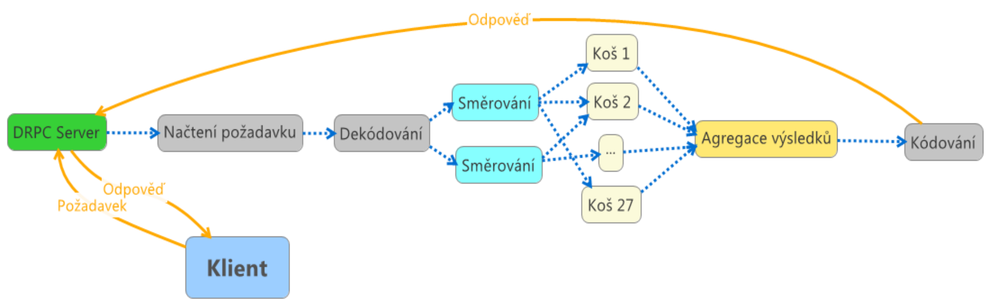
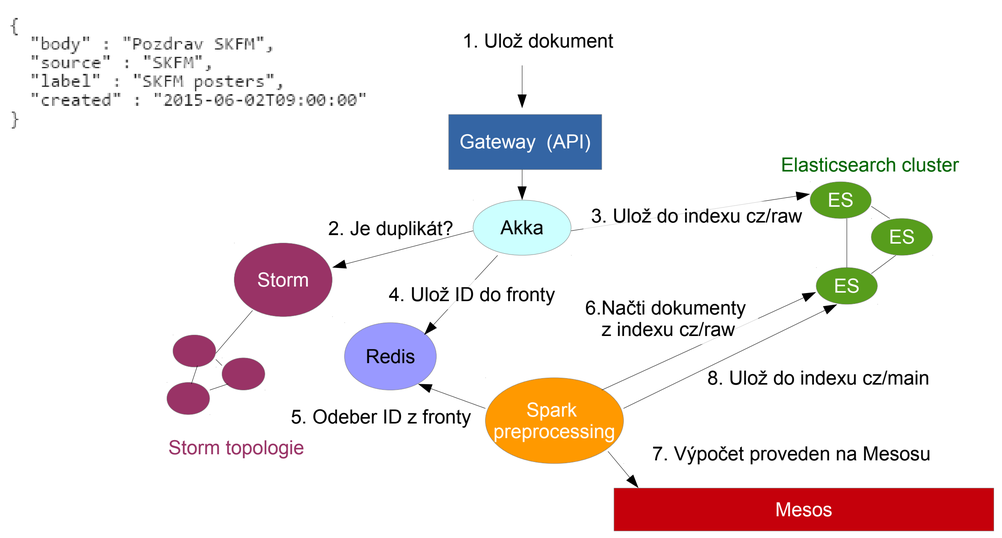


Schéma 1: Topologie pro detekci podobných dokumentů

Postup zpracování požadavku

1) Klient kontaktuje DRPC server s otiskem dokumentu
2) Otisk dokumentu je reprezentován 256 bity - MinHash z tokenů text
3) Otisky jsou v topologii jsou rozděleny do 27 košů podle jejich bitové kardinality
4) Požadavek je směrován do košů podle požadované minimální podobnosti
5) Otisk je uložen do koše se stejnou kardinalitou
6) Podobnost dokumentů určena podle Jaccardova koeficientu otisků
7) Vrácení výsledků z košů
8) Vybrán nejlepší výsledek
9) Odpověď klientovi přes DRPC server

Nasazení



Obrázek 2: Schéma zpracování dokumentů pro tvorbu jazykového modelu

Systém je použit pro detekci podobných dokumentů uvnitř pracovní fronty pro předzpracování ukládaných dokumentů. Díky online detekci podobných dokumentů a online předzpracování, je možné nově příchozí dokumenty použít pro tvorbu jazykového modelu během několika sekund od dotazu na uložení. Výkonnost tohoto řešení je limitována rychlostí detekce podobných dokumentů. Vliv velikosti historie na propustnost řešení je zobrazen v tabulce 1.

Velikost historie	Požadavků/s
270 000	200
2 700 000	50
27 000 000	20

Tabulka 1: Vliv velikosti historie na propustnost

Testy

Systém byl otestován na přibližně 800 MB dat (deníky, webové příspěvky). Při nastavení minimální podobnosti na 94%, bylo výsledkem přibližně 140 MB textu obsahujícího velmi podobné dokumenty. Jelikož je srovnávání podobnosti velice subjektivní věc, není možné jednoduše vyhodnotit celkovou úspěšnost. I přesto podle manuální kontroly systém velice dobře detekuje běžné úpravy používané při kopírování: vynechání vět, změna pořadí slov, pravopisné chyby, vložené věty nebo i kusy vloženého HTML kódu. Odolnost systému byla otestována náhodným vypnutím strojů a systém se vždy dokázal obnovit do stavu plné funkcionality, s pouze krátkodobým přerušením činnosti. Problematické je pouze zálohování historie, které je nastaveno na jednou za minutu. Z toho vyplývá, že v případě výpadku systém může ztratit až poslední minutu historie. Periodu zálohování je možné konfigurovat a interval určuje poměr mezi konzistencí a výkonem řešení.

Závěr

Vytvořený systém umožňuje online detekovat podobné dokumenty a jeho propustnost přesahuje plánované zatížení. Řešení je odolné vůči výpadku strojů a má plně konfigurovatelnou škálovatelnost podle množství přidělených zdrojů, délky ukládané historie a intervalu zálohování. Aktuálně je systém použit uvnitř pracovní fronty pro předzpracování dokumentů, z kterých je následně vytvářen jazykový model.

Reference

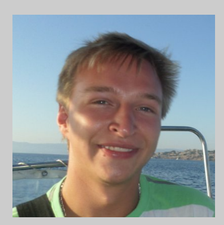
ANDERSON, Quinton. Storm real-time processing cookbook. Birmingham, England: Packt Publishing, 2013. ISBN 978-1-78216-443-2.

SHRIVASTAVA, A. & Li, P. (2014), In Defense of MinHash Over SimHash, in 'Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)' .

CARLSON, Josiah L. Redis in action. Shelter Island, NY: Manning, 2013. ISBN 9781617290855.

Kontakt

Ing. Ondřej Smola
ondrej.smola@tul.cz



Tato práce byla podpořena z projektu Studentské grantové soutěže (SGS) na Technické univerzitě v Liberci v roce 2015.