Spell: Streaming Parsing of System Event Logs

Min Du, Feifei Li

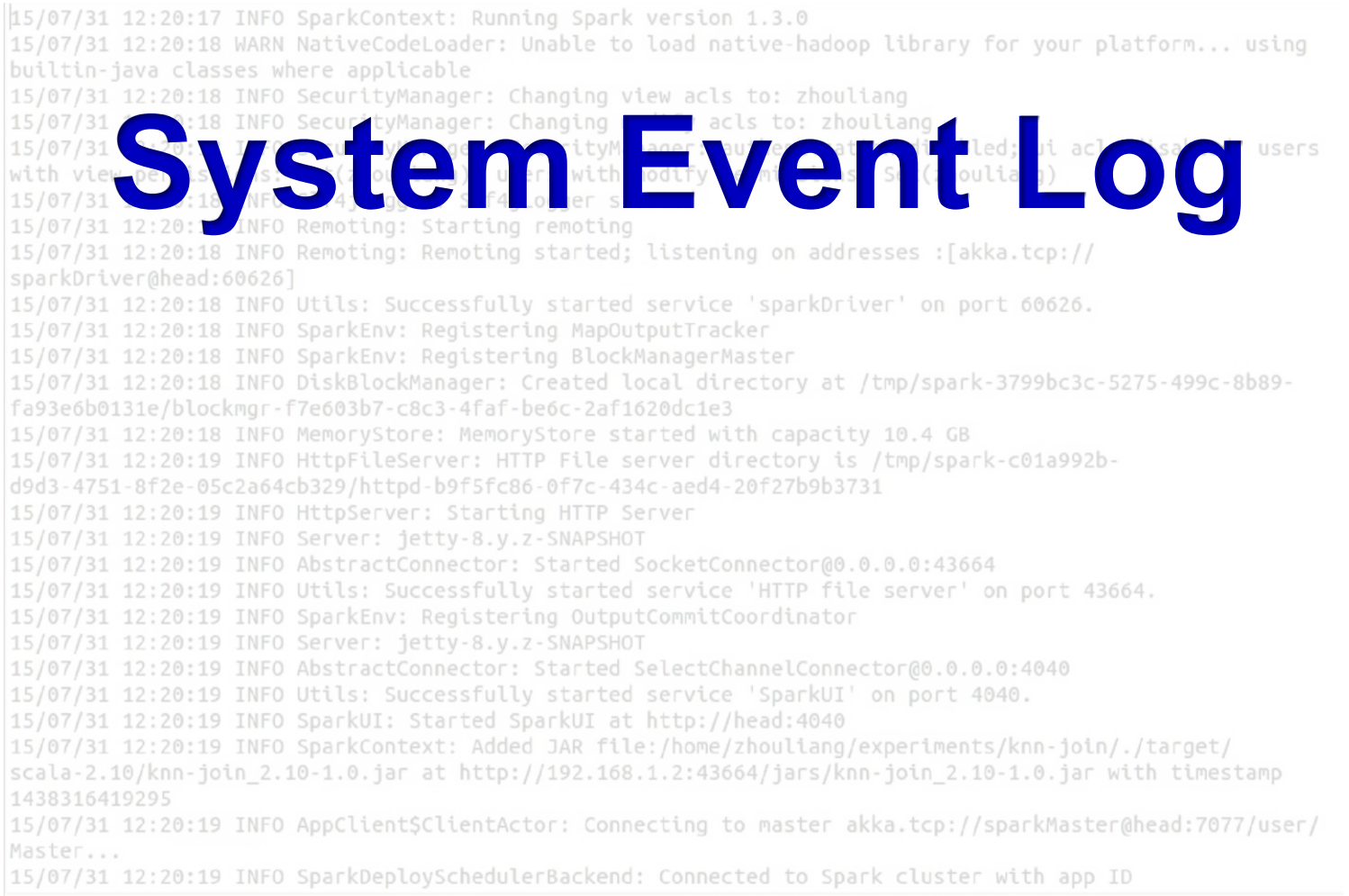
School of Computing,

University of Utah

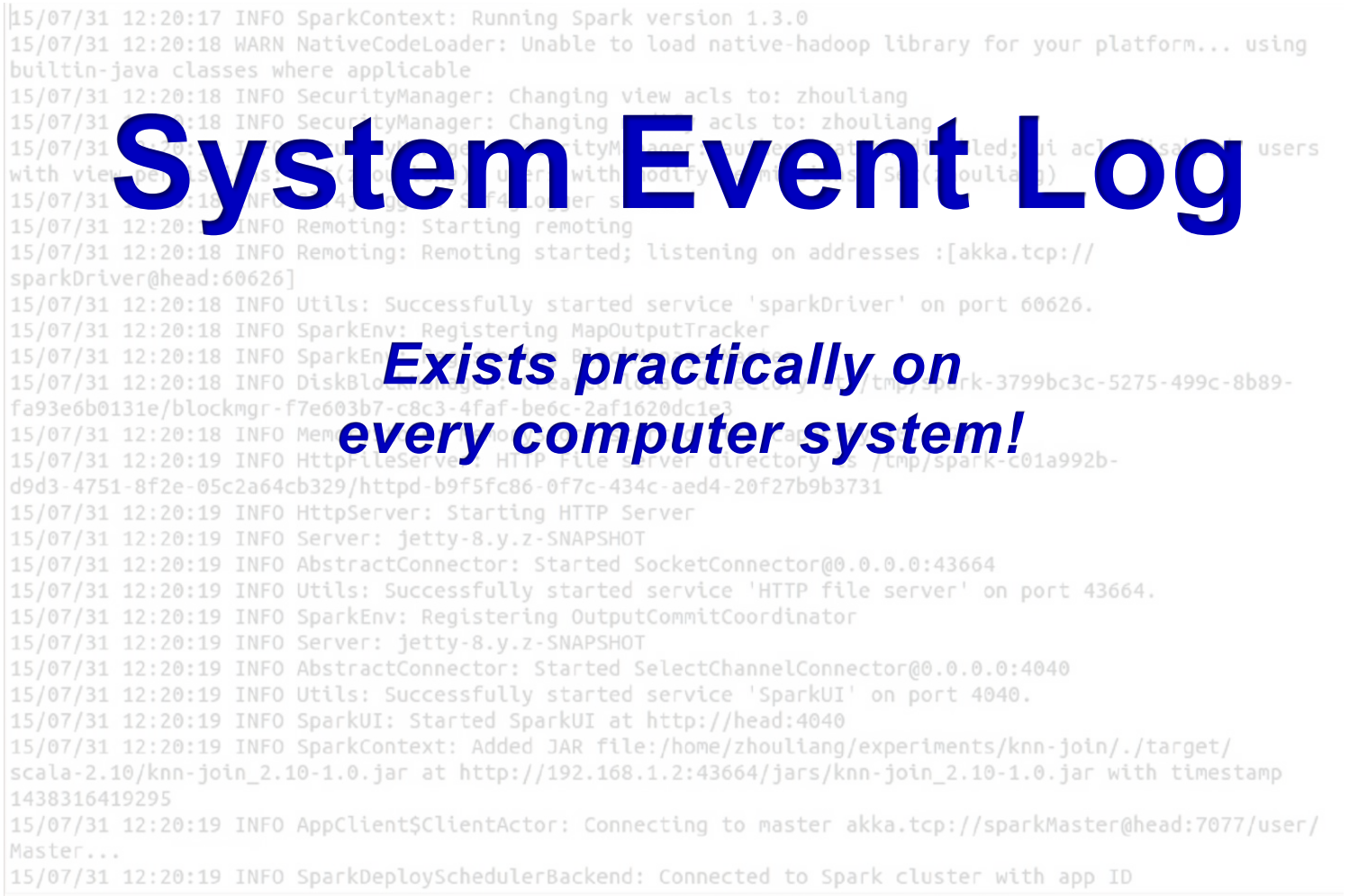
Background

```
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
builtin-java classes where applicable
15/07/31 12:20:18 INFO SecurityManager: Changing view acls to: zhouliang
15/07/31 12:20:18 INFO SecurityManager: Changing modify acls to: zhouliang
15/07/31 12:20:18 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users
with view permissions: Set(zhouliang); users with modify permissions: Set(zhouliang)
15/07/31 12:20:18 INFO Slf4jLogger: Slf4jLogger started
15/07/31 12:20:18 INFO Remoting: Starting remoting
15/07/31 12:20:18 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://
sparkDriver@head:60626]
15/07/31 12:20:18 INFO Utils: 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-
fa93e6b0131e/blockmgr-f7e603b7-c8c3-4faf-be6c-2af1620dc1e3
15/07/31 12:20:18 INFO MemoryStore: MemoryStore started with capacity 10.4 GB
15/07/31 12:20:19 INFO HttpFileServer: HTTP File server directory is /tmp/spark-c01a992b-
d9d3-4751-8f2e-05c2a64cb329/httpd-b9f5fc86-0f7c-434c-aed4-20f27b9b3731
15/07/31 12:20:19 INFO HttpServer: 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 Utils: 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 Utils: Successfully started service 'SparkUI' on port 4040.
15/07/31 12:20:19 INFO SparkUI: Started SparkUI at http://head:4040
15/07/31 12:20:19 INFO SparkContext: Added JAR file:/home/zhouliang/experiments/knn-join/./target/
scala-2.10/knn-join_2.10-1.0.jar at http://192.168.1.2:43664/jars/knn-join_2.10-1.0.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
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 builtin-java classes where applicable
15/07/31 12:20:18 INFO SecurityManager: Changing view acls to: zhouliang
15/07/31 12:20:18 INFO SecurityManager: Changing acls to: zhouliang
15/07/31 12:20:18 INFO Remoting: Starting remoting
15/07/31 12:20:18 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://sparkDriver@head:60626]
15/07/31 12:20:18 INFO Utils: 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-fa93e6b0131e/blockmgr-f7e603b7-c8c3-4faf-be6c-2af1620dc1e3
15/07/31 12:20:18 INFO MemoryStore: MemoryStore started with capacity 10.4 GB
15/07/31 12:20:19 INFO HttpFileServer: HTTP File server directory is /tmp/spark-c01a992b-d9d3-4751-8f2e-05c2a64cb329/httpd-b9f5fc86-0f7c-434c-aed4-20f27b9b3731
15/07/31 12:20:19 INFO HttpServer: 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 Utils: 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 Utils: Successfully started service 'SparkUI' on port 4040.
15/07/31 12:20:19 INFO SparkUI: Started SparkUI at http://head:4040
15/07/31 12:20:19 INFO SparkContext: Added JAR file:/home/zhouliang/experiments/knn-join/./target/scala-2.10/knn-join_2.10-1.0.jar at http://192.168.1.2:43664/jars/knn-join_2.10-1.0.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



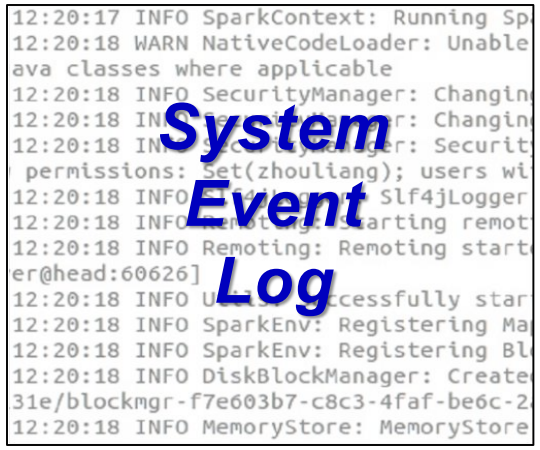
System Event Log

Exists practically on every computer system!

Background


System Event Log
Exists practically on
every computer system!
Automatic Analysis?

Background



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

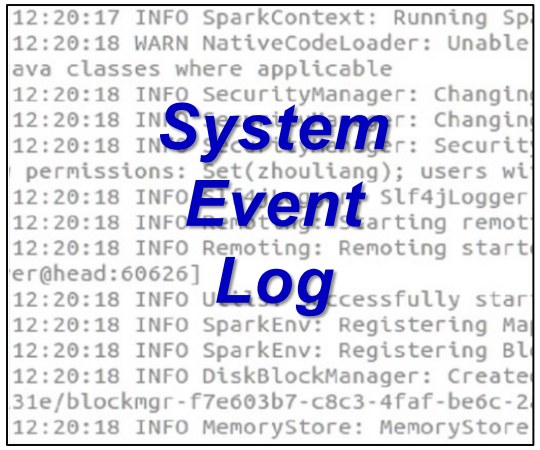
Executor updated: app-2 is now LOADING

TaskSetManager: Starting task 0 in stage 2

TaskSetManager: Starting task 1 in stage 5

……

Background



System Event Log

Structured Data

Message/Event type
Log key
……

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

……

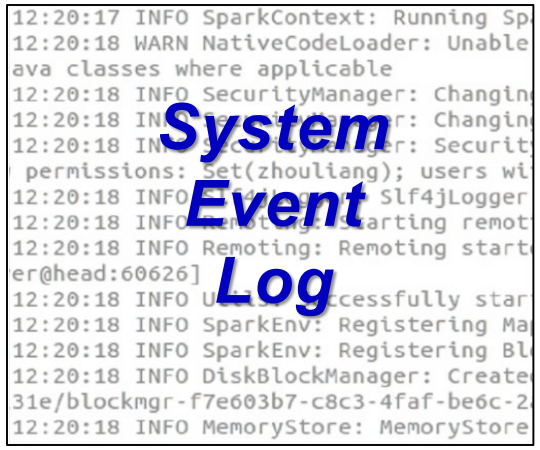
Started service * on port *

Executor updated: * is now LOADING

TaskSetManager: Starting task * in stage *

……

Background



System Event Log

Structured Data

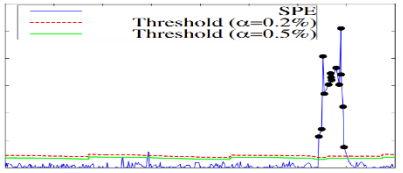
Message/Event type

Log key

……

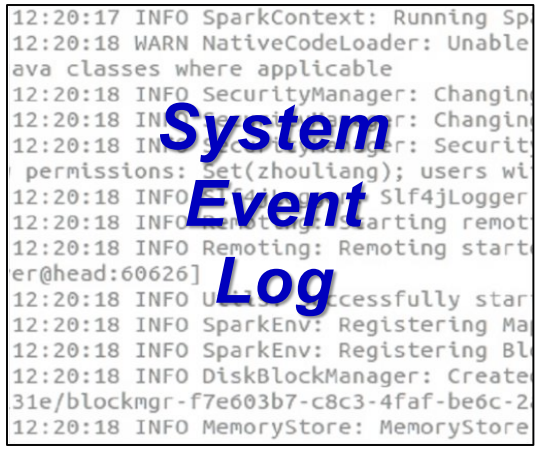
printf("***Started service*** %s ***on port*** %d", x, y);

Anomaly Detection



L O G A N A L Y S I S

Background



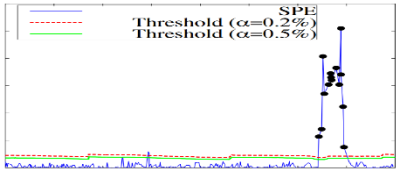
System Event Log

Structured Data

Message/Event type
Log key
……

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

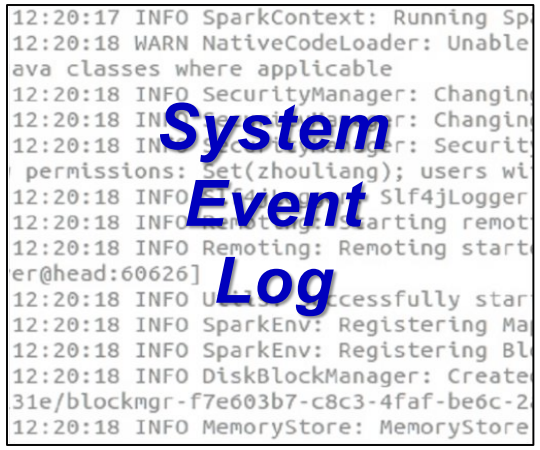
Anomaly Detection



L O G A N A L Y S I S

- **Message count vector:**
  Xu'SOSP09, Lou'ATC10, Lin'ICSE16, etc.

Background



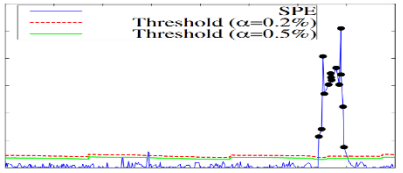
System Event Log

Structured Data

Message/Event type
Log key
……

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

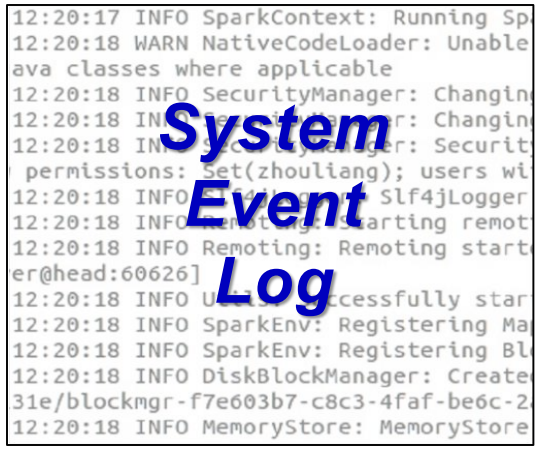
Anomaly Detection



L O G A N A L Y S I S

- **Message count vector:**
  Xu'SOSP09, Lou'ATC10, Lin'ICSE16, etc.
- **Build workflow model:**
  Lou'KDD10, Beschastnikh'ICSE14, Yu'ASPLOS16, etc.

Background



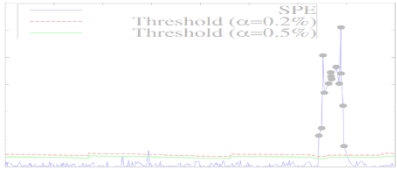
System Event Log

Structured Data

Message/Event type
Log key
……

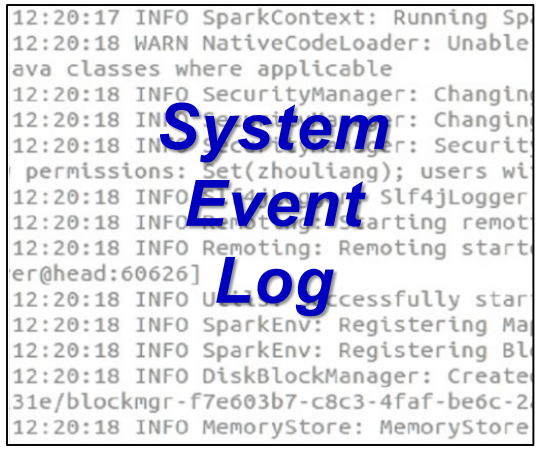
printf("***Started service*** %s ***on port*** %d", x, y);

Anomaly Detection



L O G P A R S I N G

Background



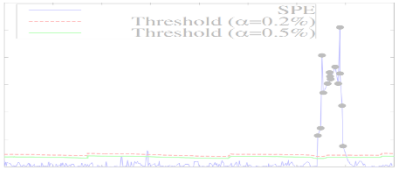
System Event Log

Structured Data

Message/Event type
Log key
……

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

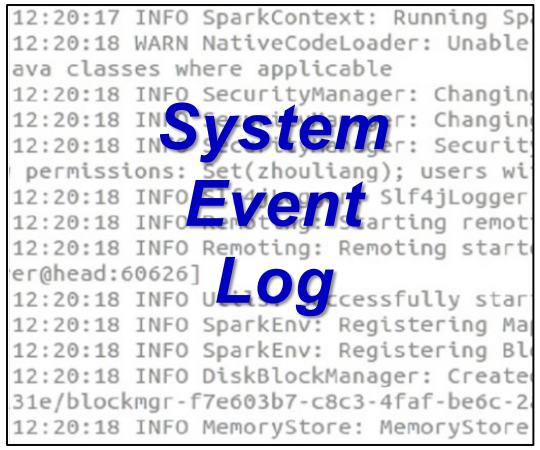
Anomaly Detection



L O G P A R S I N G

- **Use source code as template to parse logs:**
  Xu'SOSP09

Background



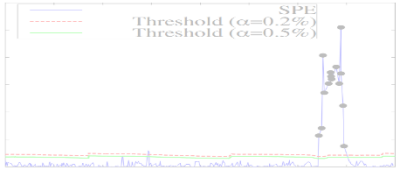
System Event Log

Structured Data

Message/Event type
Log key
……

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

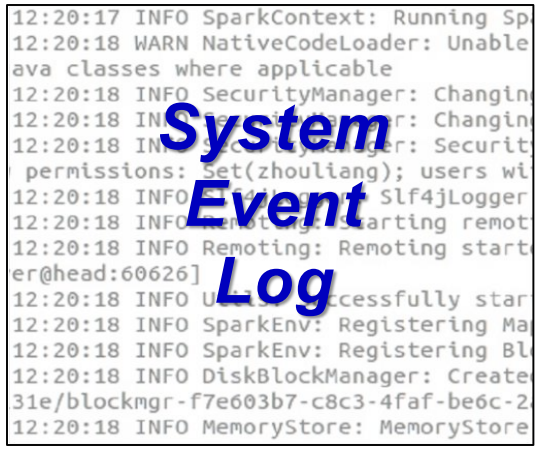
Anomaly Detection



L O G P A R S I N G

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

Background



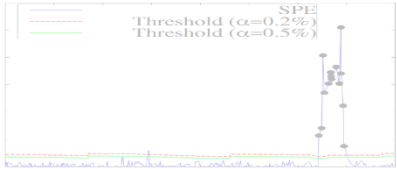
System Event Log

Structured Data

Message/Event type
Log key
……

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

Anomaly Detection

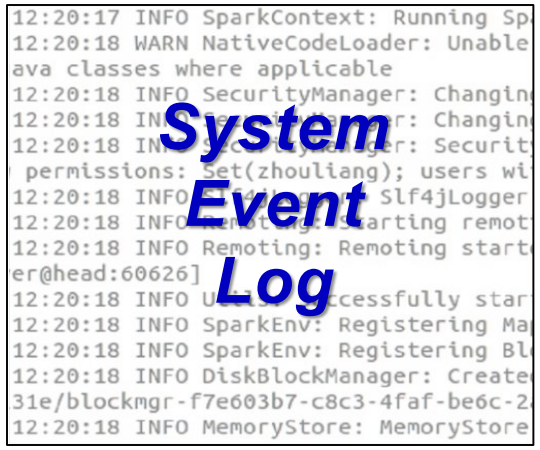


L O G P A R S I N G

- **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



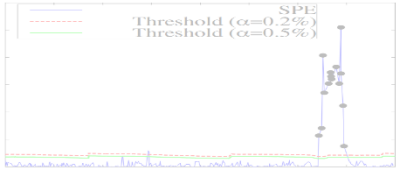
System Event Log

Structured Data

Message/Event type
Log key
……

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

Anomaly Detection



L O G P A R S I N G

- **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 *S*treaming *P*arser for *E*vent *L*ogs using an *L*CS (longest common subsequence) based approach.**

Our approach

***Spell*, a structured *S*treaming *P*arser for *E*vent *L*ogs using an *L*CS (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 *S*treaming *P*arser for *E*vent *L*ogs using an *L*CS (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 *S*treaming *P*arser for *E*vent *L*ogs using an *L*CS (longest common subsequence) based approach.**

Example:

Two log entries:

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

LCS:

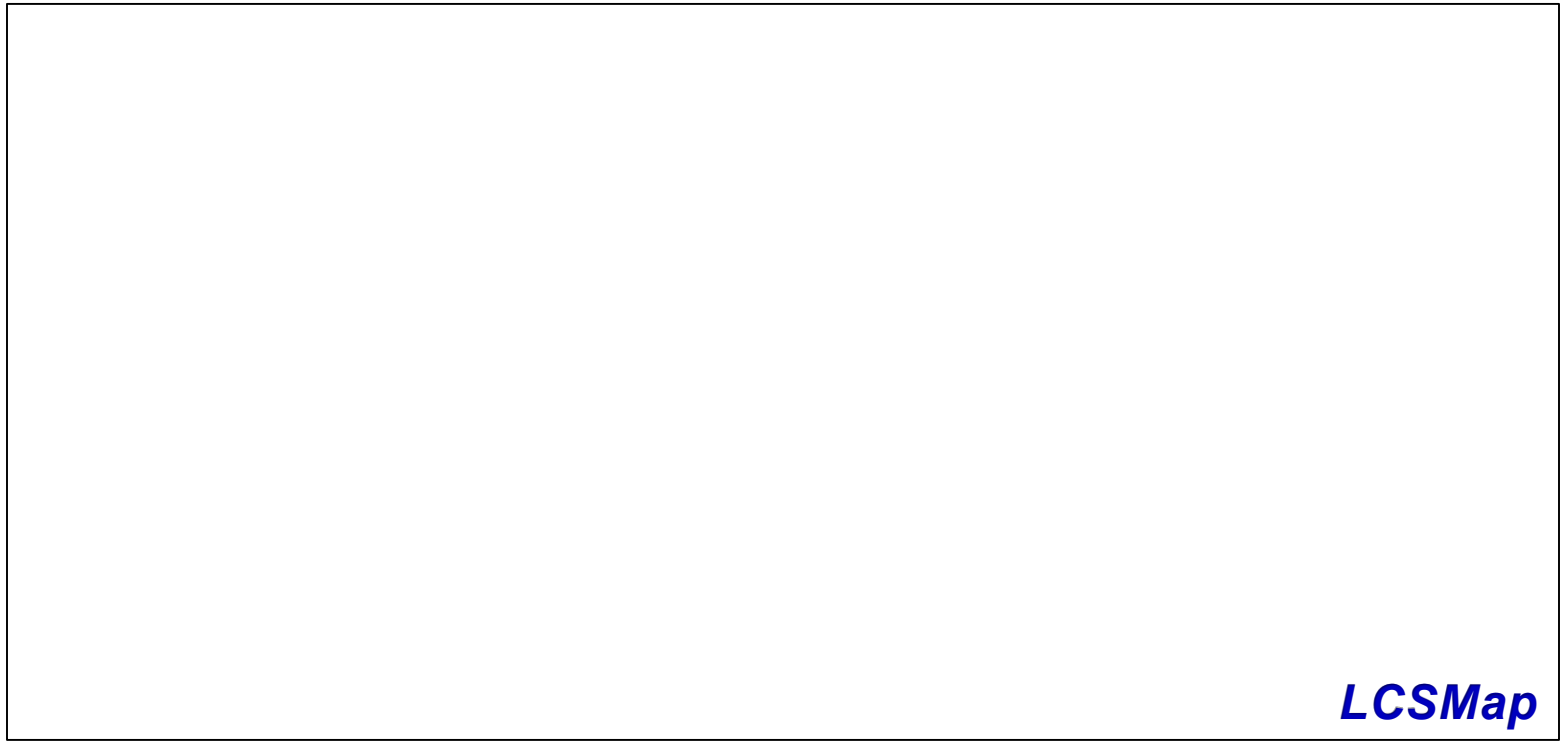
*Temperature * exceeds warning threshold*

Naturally a message type!

printf("Temperature %s exceeds warning threshold")

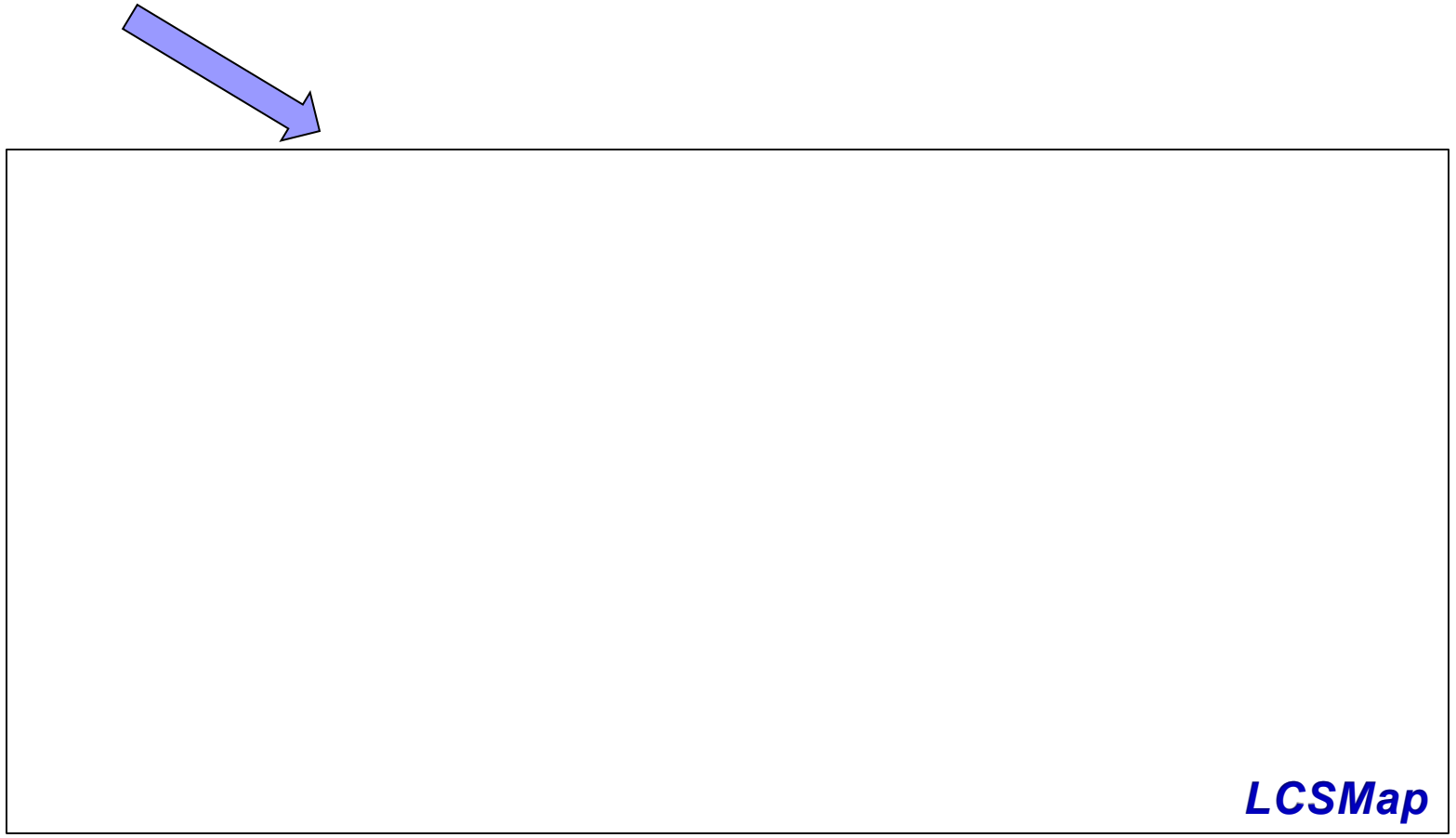
SPELL – Basic workflow

Add new log entry into LCSMap in a streaming fashion, update existing message type if *length(LCS) > 0.5 * length(new log entry)*



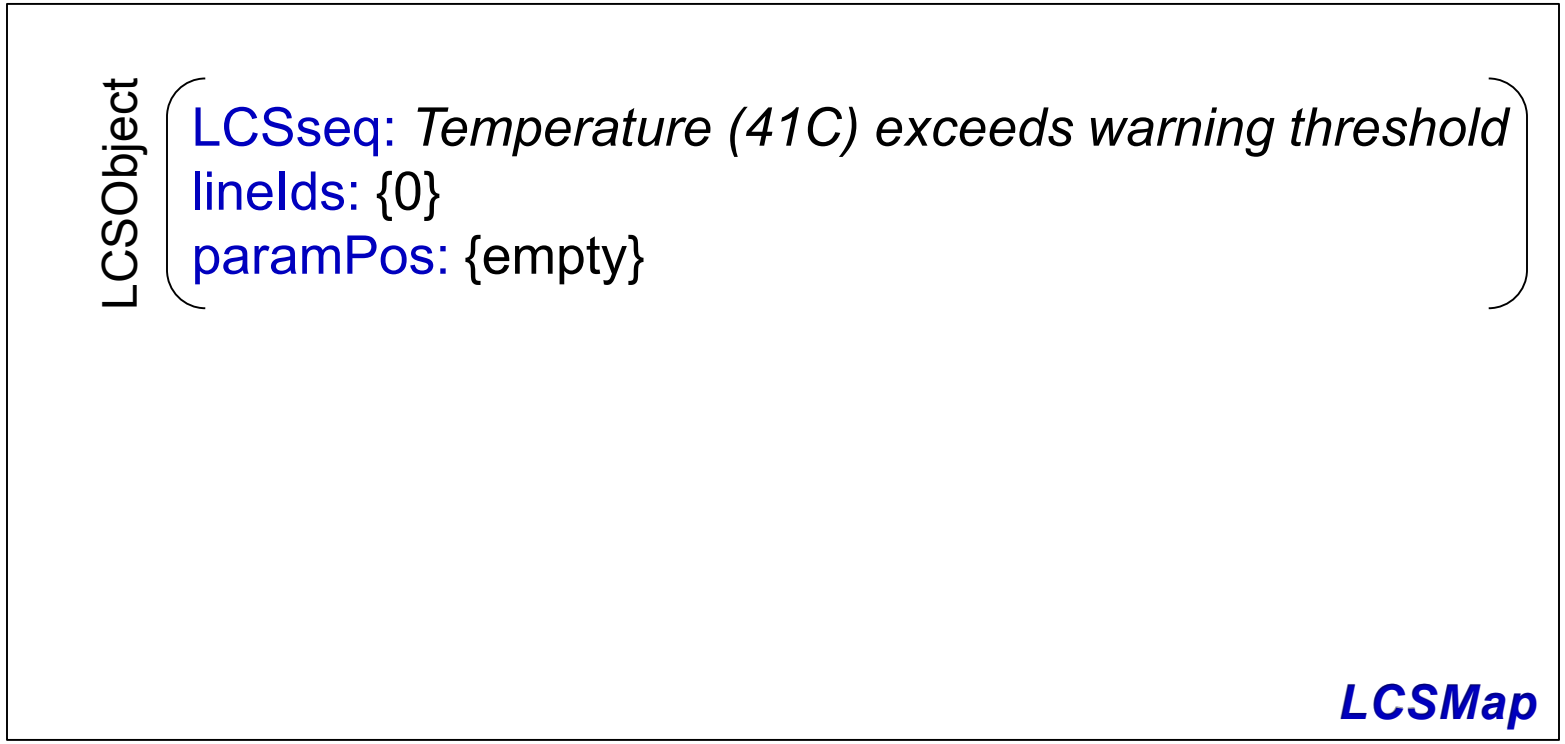
SPELL – Basic workflow

new log entry: *Temperature (41C) exceeds warning threshold*



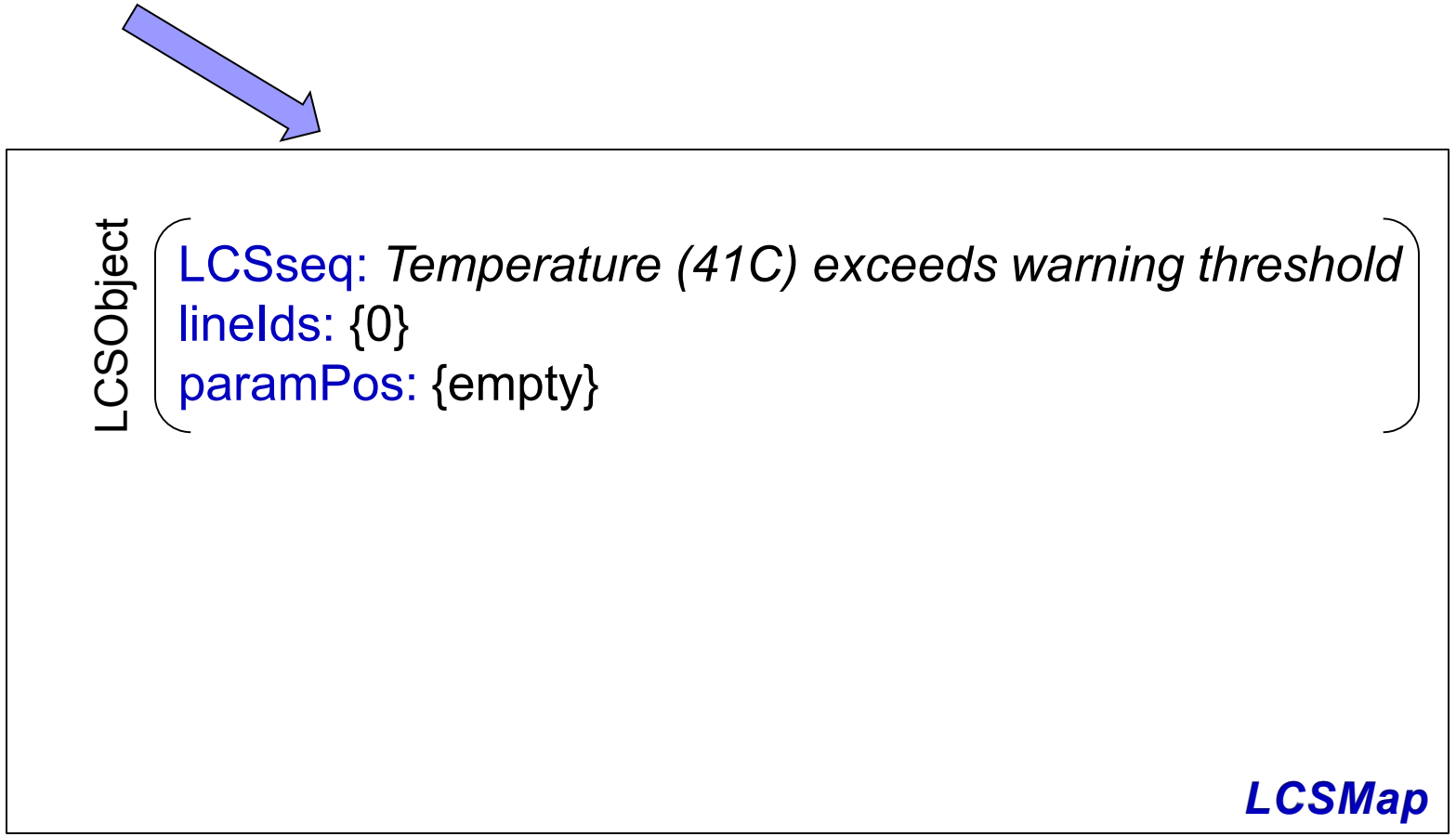
SPELL – Basic workflow

new log entry:



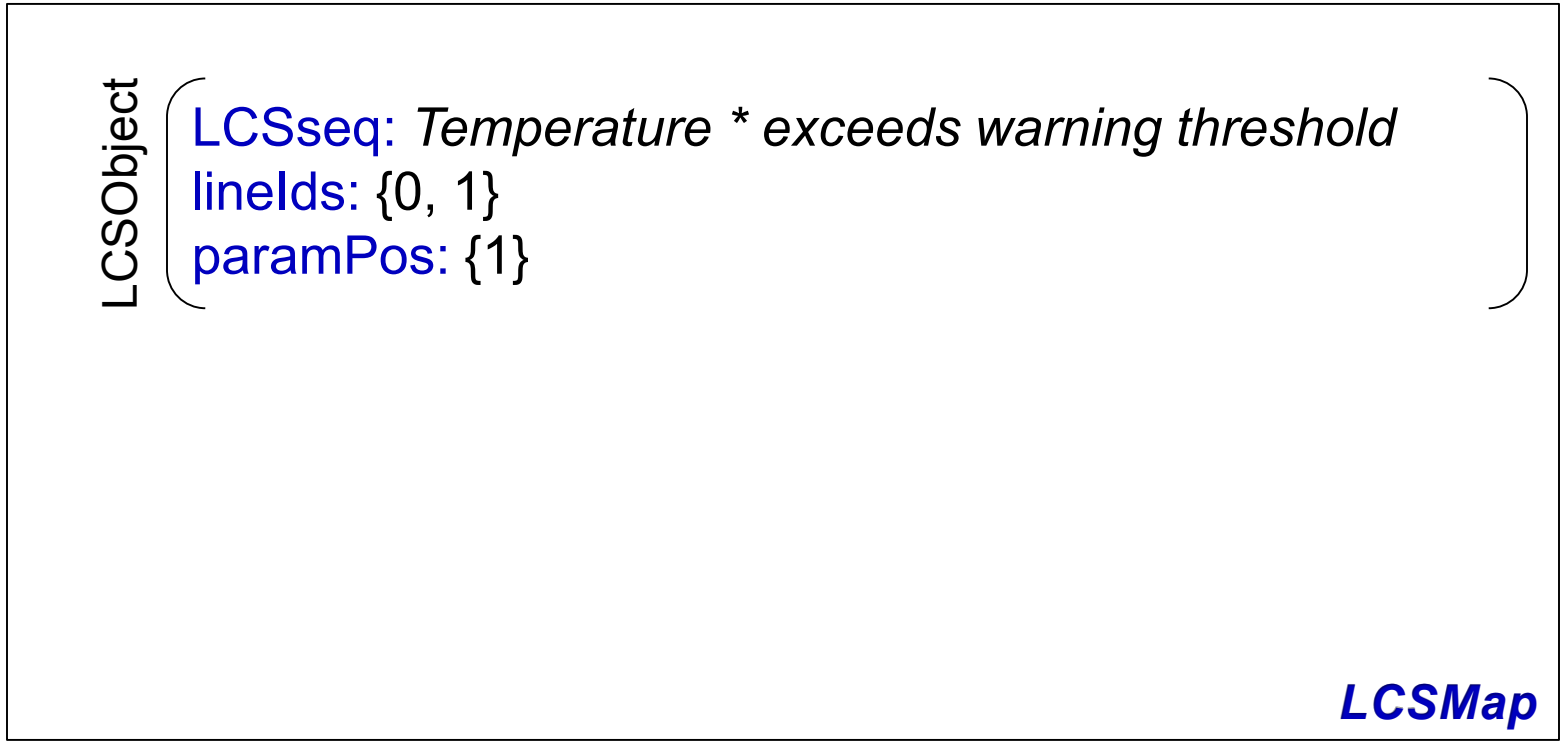
SPELL – Basic workflow

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



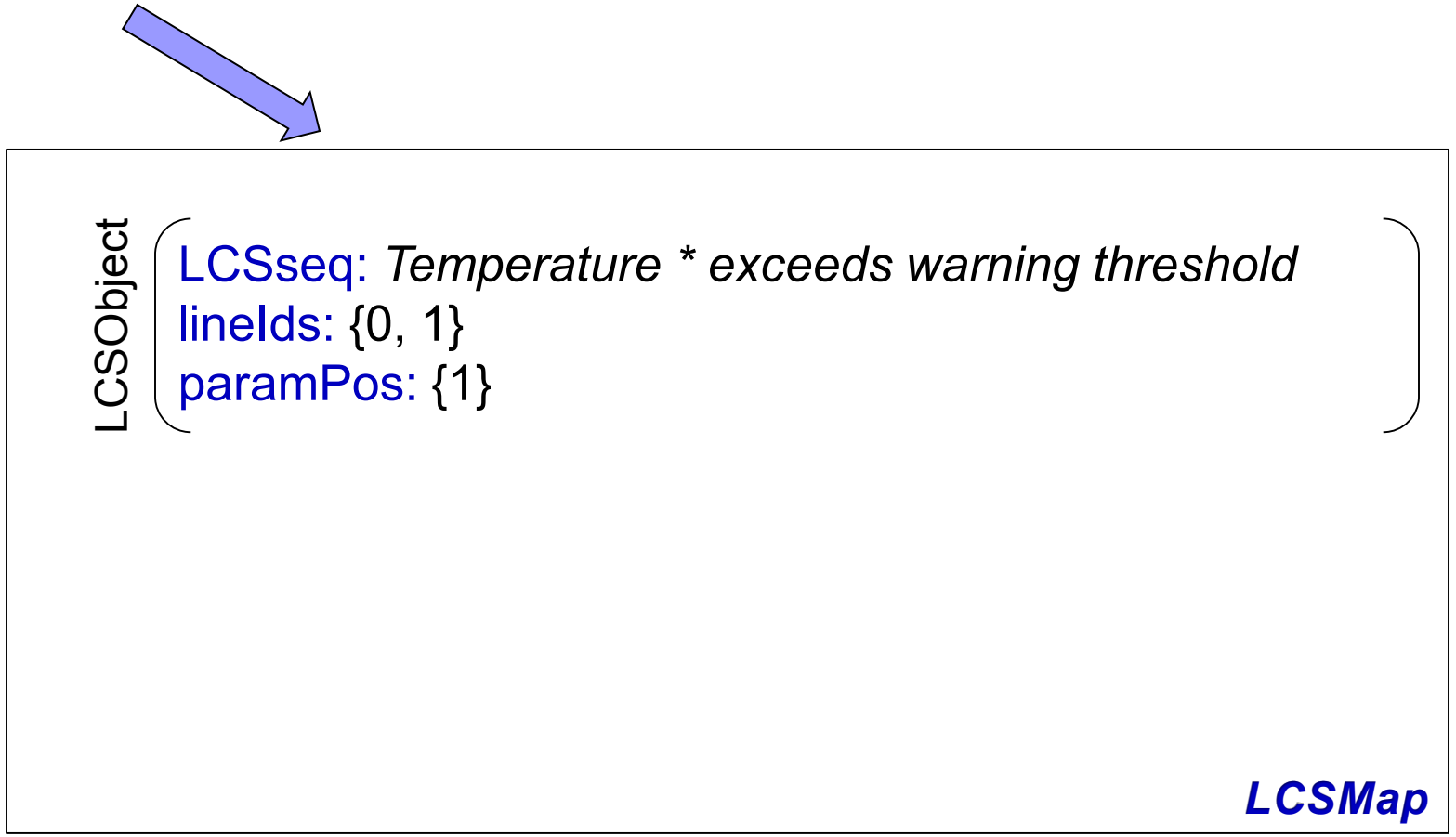
SPELL – Basic workflow

new log entry:



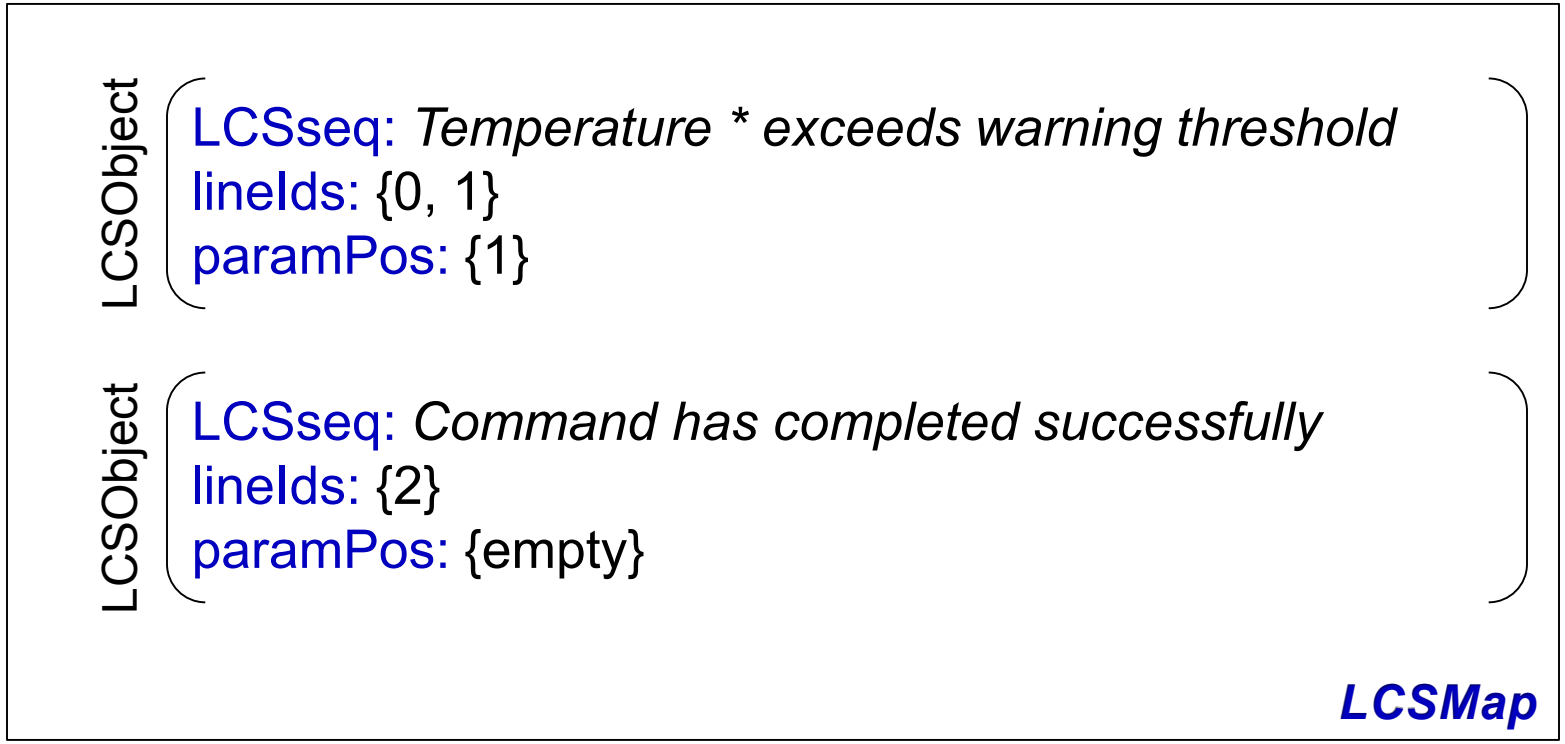
SPELL – Basic workflow

new log entry: *Command has completed successfully*



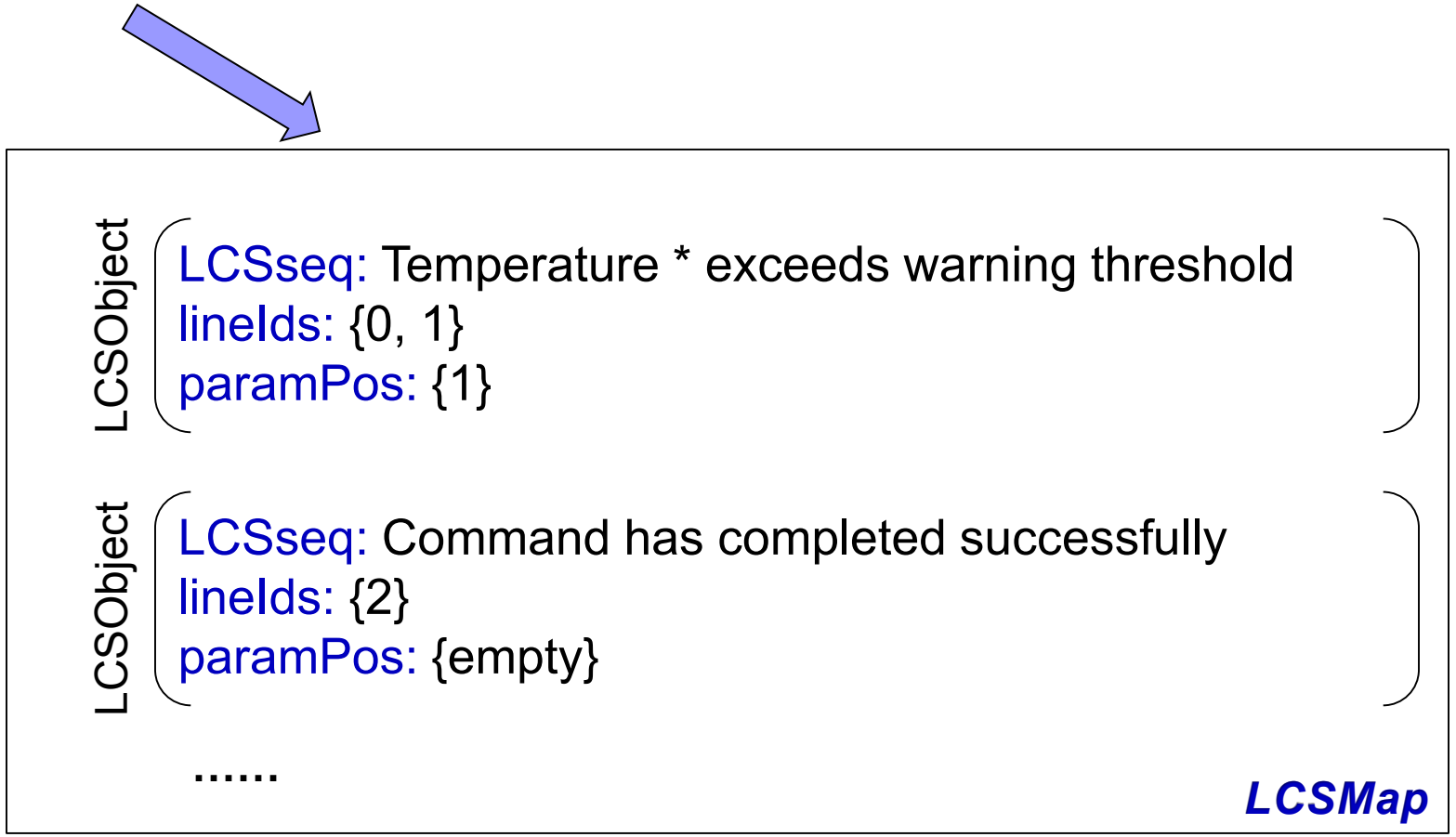
SPELL – Basic workflow

new log entry:



SPELL – Basic workflow

new log entry: ……



SPELL – Improvement on efficiency

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

SPELL – Improvement on efficiency

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

Naïve way: Dynamic Programing

SPELL – Improvement on efficiency

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

Naïve way: Dynamic Programing

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 Programing

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

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

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:

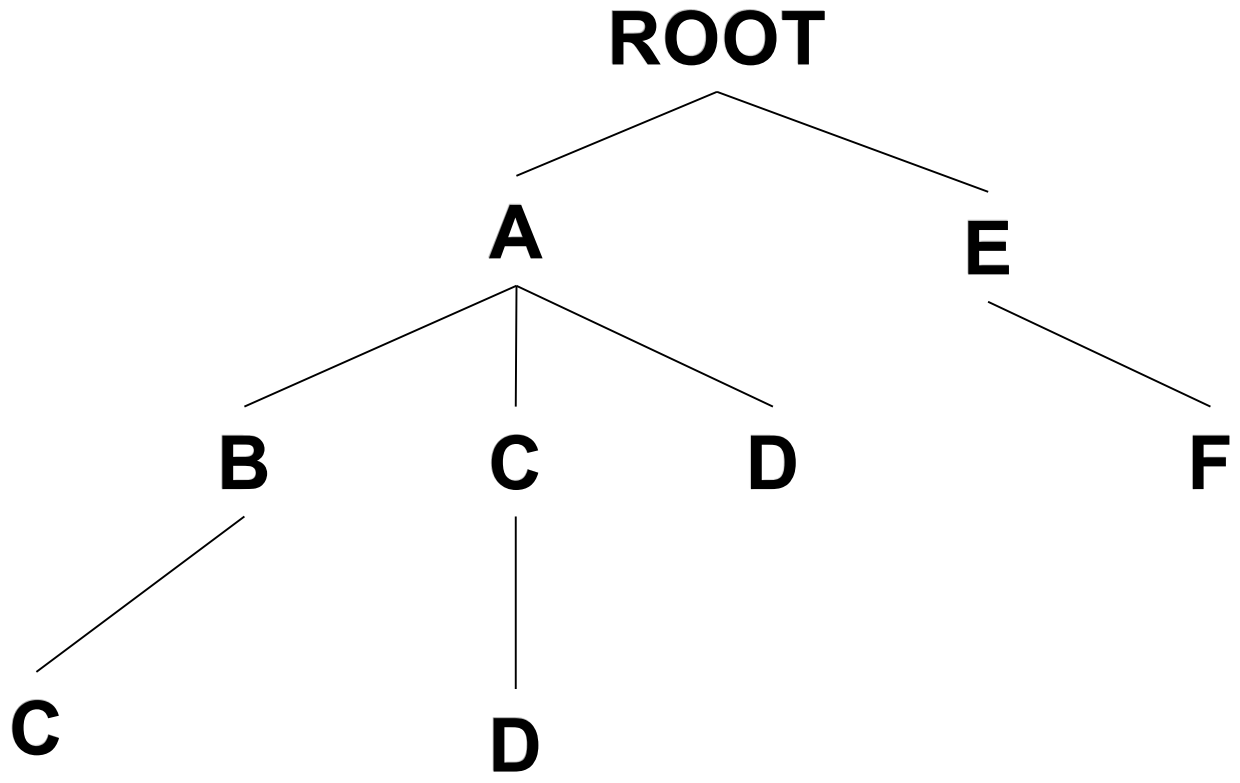
A B C
A C D
A D
E F

SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

Existing message types:

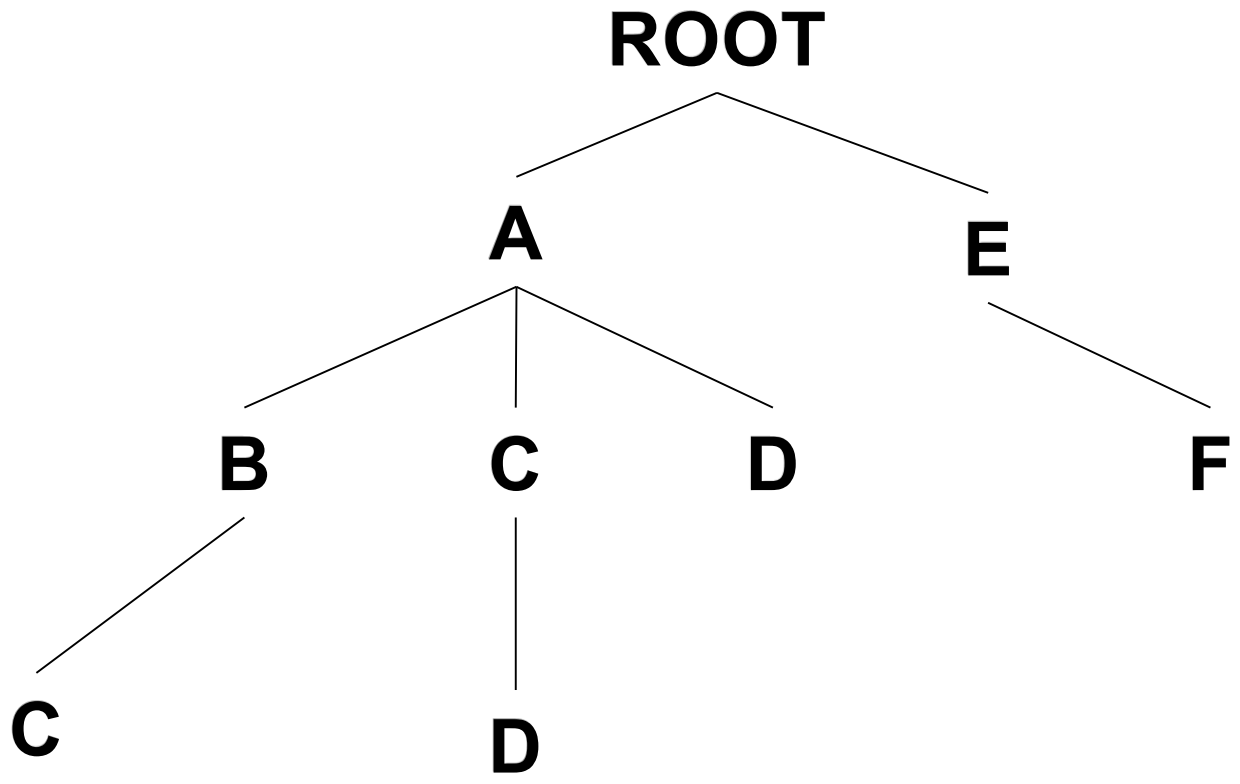
A B C
A C D
A D
E F



SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

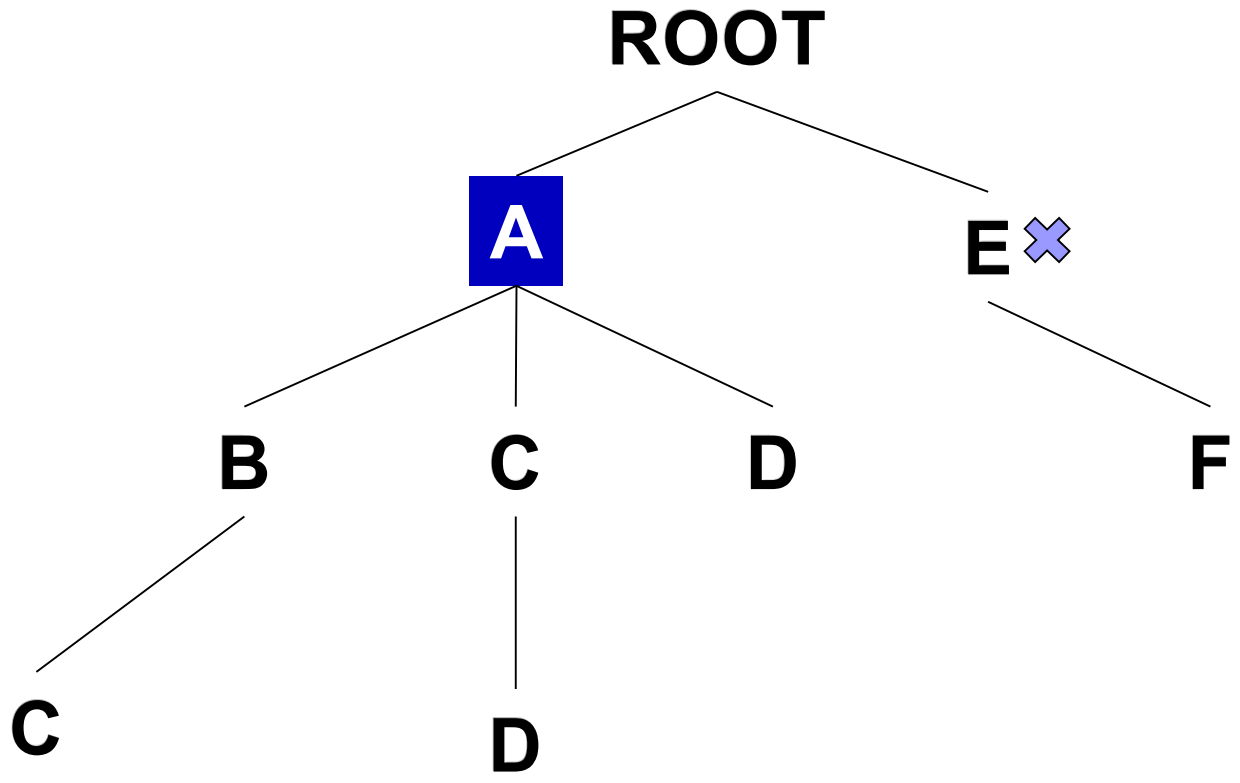
New log entry: *A B P C*



SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

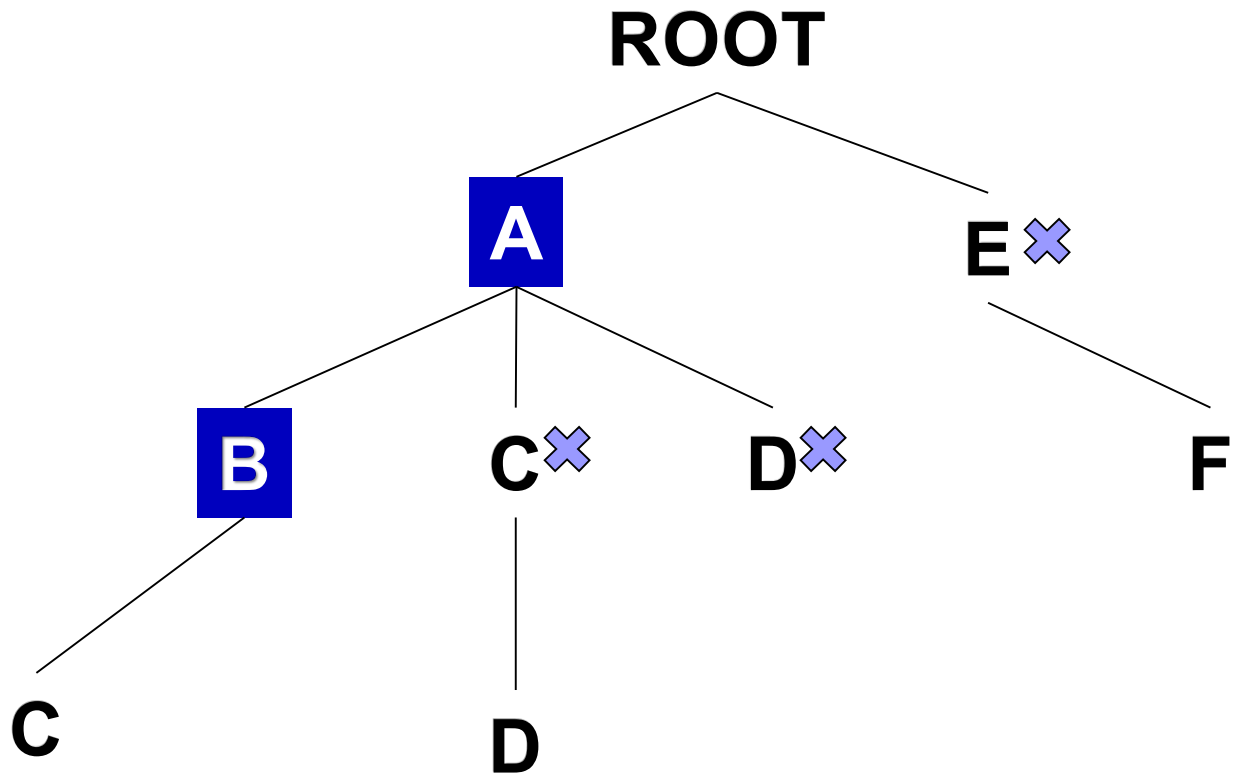
New log entry: **A** *B* *P* *C*



SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

New log entry: **A** **B** *P* *C*

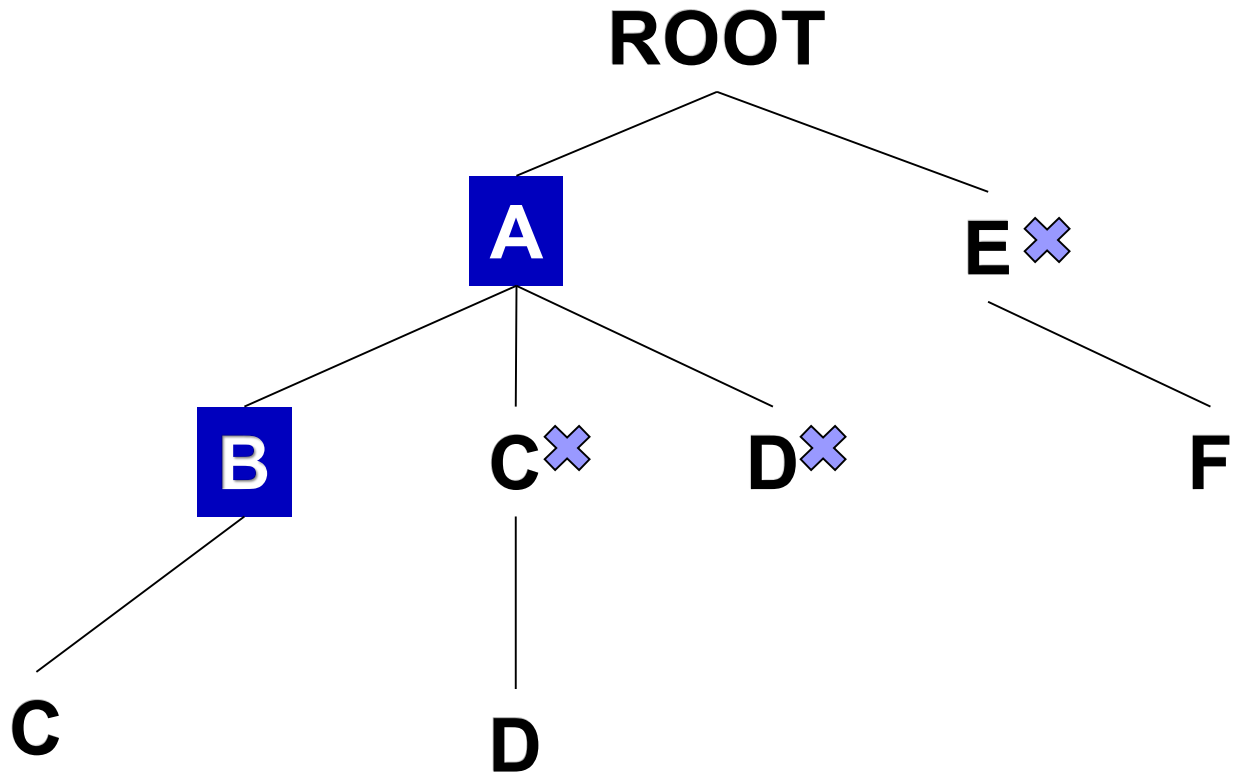


SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

New log entry: **A** **B** *P* *C*

Parameter:

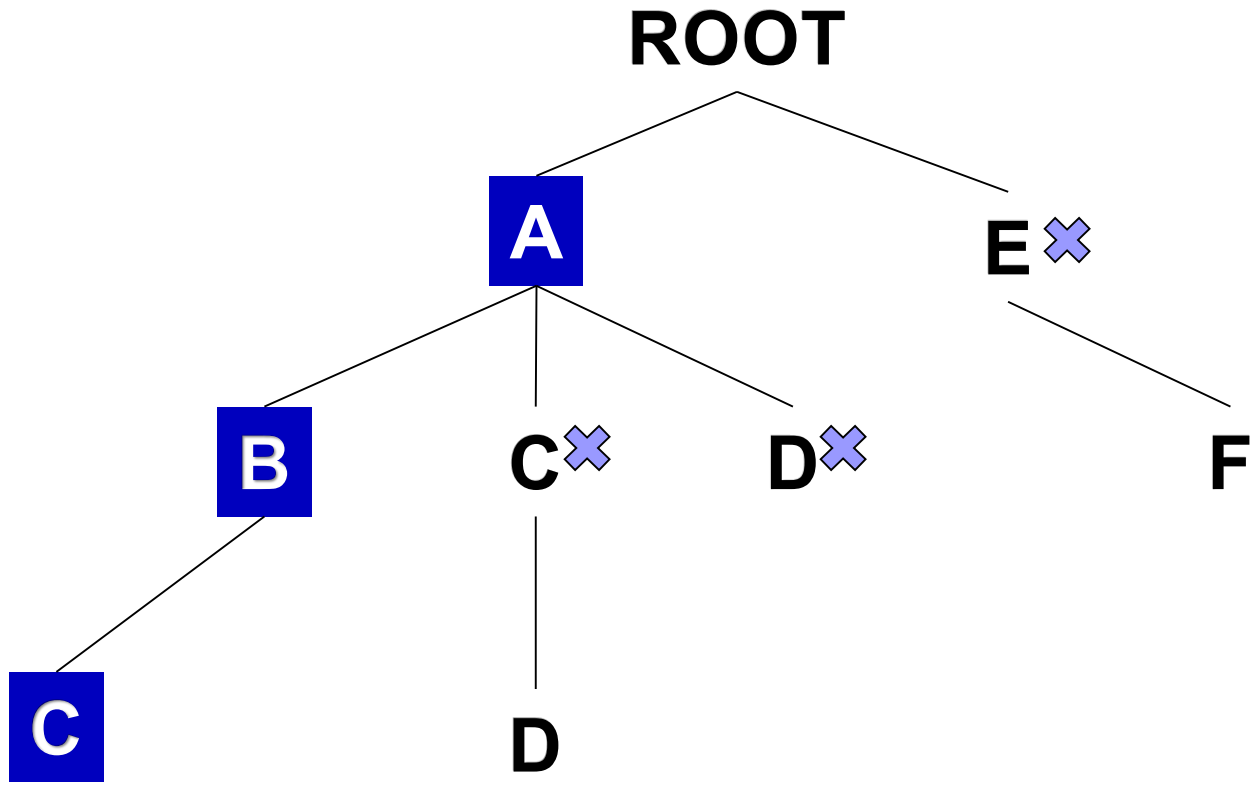


SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

New log entry: **A** **B** *P* **C**

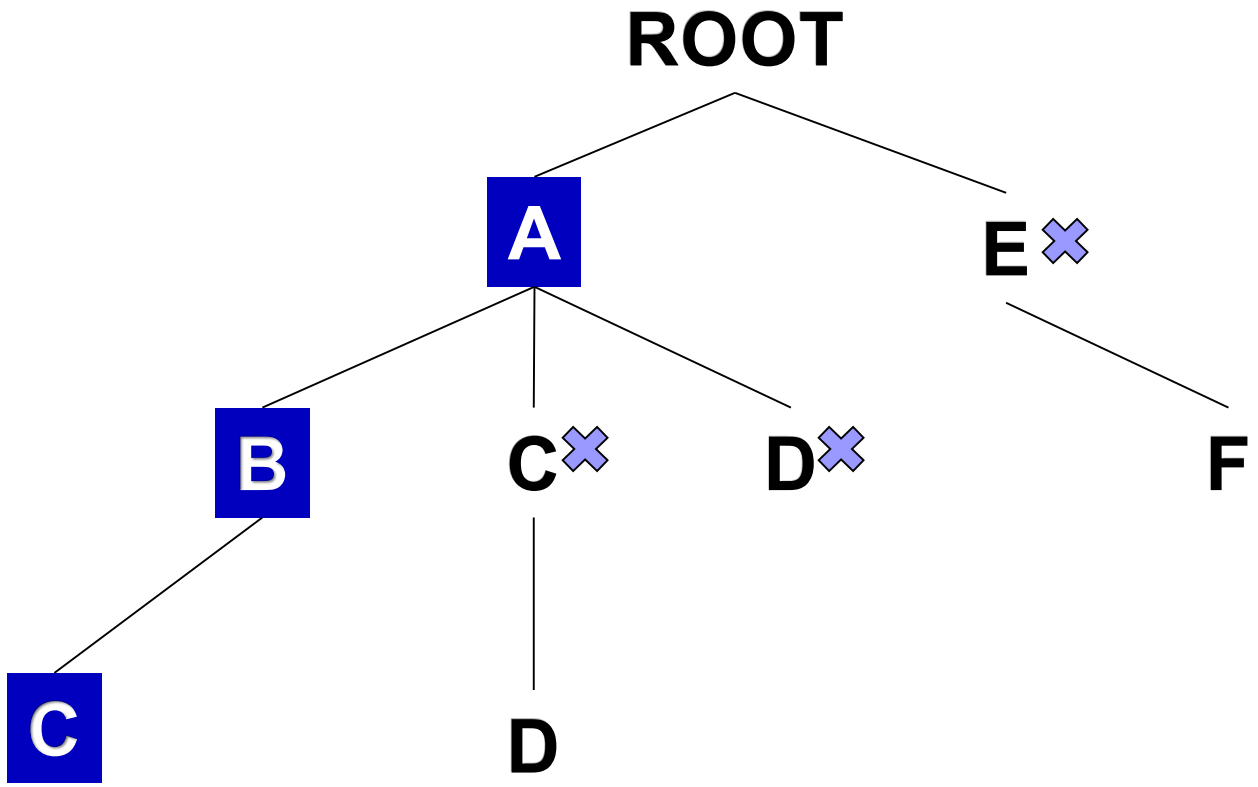
Parameter:



SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

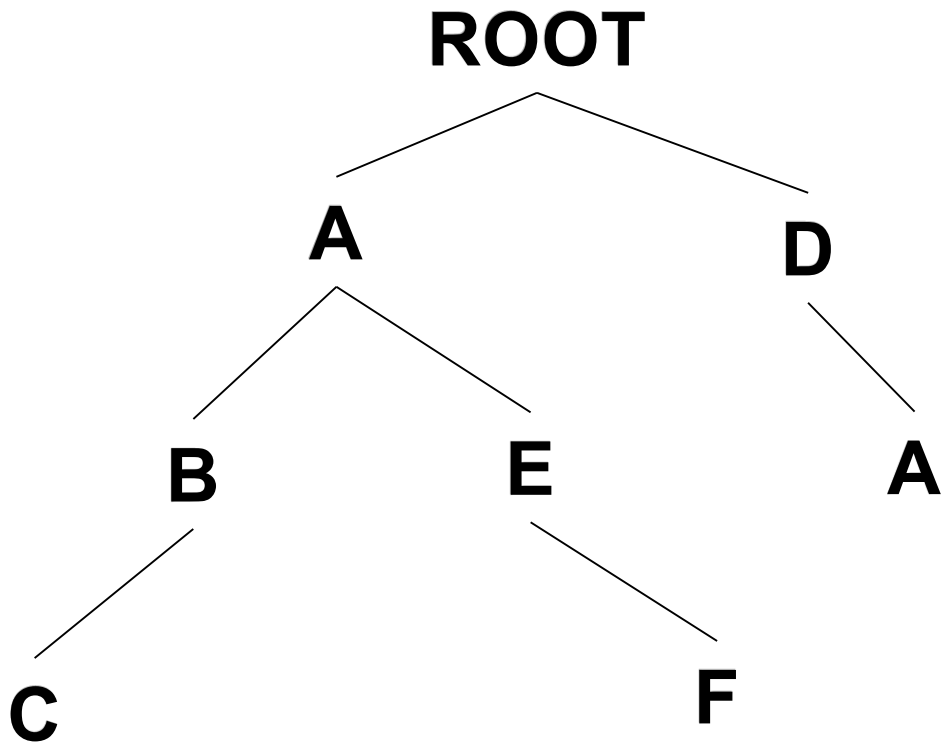
Time Complexity:
$O(n)$ for each log entry



SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

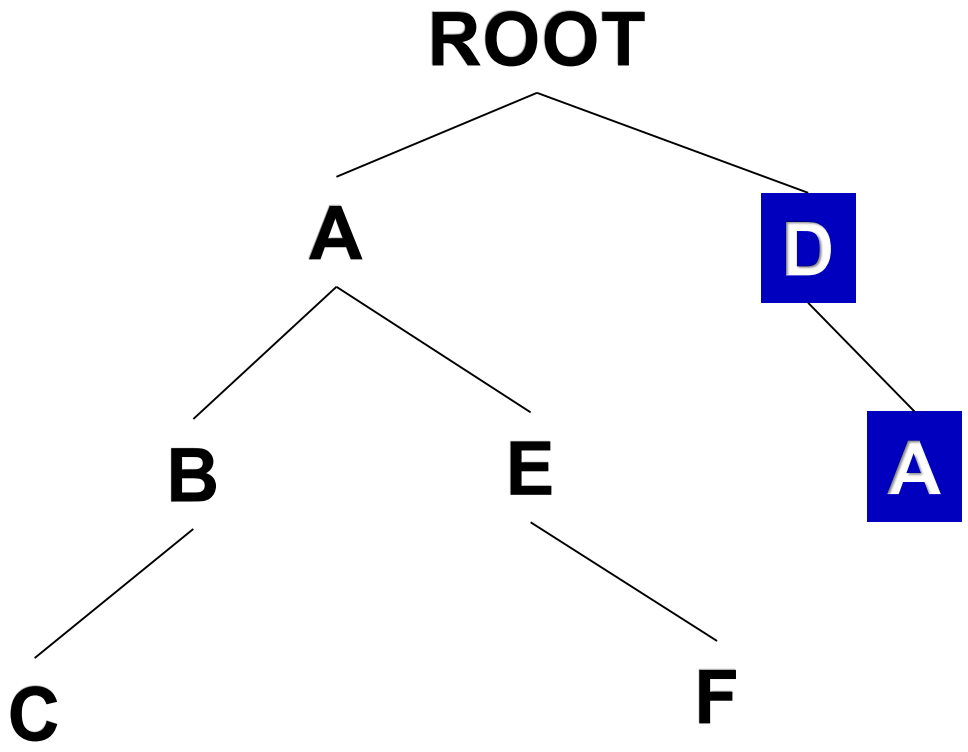
Problem:
New log entry: *D A P B C*



SPELL – Improvement on efficiency

Improvement 1: Prefix Tree

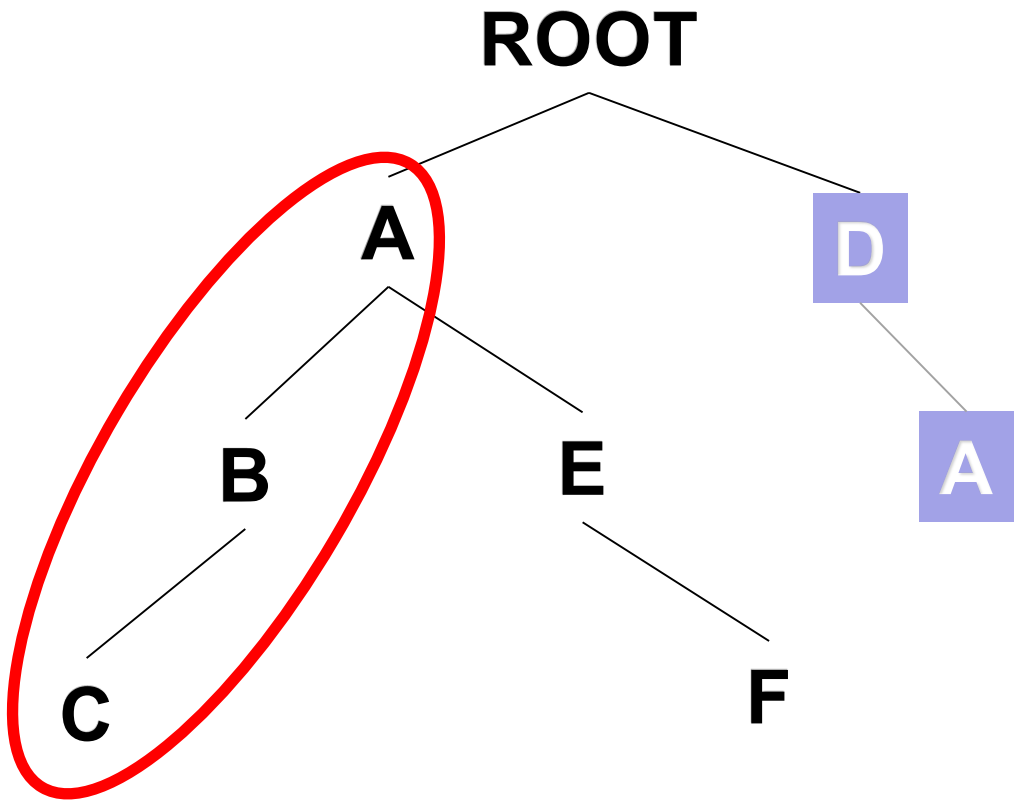
Problem:
New log entry: D A *P B C*
Matches D A



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

$Pointer\ P_m$

Message types:

[A B C]

[A E F]

[D A]

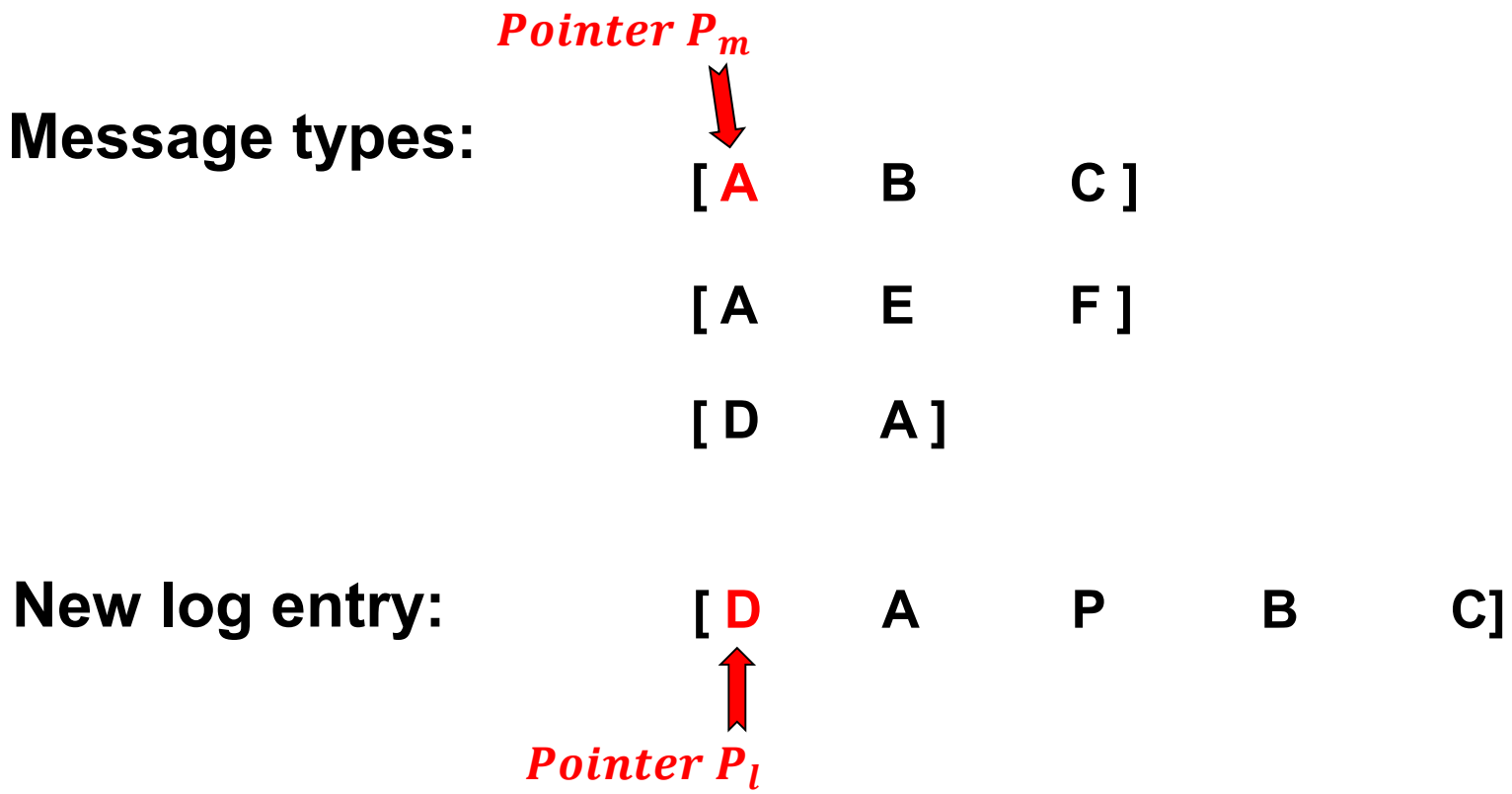
New log entry: [D A P B C]

$Pointer\ P_l$

SPELL – Improvement on efficiency

Improvement 2: Simple Loop

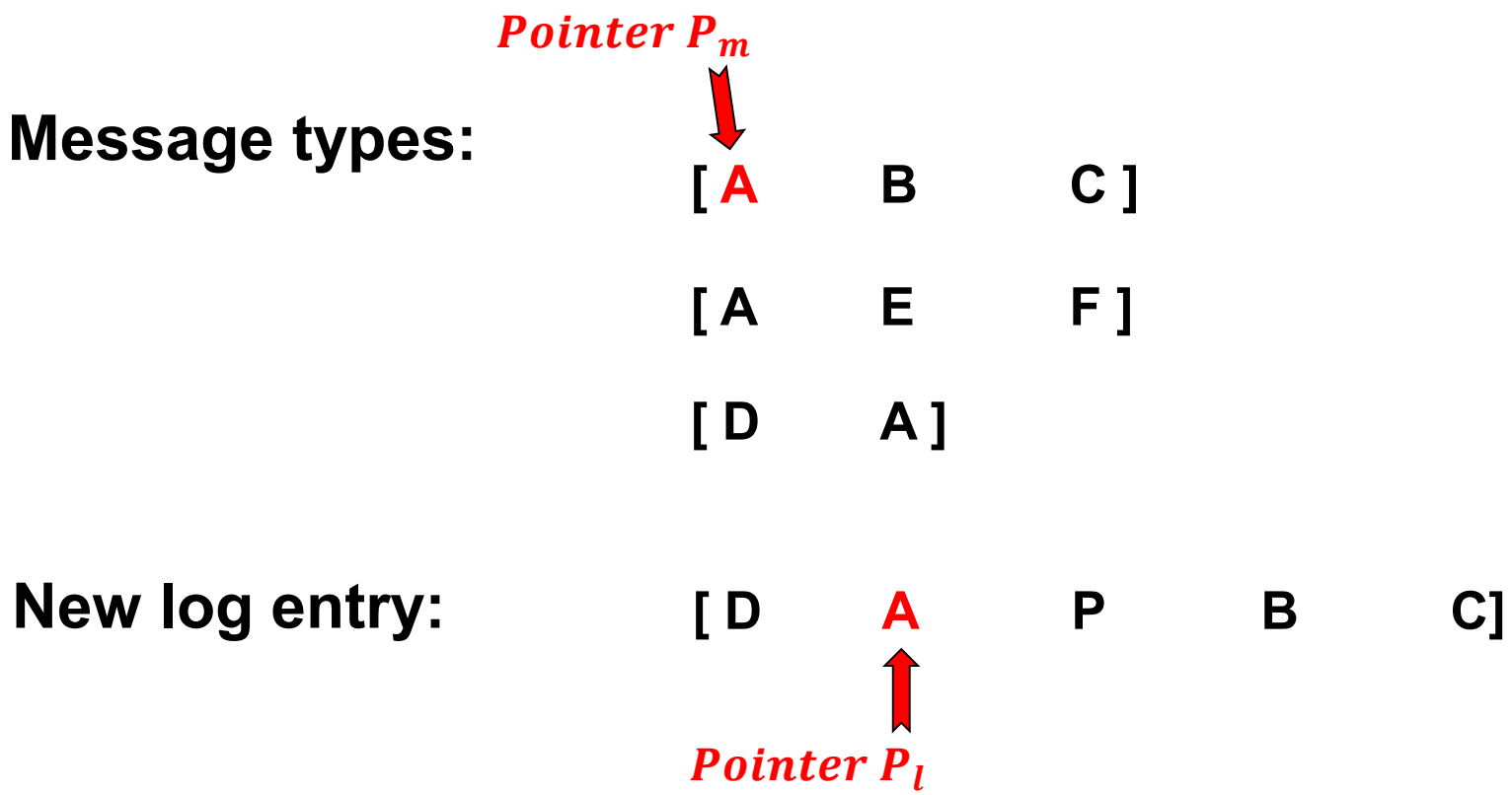
Compare each message type with new log entry



SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

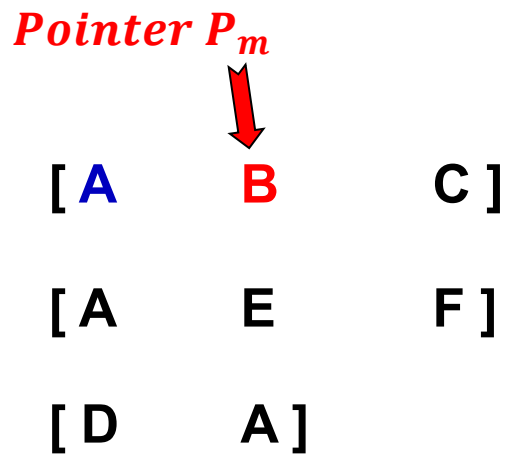


SPELL – Improvement on efficiency

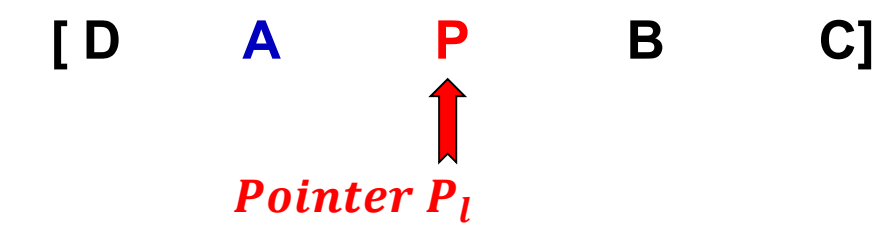
Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:



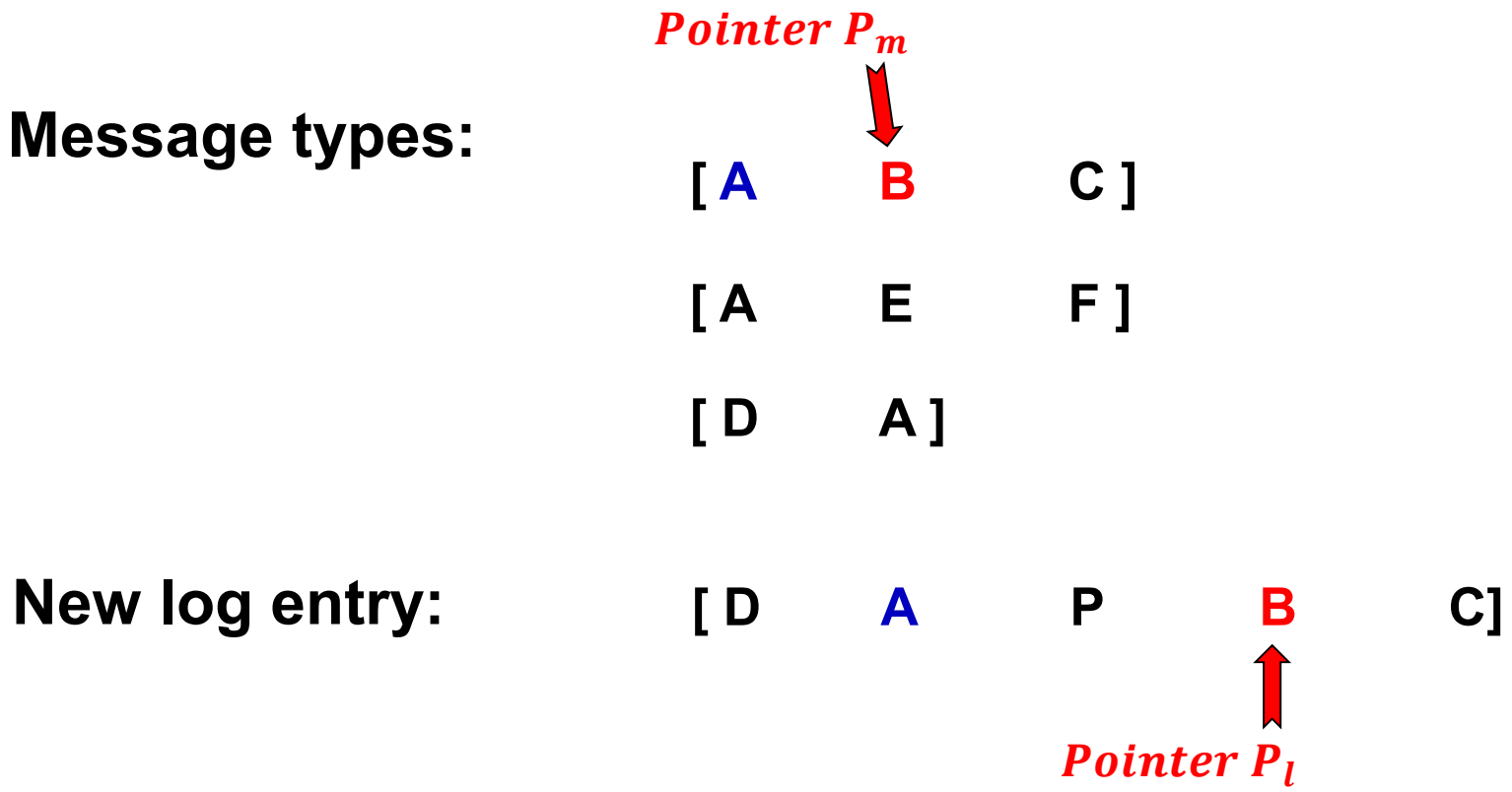
New log entry:



SPELL – Improvement on efficiency

Improvement 2: Simple Loop

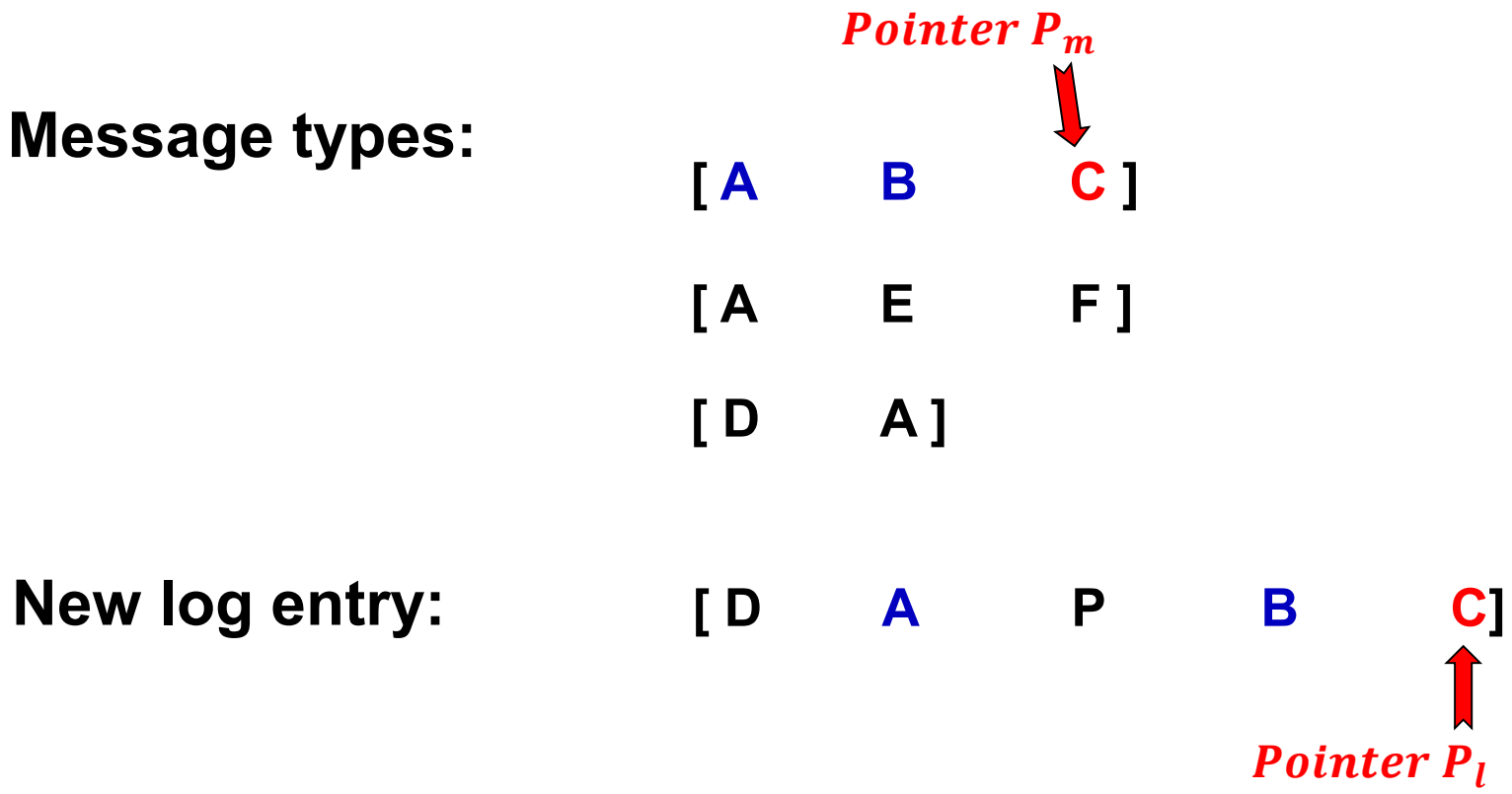
Compare each message type with new log entry



SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

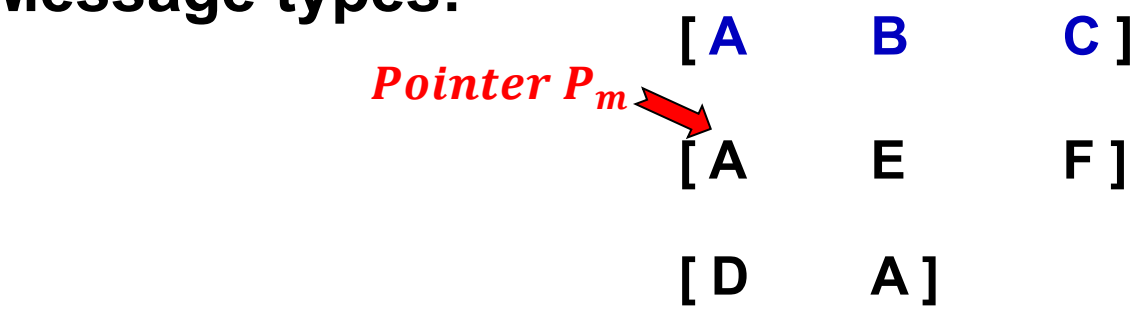


SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:



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:		Matched length:
	[A B C]	3
	[A E F]	N/A
	[D A]	2

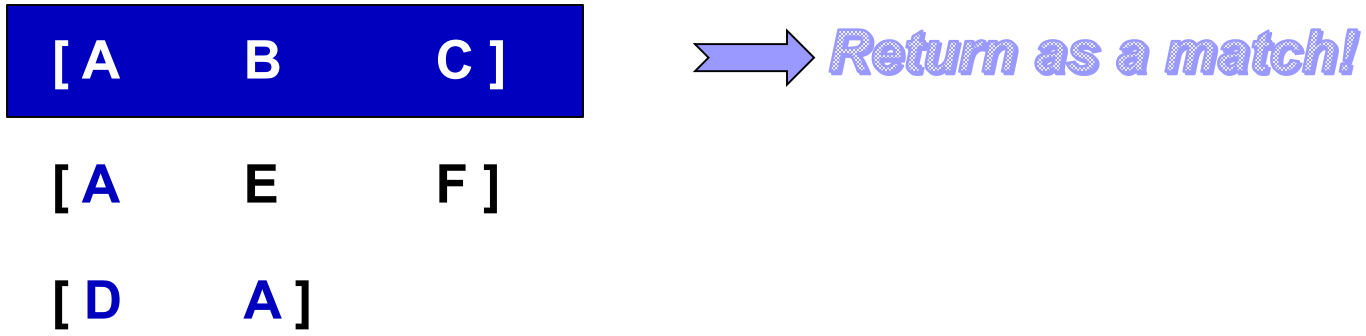
New log entry: [D A P B C]

SPELL – Improvement on efficiency

Improvement 2: Simple Loop

Compare each message type with new log entry

Message types:



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:

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

New log entry: [D A P B C]

Time complexity

$$O(m\,n)$$

Number of message types

Log entry length

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]

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

Time complexity

$$O(m\,n)$$

Number of message types

Log entry length

Evaluation

Methods to compare:

IPLoM (Makanju'KDD09):

Partition log file using 3-step heuristics (log entry length, etc.)

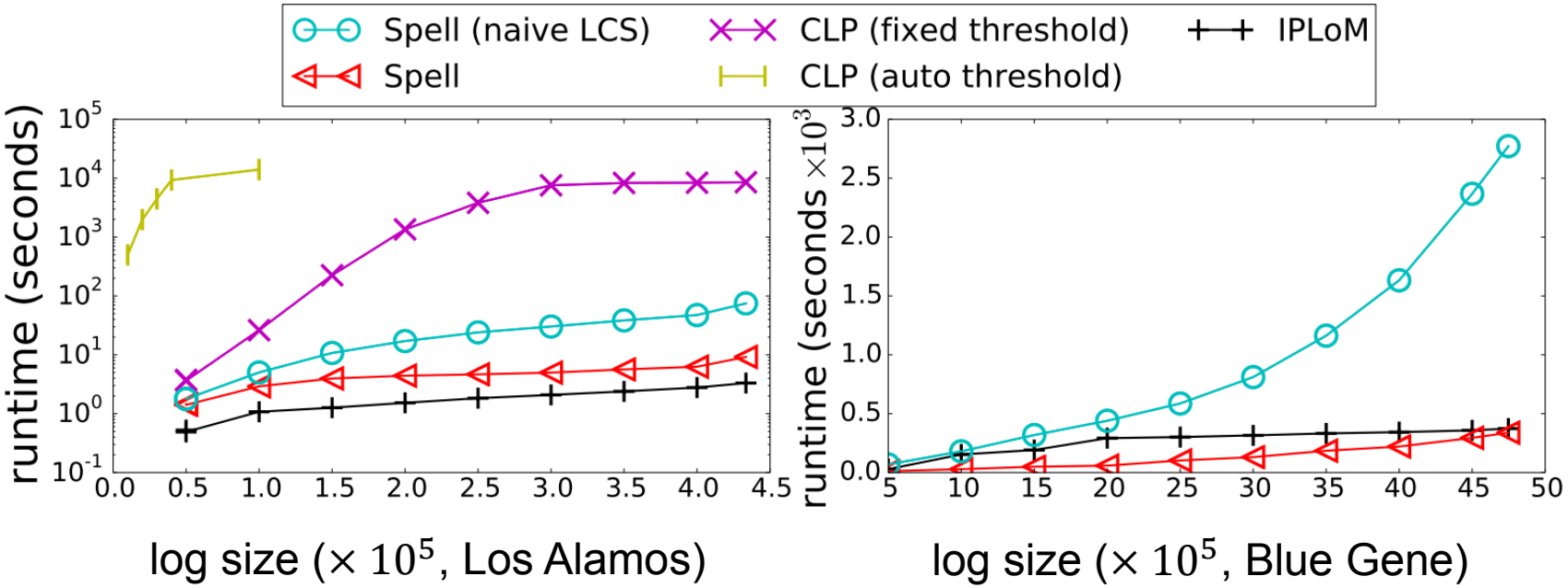
CLP (Fu'ICDM09)

Cluster similar logs together based on weighted edit distance

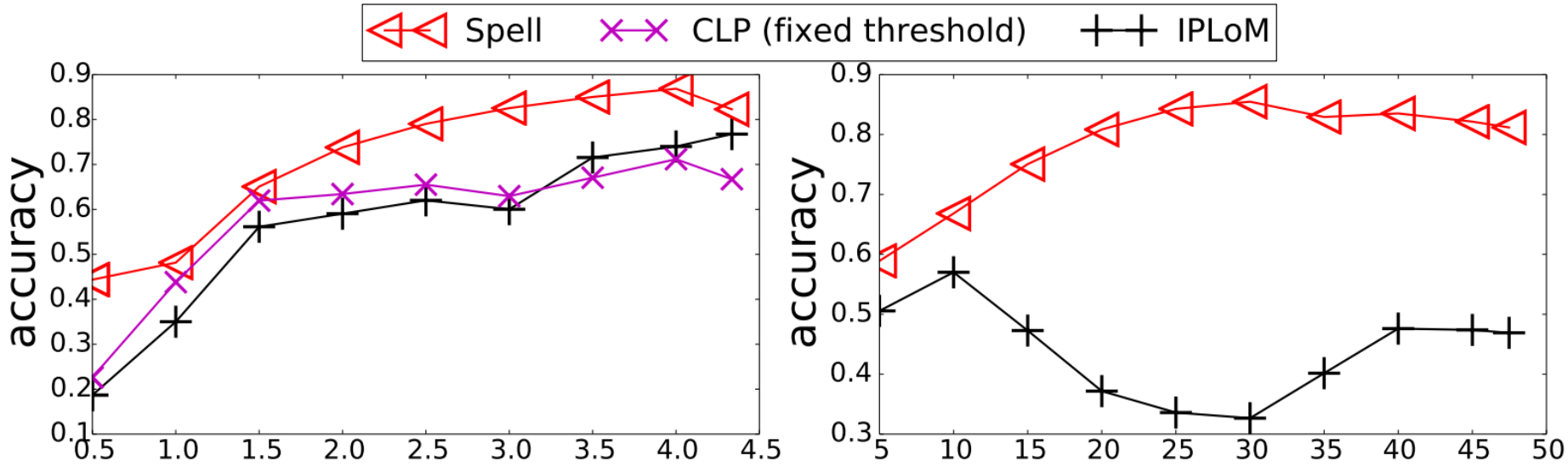
Log dataset:

Log type	Count	Message type ground truth
Los Alamos HPC log	433,490	Available online
BlueGene/L log	4,747,963	Available online

Evaluation - Efficiency



Evaluation - Effectiveness



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

	Los Alamos HPC log	BlueGene/L log
prefix tree	397,412 (91.68%)	4,457,719 (93.89%)
simple loop	35,691 (8.23%)	288,254 (6.07%)
naive LCS	387 (0.09%)	1,990 (0.042%)

AMORTIZED COST OF EACH MESSAGE TYPE LOOKUP STEP IN Spell

	Los Alamos HPC log	BlueGene/L log
prefix tree (ms)	0.006	0.011
simple loop (ms)	0.020	0.087
naive LCS (ms)	0.175	0.580

Evaluation - Effectiveness

COMPARISON OF Spell WITH AND WITHOUT PRE-FILTER

Spell$_I$	Los Alamos HPC log		BlueGene/L log	
With pre-filtering	True message types found	Accuracy	True message types found	Accuracy
False	55	0.822786	165	0.811798
True	55	0.822786	164	0.811791