

MASARYKOVA UNIVERZITA  
FAKULTA INFORMATIKY



# Distributed Complex Event Processing

DIPLOMOVÁ PRÁCE

**Bc. Tomáš Skopal**

Brno, jaro 2016



*Místo tohoto listu vložte kopie oficiálního podepsaného zadání práce a prohlášení autora školního díla.*



## **Prohlášení**

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Bc. Tomáš Skopal

**Vedoucí práce:** RNDr. Filip Nguyen

## **Shrnutí**

Goal of this thesis is to develop a Peer to Peer algorithm for distributed Event Pattern matching.. The application should be able to run any number of processing nodes. For the needs of this thesis, example of 4 nodes will be sufficient.

## **Klíčová slova**

keyword1, keyword2, ...





# Obsah

1	Úvod . . . . .	1
2	<b>Zpracování událostí</b> . . . . .	3
2.1	CEP . . . . .	3
2.2	Událost v CEP . . . . .	4
2.3	Distribuované CEP . . . . .	4
3	<b>Nástroje pro distribuované zpracování událostí</b> . . . . .	5
3.1	Apache Samsa . . . . .	5
3.2	Apache Storm . . . . .	5
3.3	CAVE . . . . .	5
4	<b>Vytvoření clusteru v rámci sítě</b> . . . . .	7
4.1	Motivace . . . . .	7
4.2	Technologie . . . . .	7
4.2.1	Apache Maven . . . . .	7
4.2.2	Apache Kafka . . . . .	7
4.2.3	Apache ZooKeeper . . . . .	7
4.2.4	Esper . . . . .	7
4.3	Události v systému . . . . .	7
4.3.1	Hrubozrné . . . . .	8
4.3.2	Jemnozrné . . . . .	8
4.4	Konfigurace . . . . .	8
4.5	Implementace . . . . .	8
4.5.1	Architektura aplikace . . . . .	8
4.5.2	Iniciální spuštění . . . . .	8
4.5.3	Přechody mezi stavy . . . . .	8
4.5.4	Esper pravidla . . . . .	8
4.6	Známa omezení . . . . .	9
5	<b>Závěr</b> . . . . .	11



# 1 Úvod

Cílem této práce je vytvoření ...



## 2 Zpracování událostí

Se zvyšujícím se počtem zařízení, která jsou schopna produkovat data, se zvyšuje potřeba tato data analyzovat. Běžně rozšířeným způsobem je zpracování dat dávkově. Tedy, data se uloží a ve vhodnou dobu, typicky v noci, se analyzují.

Pokud však uvažujeme reálný provoz na síti, který se dnes v centrálních uzlech pohybuje okolo  $1Tb/s$ , je dávkové zpracování nereálné. Potřebujeme data analyzovat za běhu (angl. real time).

Jednotkou zpracování dat je událost (angl. event). Událost je základním pojmem používaným v oblasti zpracování událostí. Je definována jako objekt, který reprezentuje záznam o aktivitě v daném systému. Událost může mít vlastnosti. Typickým příkladem takové vlastnosti je čas vzniku události. [1] Jednoduchým příkladem události může být paket. Je to datová schránka, která obsahuje informace, které můžeme analyzovat. Samostatný paket nemá téměř žádnou vypovídající hodnotu, kdežto proud paketů je základem Internetu.

Takový proud událostí skrývá množství dat, která je možné získat až při komplexní analýze, která zohledňuje více událostí v řadě. To nazýváme *komplexní zpracování dat* (angl. *complex event processing* neboli CEP)

### 2.1 CEP

Je těžké shrnout celý vědní obor pod jednu všeobíhající definici. David Luckham ve své knize THE POWER OF EVENTS: AN INTRODUCTION TO COMPLEX EVENT PROCESSING IN DISTRIBUTED ENTERPRISE SYSTEMS [1] říká, že CEP je soubor technik a nástrojů, které pomáhají k pochopení a kontrole událostmi řízených systémů.

### 2.2 Událost v CEP

### 2.3 Distribuované CEP

Zde chci popsat co to je distribuované zpracování dat. V další kapitole pak navážu, čím jaké nástroje v současnosti pro zpracování dat máme (Samsa, Storm)

## **3 Nástroje pro distribuované zpracování údajů**

### **3.1 Apache Samsa**

### **3.2 Apache Storm**

### **3.3 CAVE**

<http://dl.acm.org/citation.cfm?doid=2675743.2771834>





## **4 Vytvoření clusteru v rámci sítě**

### **4.1 Motivace**

### **4.2 Technologie**

Tato kapitola popisuje jednotlivé technologie, které jsou použity při implementaci algoritmu. Na konci každé subsekce bude popis toho jak konkrétně je technologie použita v mém řešení.

#### **4.2.1 Apache Maven**

#### **4.2.2 Apache Kafka**

#### **4.2.3 Apache ZooKeeper**

Popis technologie podle dokumentace

Konkrétní použití: Jak se sestavuje strom a k čemu slouží (je důležité zmínit, že jde pouze o virtuální stav a je nutné myslet na to, že aplikace běží na daném uzlu pouze jednou). Jaké má zookeeper tree výhody (dá se zjistit jestli a jaké má uzel potomky, dá se kterýkoli uzel modifikovat, což způsobí příslušnou změnu v aplikaci).

CuratorFramework

#### **4.2.4 Esper**

### **4.3 Události v systému**

Kapitola popisující hlavní myšlenku povahy dat, která bude algoritmus vyhodnocovat. Také zde budou ukázky používaných dat.

### 4.3.1 Hrubozrnné

### 4.3.2 Jemnozrnné

## 4.4 Konfigurace

## 4.5 Implementace

Pro implementaci byla zvolena Java (konkrétně ve verzi 1.8), protože všechny použité technologie mají dobrou API pro Javu a většina příkladů je právě v Javě. Dalším důvodem je také to, že Java se dobře hodí pro běh aplikací tohoto druhu, protože má dobrou práci s vlákny. Posledním, méně důležitým, důvodem je popularita Javy a snadná čitelnost kódu.

### 4.5.1 Architektura aplikace

Popis jednotlivých maven modulů. K čemu který slouží

### 4.5.2 Iniciální spuštění

### 4.5.3 Přejechy mezi stavy

Jedna z nejdůležitějších kapitol, která ukazuje co způsobí, že aplikace začne vyhodnocovat jemnozrnné události. Počínaje tím, že Esper vyhodnotí proud událostí a emituje ep-událost. Dále přes zpracování ep-události, nastavení příslušných dat jednotlivým zk-uzlům v zk-stromě, či modifikaci zk-stromu. Až po ukončení zpracovávání jemnozrnných událostí a návrat k iniciálnímu stavu.

Bude zde také zmíněno dynamické nasazování nových ep pravidel. To by se dalo shrnout pod nadpis "ovládání clusteru z venčí" - řešeno přes nastavování dat jednotlivým uzlům v zookeeperu.

### 4.5.4 Esper pravidla

Mini kapitola, kterou bych věnoval použitým esper pravidlům.

## 4.6 Známá omezení

Diskuse nedostatků nebo možných vylepšení výše navrženého řešení.

- Momentální nemožnost spustit více consumerů na jednom stroji.
- Velmi náročný monitoring a spouštění jednotlivých uzlů.
- Zatím nevím jak je to s dynamickým přidáváním nových uzlů do hierarchie.
- Nejsou vůbec otestovány výpadky některých uzlů. Zookeeper to zvládne, kafka také, ale co se stane s virtuálním zk-tree v aplikaci?



## 5 Závěr

Závěr bude v tomto případě obsahovat obšírnější zhodnocení toho jak se povedlo splnit zadání. Že výsledkem práce je navržené řešení za použití kafka, zk, esmeru, Javy.



## Bibliografie

- [1] LUCKHAM, David. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Pearson Education, Inc., 2002, ISBN 9780201727890