Trace Analysis With Critical Path Sequences

Madeline Janecek

Naser Ezzati-Jivan

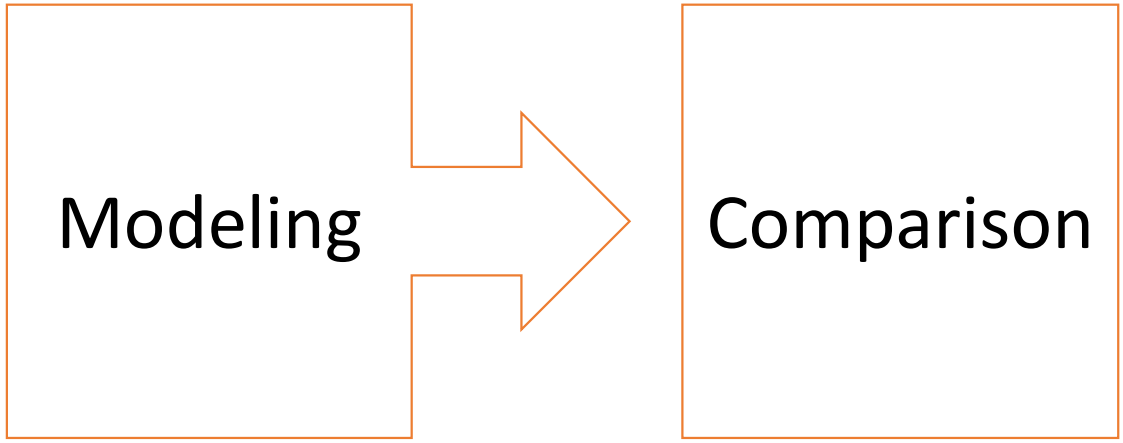
https://youtu.be/k2P7cPSQi_A

Motivation

- Identify the root cause of abnormal executions
- An execution's critical path is a good candidate for this analysis
  - shows interactions between threads
  - shows execution time and how its split up between different states

Root Cause Analysis

- Modeling
  - Feature based modeling
    - recently submitted work to The 32nd International Symposium on Software Reliability Engineering (ISSRE 2021)
  - Sequence based modeling

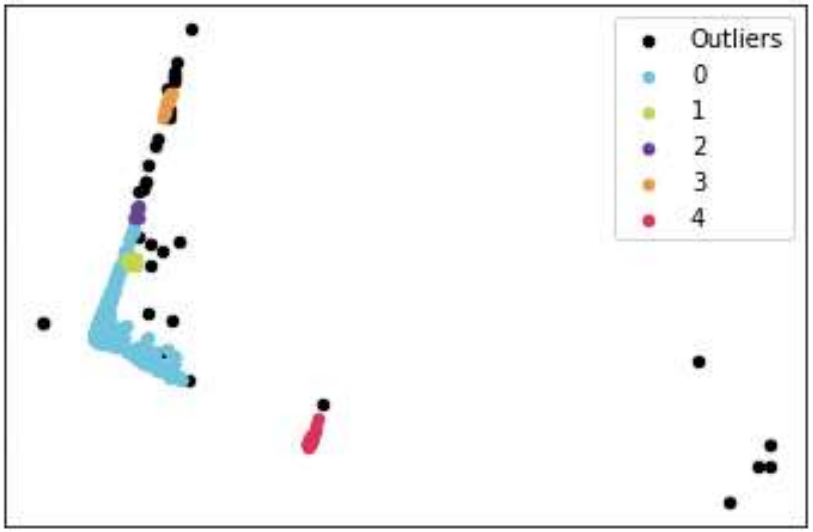


Past Work…

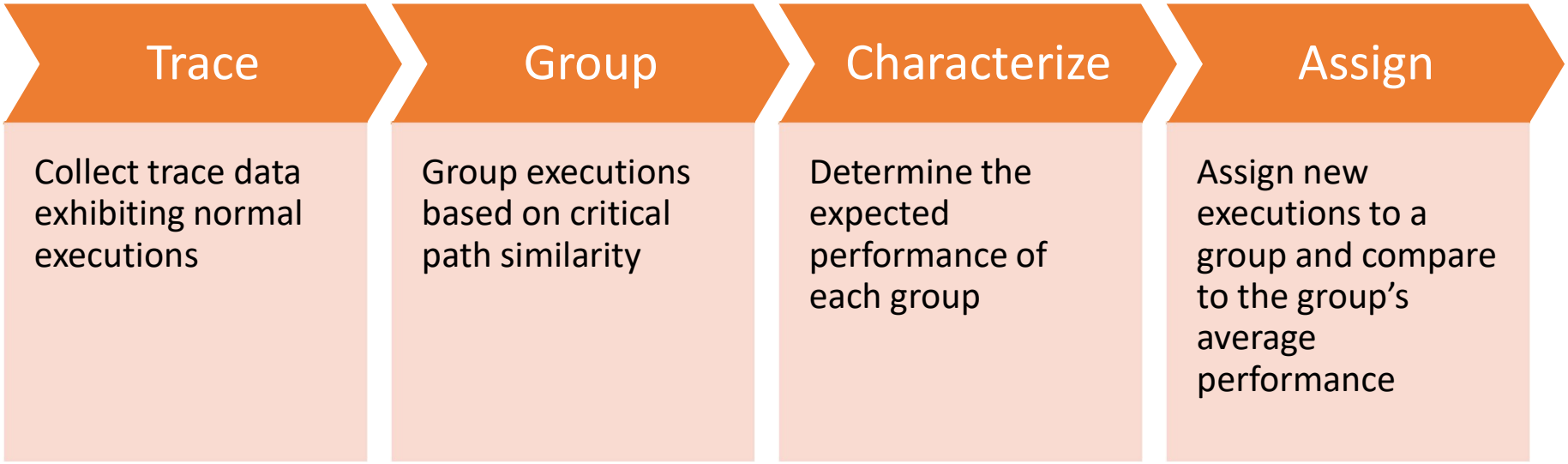
Critical path vectorization and clustering

State	Index	Duration
SYSTEMCALL	1	d1
USERMODE	2	d2
SYSTEMCALL	1	d3
BLOCKED_WAITPROCESS	6	d4
BLOCKED_CPU	4	d5
SYSTEMCALL	1	d6

Count	0	3	1	0	1	0	1	0	0	0
Duration	0	d1+d3+d6	d2	0	d5	0	d4	0	0	0



Methodology

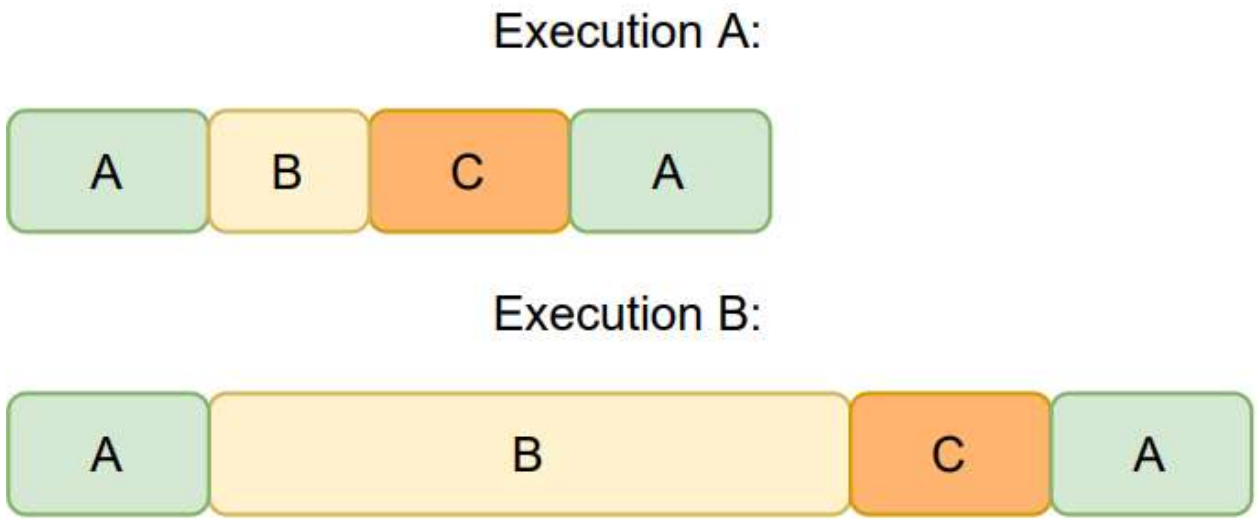


Challenge

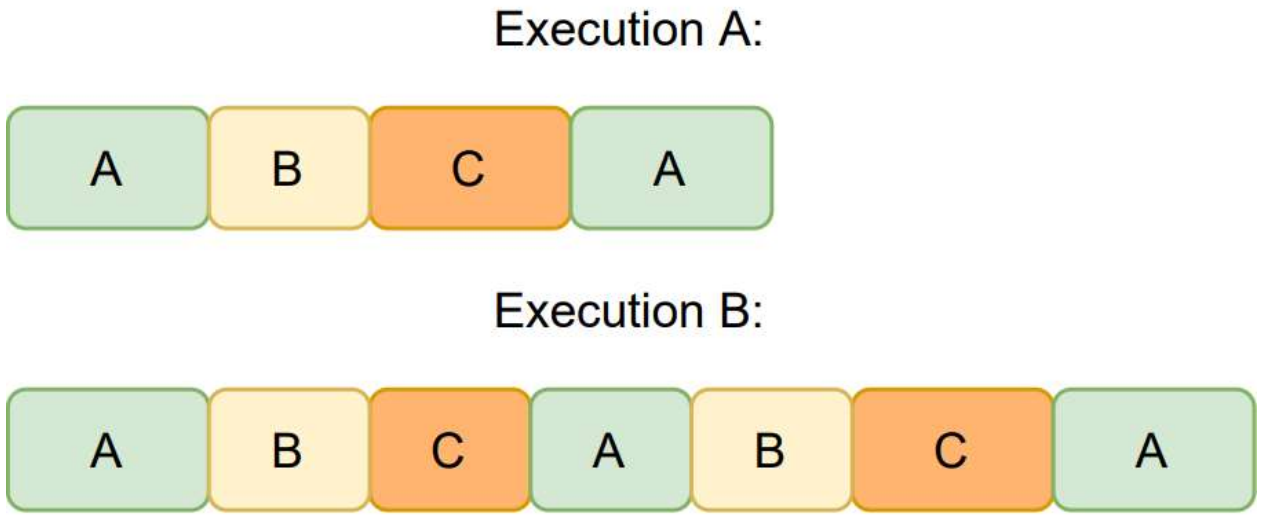
A single model of normal execution risks being overgeneralized

Two normal executions may produce different critical paths

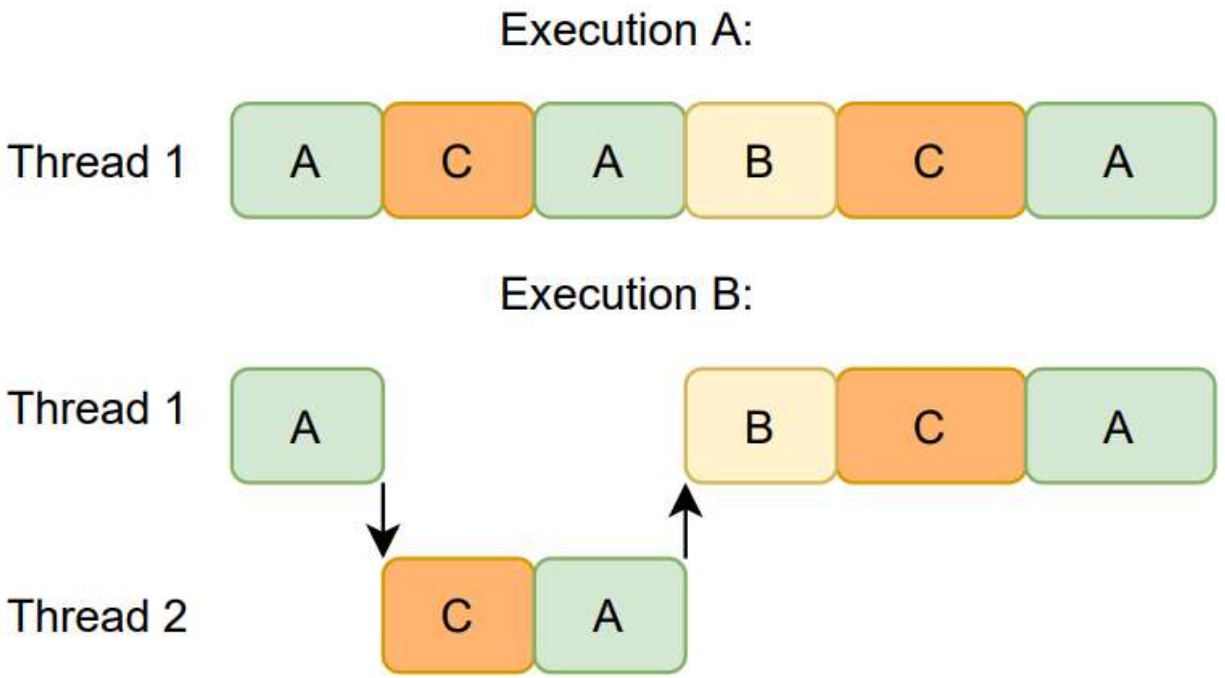
Timing Differences



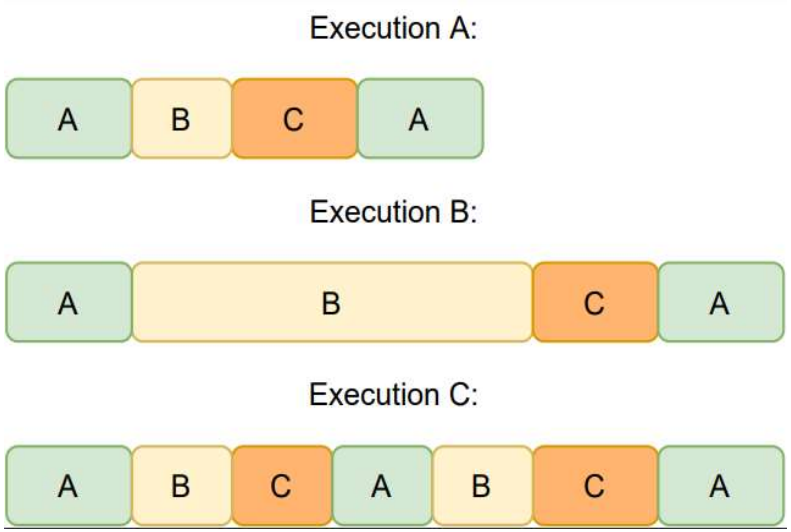
Structural Differences



Different Topologies



String Representation



A B C A

A B C A

A B C A B C A

Grouping

- Executions that produce a string similarity score greater than a predefined threshold are grouped together
- Executions with similar critical path sequences should perform similarly

Jaro-Winkler Distance*

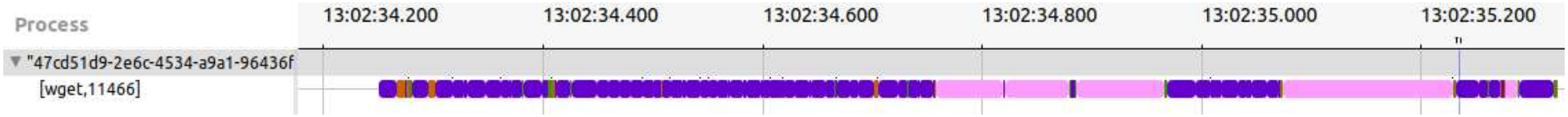
A string similarity metric that results in a similarity score ranging from 0 to 1

This is calculated using:

- The number of matching characters
- The number of transpositions
- The two strings' length
- The length of the matching prefix

*https://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance

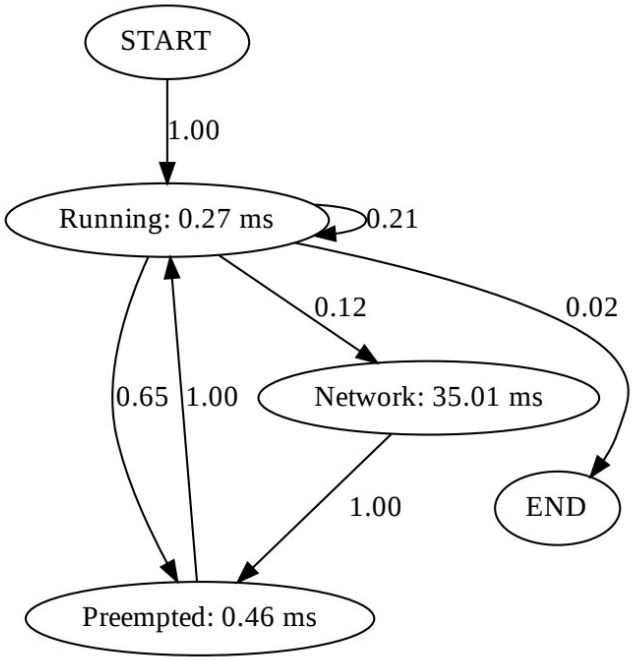
Example – wget

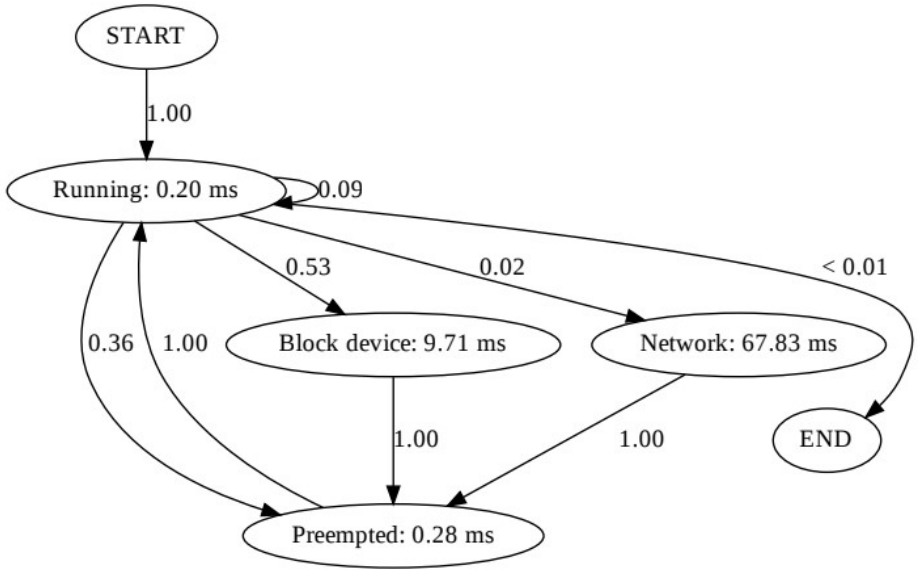



```
Total number of critical path groups found: 5
Executions in each group:
Group 1: 63.79% of executions
Group 2: 0.69% of executions
Group 3: 23.41% of executions
Group 4: 4.56% of executions
Group 5: 7.44% of executions
```

- We collected data from over 1000 executions of wget
- At a 90% threshold, 5 groups were identified

Example - wget



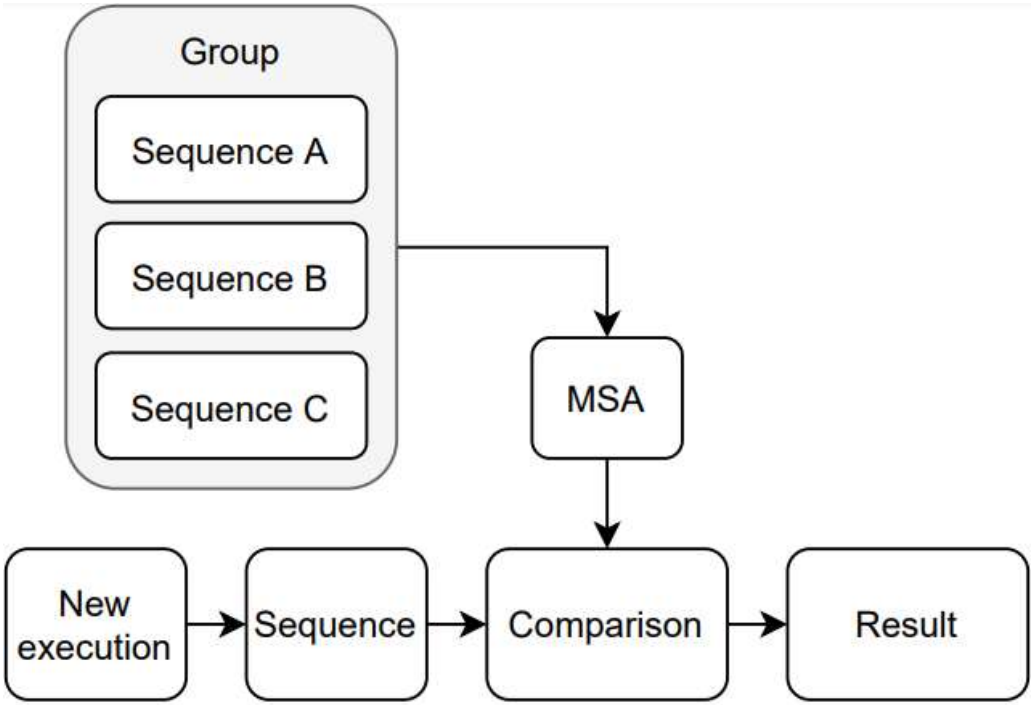


Characterization and Comparison

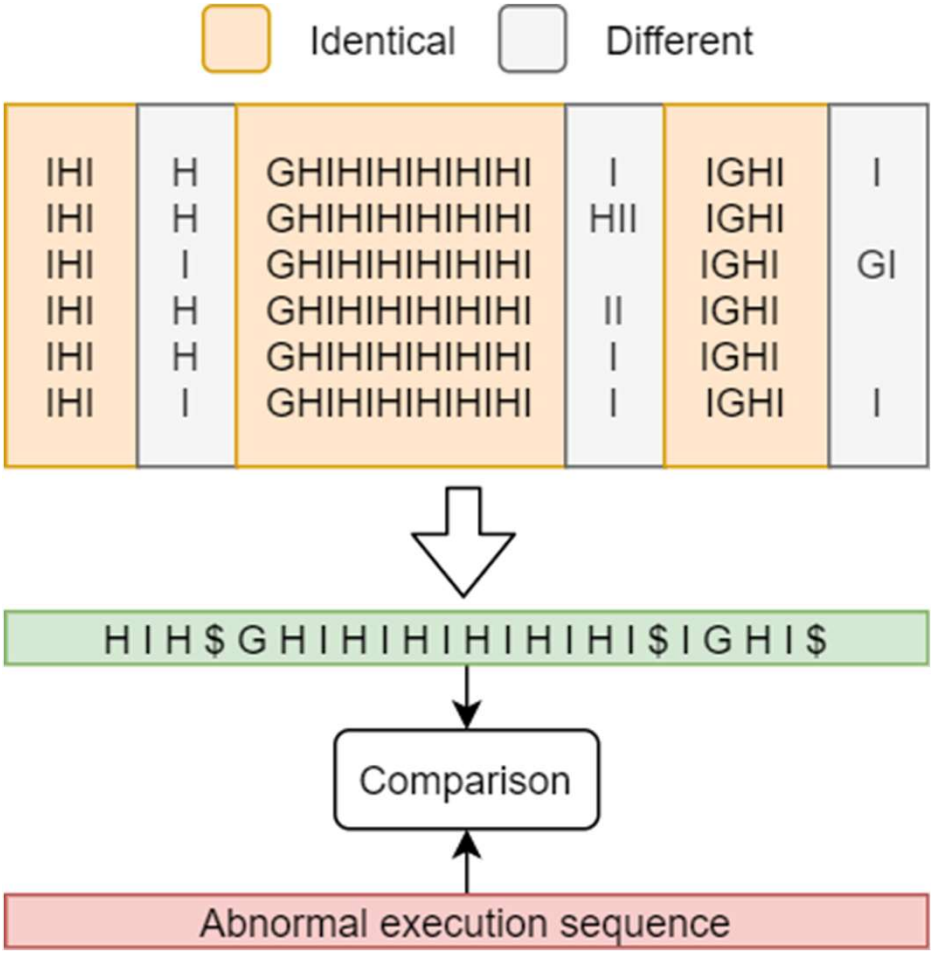
```
Assigning trace to group...
Trace has been assigned to Group 1 with a score of 93.05% similarity.

Examining group stats...
Execution's duration was within the group's normal range.
Trace times by critical path state (compared to group's average):
BLOCK_DEVICE 0.0000 ms (-0.0506)
BLOCKED      0.0000 ms (+0.0000)
DEFAULT      0.0000 ms (+0.0000)
EPS          0.0000 ms (+0.0000)
INTERRUPTED  0.0000 ms (+0.0000)
IPI          0.0000 ms (+0.0000)
NETWORK      169.5306 ms (-0.8702)
PREEMPTED    6.6214 ms (-17.2806)
RUNNING      11.6186 ms (-0.7740)
TIMER        0.0000 ms (-1.4774)
UNKNOWN      0.0000 ms (+0.0000)
USER_INPUT   0.0000 ms (+0.0000)
TOTAL        187.7706 ms (-20.4529)
```

Multiple Sequence Alignment

- MSA is used in bioinformatics to identify regions of similarity between sequences of DNA, RNA or proteins



Multiple Sequence Alignment



Demo

Conclusion and Future Work

- Our future work will include fully implementing this process in Trace Compass
- All work is available: https://github.com/janecekm/CriticalPathSequenceAnalysis