Spell: Streaming Parsing of System Event Logs

Min Du, Feifei Li

School of Computing,
University of Utah
Background

15/07/31 12:20:17 INFO SparkContext: Running Spark version 1.3.0
15/07/31 12:20:18 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using built-in-java classes where applicable
15/07/31 12:20:18 INFO SecurityManager: Changing view accts to: zhouliang
15/07/31 12:20:18 INFO SecurityManager: Changing modify accts to: zhouliang
15/07/31 12:20:18 INFO SecurityManager: SecurityManager: authentication disabled; ui accts disabled; users with view permssions: Set(zhouliang); users with modify permssions: Set(zhouliang)
15/07/31 12:20:18 INFO Sf4jLogger: Sf4jLogger started
15/07/31 12:20:18 INFO Remote: Starting remote
15/07/31 12:20:18 INFO Remote: Remote started; listening on addresses :[akka.tcp://sparkDriver@head:60626]
15/07/31 12:20:18 INFO Util: Successfully started service 'sparkDriver' on port 60626.
15/07/31 12:20:18 INFO SparkEnv: Registering MapOutputTracker
15/07/31 12:20:18 INFO SparkEnv: Registering BlockManagerMaster
15/07/31 12:20:18 INFO DiskBlockManager: Created local directory at /tmp/spark-3799bc3c-5275-499c-8b89-fe93ed0b313e/blockmgr-f7e603b7-c8c3-4fa5-be0c-2af14200c1e3
15/07/31 12:20:18 INFO MemoryStore: MemoryStore started with capacity 10.4 GB
15/07/31 12:20:19 INFO FileServer: HTTP file server directory is /tmp/spark-c01a992b-d9d3-4751-8f2e-05c2a644cb329/httpd-b9f5fc86-0f7c-43ac-aed4-20f27b9b3731
15/07/31 12:20:19 INFO FileServer: Starting HTTP Server
15/07/31 12:20:19 INFO Server: jetty-8.y.z-SNAPSHOT
15/07/31 12:20:19 INFO AbstractConnector: Started SocketConnector@0.0.0.0:43664
15/07/31 12:20:19 INFO Util: Successfully started service 'HTTP file server' on port 43664.
15/07/31 12:20:19 INFO SparkEnv: Registering OutputCommitCoordinator
15/07/31 12:20:19 INFO Server: jetty-8.y.z-SNAPSHOT
15/07/31 12:20:19 INFO AbstractConnector: Started SelectChannelConnector@0.0.0.0:4040
15/07/31 12:20:19 INFO Util: Successfully started service 'SparkUI' on port 4040.
15/07/31 12:20:19 INFO SparkContext: Added JAR file: /home/zhouliang/experiments/knn-join/./target/scala-2.18/knn-join_2.18-1.0.jar at http://192.168.1.2:43664/jars/knn-join_2.18-1.0.jar with timestamp 1438316419295
15/07/31 12:20:19 INFO AppClient$ClientActor: Connecting to master akka.tcp://sparkMaster@head:7077/user/Master...
15/07/31 12:20:19 INFO SparkDeploySchedulerBackend: Connected to Spark cluster with app ID
Background

System Event Log
Background

System Event Log

Exists practically on every computer system!
System Event Log

 Exists practically on every computer system!

Automatic Analysis?
Background

System Event Log

Started service A on port 80
Started service B on port 90
Started service C on port 100
Executor updated: app-1 is now LOADING
Executor updated: app-2 is now LOADING
TaskSetManager: Starting task 0 in stage 2
TaskSetManager: Starting task 1 in stage 5
......
Background

System Event Log

12:20:17 INFO SparkContext: Running Sp
12:20:18 WARN NativeCodeLoader: Unable ava classes where applicable
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Security
permissions: x=1023, u=1023, g=0, o=0; users wi
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Security
permissions: x=1023, u=1023, g=0, o=0; users wi
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Security
permissions: x=1023, u=1023, g=0, o=0; users wi
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Security
permissions: x=1023, u=1023, g=0, o=0; users wi
12:20:18 INFO SecurityManager: Changin

```
printf("Started service %s on port %d", x, y);
```

Started service A on port 80
Started service B on port 90
Started service C on port 100
Executor updated: app-1 is now LOADING
Executor updated: app-2 is now LOADING
TaskSetManager: Starting task 0 in stage 2
TaskSetManager: Starting task 1 in stage 5
……

Spell: Streaming Parsing of System Event Logs
Background

System Event Log

12:20:17 INFO SparkContext: Running Sp
12:20:18 WARN NativeCodeLoader: Unable to load classes where applicable
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Securit
permissions: set(zhoulang); users wi
12:20:18 INFO SecurityManager: Slf4jLogger
12:20:18 INFO SparkEnv: Starting remot
12:20:18 INFO Remoting: Remoting start
r@head:60626]
12:20:18 INFO Crypto: Successfully star
12:20:18 INFO SparkEnv: Registering Bl
12:20:18 INFO DlskBlockManager: Create
12:20:18 INFO MemoryStore: MemoryStore

Structured Data

Message/Event type
Log key
......
printf(“Started service
%s on port %d”, x, y);

Anomaly Detection

LOG ANALYSIS
Background

System Event Log

Structured Data
Message/Event type
Log key
......
printf("Started service \%s on port \%d", x, y);

Anomaly Detection

LOG ANALYSIS

- Message count vector:
  Xu’SOSP09, Lou’ATC10, Lin’ICSE16, etc.
Background

System Event Log

12:20:17 INFO SparkContext: Running Sp
12:20:18 WARN NativeCodeLoader: Unable to load classes where applicable
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SparkEnv: Registering Map
12:20:18 INFO SparkEnv: Registering Bl
12:20:18 INFO DconfBlockManager: Create
12:20:18 INFO MemoryStore: MemoryStore

12:20:18 INFO SparkContext: Running Sp
12:20:18 WARN NativeCodeLoader: Unable to load classes where applicable
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SparkEnv: Registering Map
12:20:18 INFO SparkEnv: Registering Bl
12:20:18 INFO DconfBlockManager: Create
12:20:18 INFO MemoryStore: MemoryStore

12:20:17 INFO SparkContext: Running Sp
12:20:18 WARN NativeCodeLoader: Unable to load classes where applicable
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SparkEnv: Registering Map
12:20:18 INFO SparkEnv: Registering Bl
12:20:18 INFO DconfBlockManager: Create
12:20:18 INFO MemoryStore: MemoryStore

12:20:17 INFO SparkContext: Running Sp
12:20:18 WARN NativeCodeLoader: Unable to load classes where applicable
12:20:18 INFO SecurityManager: Changin
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SecurityManager: Security permissions set(zhou-lang); users wi
12:20:18 INFO SparkEnv: Registering Map
12:20:18 INFO SparkEnv: Registering Bl
12:20:18 INFO DconfBlockManager: Create
12:20:18 INFO MemoryStore: MemoryStore

structured data

message/event type
log key

printf(“Started service %s on port %d”, x, y);

anomaly detection

message count vector:
Xu’SOSP09, Lou’ATC10, Lin’ICSE16, etc.

build workflow model:
Lou’KDD10, Beschastnikh’ICSE14,
Yu’ASPLOS16, etc.
Background

System Event Log

Structured Data

Message/Event type
Log key
......

printf("Started service \%s on port \%d", x, y);

LOG PARSING

Anomaly Detection

LOG PARSING
Background

System Event Log

Structured Data
Message/Event type
Log key
......
printf("Started service %s on port %d", x, y);

LOG PARSING

- Use source code as template to parse logs:
  Xu’SOSP09
Use source code as template to parse logs:
Xu’SOSP09
Problem: What if we don’t have source code?
Background

System Event Log

Structured Data
Message/Event type
Log key
......
printf("Started service %s on port %d", x, y);

LOG PARSING

- Use source code as template to parse logs:
  Xu’SOSP09
  Problem: What if we don’t have source code?

- Directly parse from raw system logs:
  Makanju’KDD09, Fu’ICDM09, Tang’ICDM10, Tang’CIKM11, etc.
Background

System Event Log

Structured Data
Message/Event type
Log key
......
printf("Started service %s on port %d", x, y);

LOG PARSING

- Use source code as template to parse logs:
  Xu’SOSP09
  Problem: What if we don’t have source code?

- Directly parse from raw system logs:
  Makanju’KDD09, Fu’ICDM09, Tang’ICDM10, Tang’CIKM11, etc.
  Problem: Offline batched processing, some very slow.
Our approach

*Spell*, a structured Streaming Parser for Event Logs using an LCS (longest common subsequence) based approach.
Our approach

*Spell*, a structured *Streaming Parser for Event Logs* using an LCS (longest common subsequence) based approach.

Example:

Two log entries:
- Temperature (41C) exceeds warning threshold
- Temperature (42C, 43C) exceeds warning threshold
Our approach

**Spell**, a structured **Streaming Parser** for **Event Logs** using an **LCS** (longest common subsequence) based approach.

**Example:**

Two log entries:

- Temperature (41C) exceeds warning threshold
- Temperature (42C, 43C) exceeds warning threshold

**LCS:**

- Temperature * exceeds warning threshold
Our approach

**Spell**, a structured *Streaming Parser* for *Event Logs* using an *LCS* (longest common subsequence) based approach.

**Example:**

Two log entries:

- Temperature (41C) exceeds warning threshold
- Temperature (42C, 43C) exceeds warning threshold

**LCS:**

- Temperature * exceeds warning threshold

Naturally a message type!

```c
printf("Temperature %s exceeds warning threshold")
```
SPELL – Basic workflow

Add new log entry into LCSMap in a streaming fashion, update existing message type if
\[ \text{length}(\text{LCS}) > 0.5 \times \text{length}(\text{new log entry}) \]
new log entry: Temperature (41°C) exceeds warning threshold
new log entry:

```plaintext
LCSObject

LCSseq: Temperature (41C) exceeds warning threshold
linelds: [0]
paramPos: [empty]
```

LCSMap
SPELL – Basic workflow

new log entry: Temperature (43C) exceeds warning threshold

LCSObject

LCSseq: Temperature (41C) exceeds warning threshold
linelds: {0}
paramPos: {empty}
SPELL – Basic workflow

new log entry:

LCSObject: LCSseq: *Temperature* exceeds warning threshold
lineIds: {0, 1}
paramPos: {1}
**SPELL – Basic workflow**

**new log entry:** *Command has completed successfully*

```
LCSObject

LCSseq: Temperature * exceeds warning threshold
linelIds: {0, 1}
paramPos: {1}
```

```
LCSMap
```
new log entry:

\[
\begin{align*}
\text{LCSObject} & \quad \text{LCSseq: Temperature } \ast \text{ exceeds warning threshold} \\
& \quad \text{linels: } \{0, 1\} \quad \text{paramPos: } \{1\} \\
\text{LCSObject} & \quad \text{LCSseq: Command has completed successfully} \\
& \quad \text{linels: } \{2\} \quad \text{paramPos: } \{\text{empty}\}
\end{align*}
\]
SPELL – Basic workflow

new log entry: …...

LCSObject

\[
\begin{align*}
\text{LCSseq: Temperature } & \ast \text{ exceeds warning threshold} \\
\text{linelds: } & \{0, 1\} \\
\text{paramPos: } & \{1\}
\end{align*}
\]

LCSObject

\[
\begin{align*}
\text{LCSseq: Command has completed successfully} \\
\text{linelds: } & \{2\} \\
\text{paramPos: } & \{\text{empty}\}
\end{align*}
\]

…….

_**LCSMap**_
To compute LCS of two log entries, each one has $O(n)$ length:
To compute LCS of two log entries, each one has $O(n)$ length:

Naïve way: Dynamic Programing
To compute LCS of two log entries, each one has $O(n)$ length:

**Naïve way:** Dynamic Programming

**Time complexity:**
- To compare a log entry with an existing message type: $O(n^2)$
- To compare a new log entry with $O(m)$ existing message types: $O(mn^2)$
SPELL – Improvement on efficiency

To compute LCS of two log entries, each one has $O(n)$ length:

**Naïve way:** Dynamic Programming

**Time complexity:**
- To compare a log entry with an existing message type: $O(n^2)$
- To compare a new log entry with $O(m)$ existing message types: $O(mn^2)$

*Can we do better?*
SPELL – Improvement on efficiency

Observation.

For a complex system,
number of log entries: millions
number of message types: hundreds
SPELL – Improvement on efficiency

Observation.

For a complex system,
number of log entries: millions
number of message types: hundreds

For example:

Blue Gene/L log:
4,457,719 log entries, 394 message types

Hadoop log used in Xu’SOSP09:
11,197,705 log entries, only 29 message types
Observation.

For a complex system,
number of log entries: millions
number of message types: hundreds

For example:

Blue Gene/L log:
4,457,719 log entries, 394 message types

Hadoop log used in Xu’SOSP09:
11,197,705 log entries, only 29 message types

For a majority of new log entries, their message types already exist in LCSMap!
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

Existing message types:

A B C
A C D
A D
E F
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

Existing message types:
A B C
A C D
A D
E F

Diagram:
- ROOT
  - A
    - B
    - C
  - E
    - D
    - F
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

New log entry: $A \ B \ P \ C$

Diagram:
- ROOT
  - $A$
    - $B$
    - $C$
  - $E$
    - $D$
    - $F$
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

New log entry: A B P C

![Prefix Tree Diagram]

Spell: Streaming Parsing of System Event Logs
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

New log entry: A B P C

Spell: Streaming Parsing of System Event Logs
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

New log entry: A B P C

Parameter: B C D

ROOT

A

B

C

D

E

F

Spell: Streaming Parsing of System Event Logs
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

New log entry: A B P C

Parameter:
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

Time Complexity: $O(n)$ for each log entry
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

Problem:
New log entry: $D A P B C$
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

Problem:
New log entry: D A P B C
Matches D A
SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

Problem:
New log entry: D A P B C
Matches D A
Should be: A B C
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

- [A B C]
- [A E F]
- [D A]

New log entry:

- [D A P B C]
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

\[ \text{Pointer } P_m \]

Message types:

\[ [A \ B \ C] \]

\[ [A \ E \ F] \]

\[ [D \ A] \]

New log entry:

\[ [D \ A \ P \ B \ C] \]

\[ \text{Pointer } P_l \]
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

\[
\begin{array}{ccc}
A & B & C \\
A & E & F \\
D & A \\
\end{array}
\]

New log entry:

\[
\begin{array}{ccccc}
D & A & P & B & C \\
\end{array}
\]
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

[ A  B  C ]

[ A  E  F ]

[ D  A ]

New log entry:

[ D  A  P  B  C ]
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

\[
\begin{array}{ccc}
A & B & C \\
A & E & F \\
D & A \\
\end{array}
\]

New log entry:

\[
\begin{array}{cccc}
D & A & P & B & C \\
\end{array}
\]

\[\text{Pointer } P_m \]

\[\text{Pointer } P_l\]
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

[ A  B  C ]

[ A  E  F ]

[ D  A ]

New log entry:

[ D  A  P  B  C ]

\( \text{Pointer } P_m \)

\( \text{Pointer } P_l \)
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

\[
\begin{bmatrix}
A & B & C \\
A & E & F \\
D & A \\
\end{bmatrix}
\]

New log entry:

\[
\begin{bmatrix}
D & A & P & B & C \\
\end{bmatrix}
\]
SPPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

\[
\begin{aligned}
\text{[ A  B  C ]} \\
\text{[ A  E  F ]} \\
\text{[ D  A ]}
\end{aligned}
\]

New log entry:

\[
\begin{aligned}
\text{[ D  A  P  B  C ]}
\end{aligned}
\]
<table>
<thead>
<tr>
<th>Message types</th>
<th>Matched length</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A B C]</td>
<td>3</td>
</tr>
<tr>
<td>[A E F]</td>
<td>N/A</td>
</tr>
<tr>
<td>[D A ]</td>
<td>2</td>
</tr>
<tr>
<td>New log entry:</td>
<td>[D A P B C]</td>
</tr>
</tbody>
</table>
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

```
[ A  B  C ]
[ A  E  F ]
[ D  A ]
```

New log entry:

```
[ D  A  P  B  C ]
```

Return as a match!
Spell: Streaming Parsing of System Event Logs

**SPPELL – Improvement on efficiency**

**Improvement 2: Simple Loop**

Compare each message type with new log entry

**Message types:**

- [A B C]
- [A E F]
- [D A]

**New log entry:**

- [D A P B C]

**Time complexity**

$O(mn)$

- Number of message types
- Log entry length

56

Spell: Streaming Parsing of System Event Logs
SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:

\[
\begin{bmatrix}
A & B & C \\
A & E & F \\
D & A \\
\end{bmatrix}
\]

New log entry:

\[
\begin{bmatrix}
D & A & P & B & C \\
\end{bmatrix}
\]

Time complexity

\(O(mn)\)

For remaining log entries, compare it with each message type using simple DP.
Evaluation

Methods to compare:

- IPLoM (Makanju’KDD09): Partition log file using 3-step heuristics (log entry length, etc.)
- CLP (Fu’ICDM09): Cluster similar logs together based on weighted edit distance

Log dataset:

<table>
<thead>
<tr>
<th>Log type</th>
<th>Count</th>
<th>Message type ground truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Los Alamos HPC log</td>
<td>433,490</td>
<td>Available online</td>
</tr>
<tr>
<td>BlueGene/L log</td>
<td>4,747,963</td>
<td>Available online</td>
</tr>
</tbody>
</table>
Evaluation - Efficiency

Spell: Streaming Parsing of System Event Logs

![Graph showing runtime vs. log size for different methods: Spell (naive LCS), CLP (fixed threshold), IPLoM, Spell, CLP (auto threshold). The x-axis represents log size in \(\times 10^5\) for Los Alamos and Blue Gene, while the y-axis represents runtime in seconds.]
Evaluation - Effectiveness

Spell: Streaming Parsing of System Event Logs
Conclusion

Spell:

- A streaming system event log parser
- Using LCS
- Prefix tree and simple loop to improve efficiency
- Outperform offline methods on large system log dataset

Thank you

mind@cs.utah.edu
Evaluation - Efficiency

**Number (Percentage) of Log Entries Returned by Each Step**

<table>
<thead>
<tr>
<th>Step</th>
<th>Los Alamos HPC log</th>
<th>BlueGene/L log</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix tree</td>
<td>397,412 (91.68%)</td>
<td>4,457,719 (93.89%)</td>
</tr>
<tr>
<td>simple loop</td>
<td>35,691 (8.23%)</td>
<td>288,254 (6.07%)</td>
</tr>
<tr>
<td>naive LCS</td>
<td>387 (0.09%)</td>
<td>1,990 (0.042%)</td>
</tr>
</tbody>
</table>

**Amortized Cost of Each Message Type Lookup Step in Spell**

<table>
<thead>
<tr>
<th>Step</th>
<th>Los Alamos HPC log</th>
<th>BlueGene/L log</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix tree (ms)</td>
<td>0.006</td>
<td>0.011</td>
</tr>
<tr>
<td>simple loop (ms)</td>
<td>0.020</td>
<td>0.087</td>
</tr>
<tr>
<td>naive LCS (ms)</td>
<td>0.175</td>
<td>0.580</td>
</tr>
</tbody>
</table>
## Evaluation - Effectiveness

### Comparison of Spell with and without Pre-filter

<table>
<thead>
<tr>
<th>Spell</th>
<th>Los Alamos Type Found</th>
<th>HPC Log Accuracy</th>
<th>BlueGene/L Log Type Found</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>With pre-filtering</td>
<td>True message types found</td>
<td>Accuracy</td>
<td>True message types found</td>
<td>Accuracy</td>
</tr>
<tr>
<td>False</td>
<td>55</td>
<td>0.822786</td>
<td>165</td>
<td>0.811798</td>
</tr>
<tr>
<td>True</td>
<td>55</td>
<td>0.822786</td>
<td>164</td>
<td>0.811791</td>
</tr>
</tbody>
</table>