



Finding a Needle in a Haystack

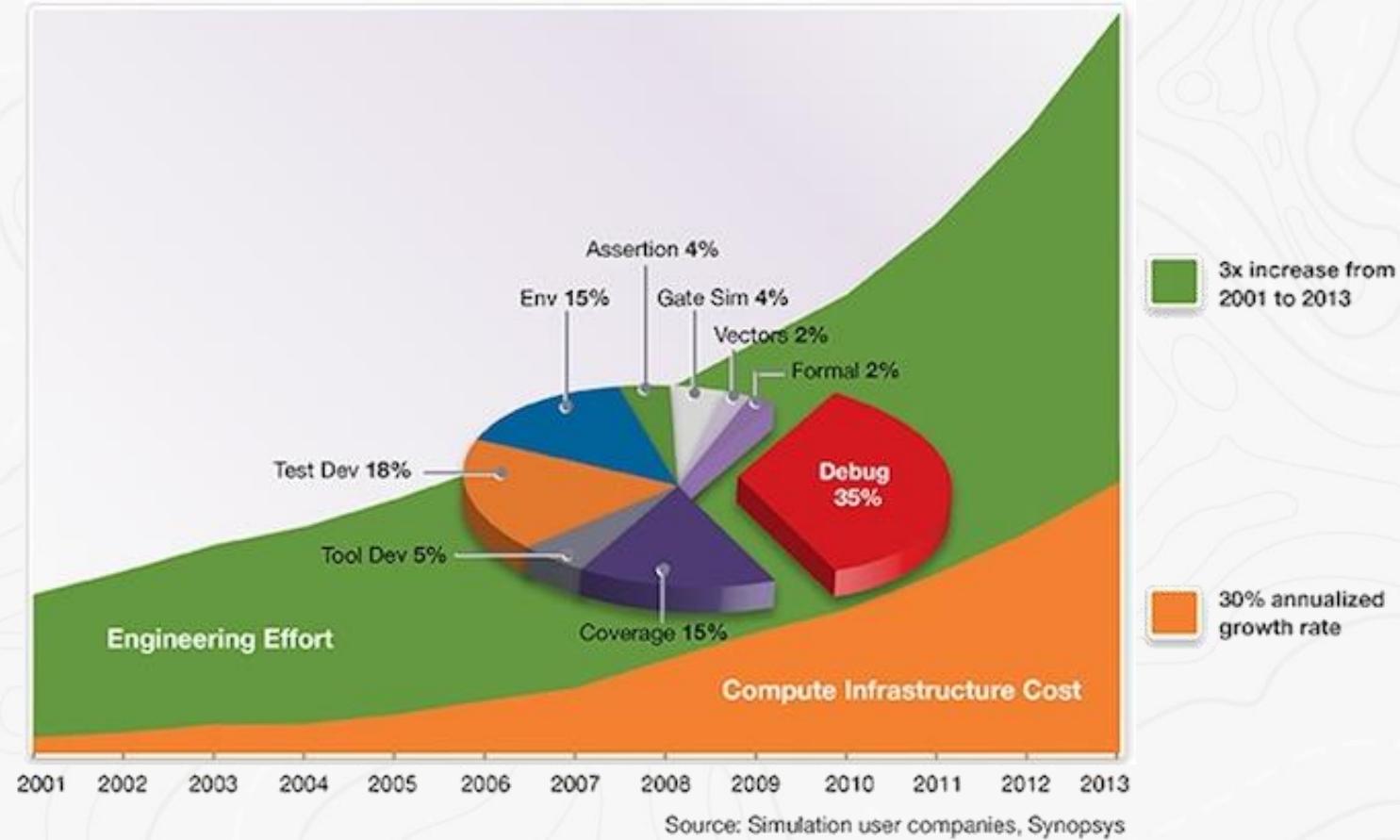
: A Novel Log Analysis Method with Test Clustering in Distributed Systems

Jin Choi

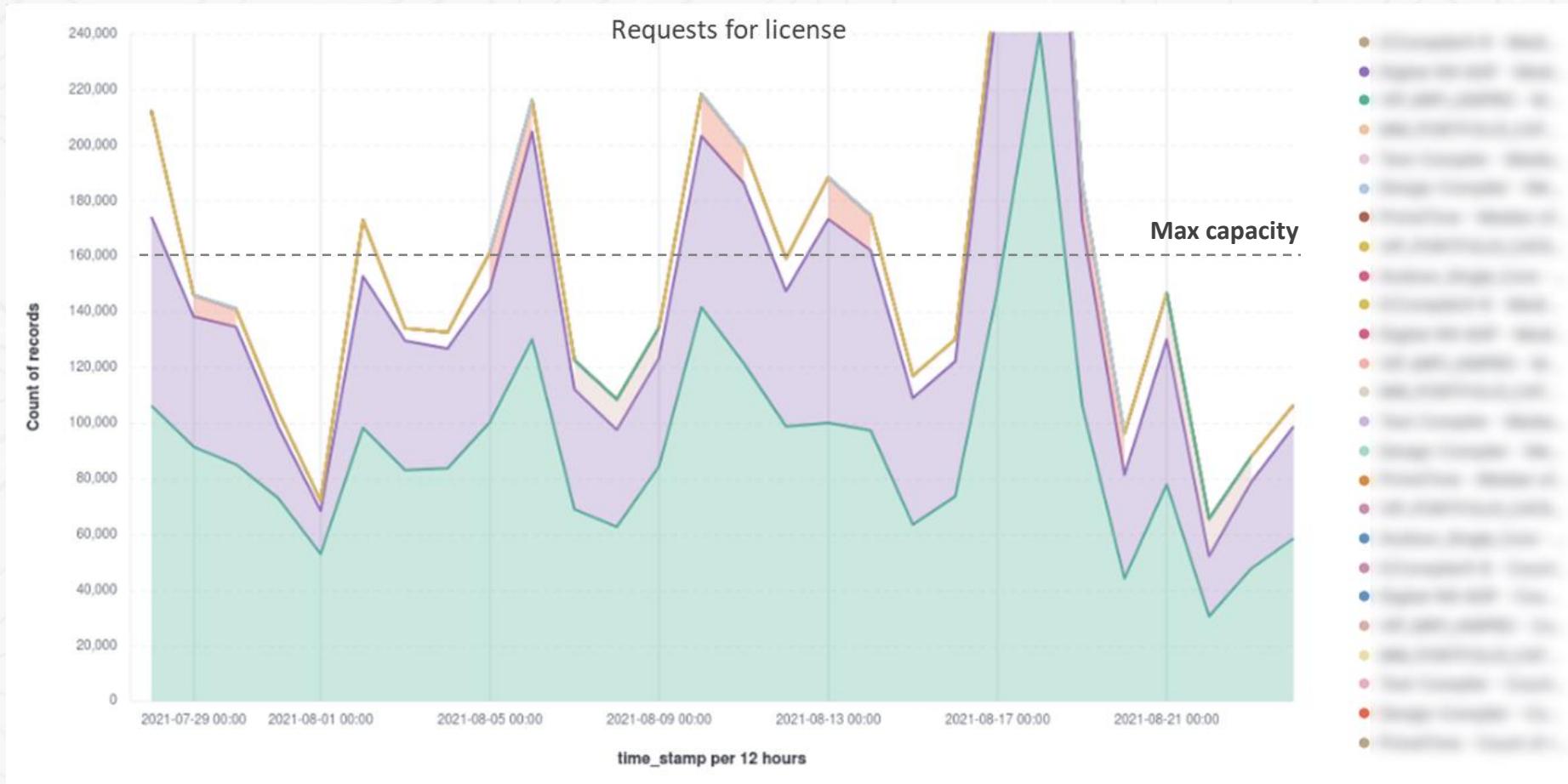
Samsung electronics co. ltd.



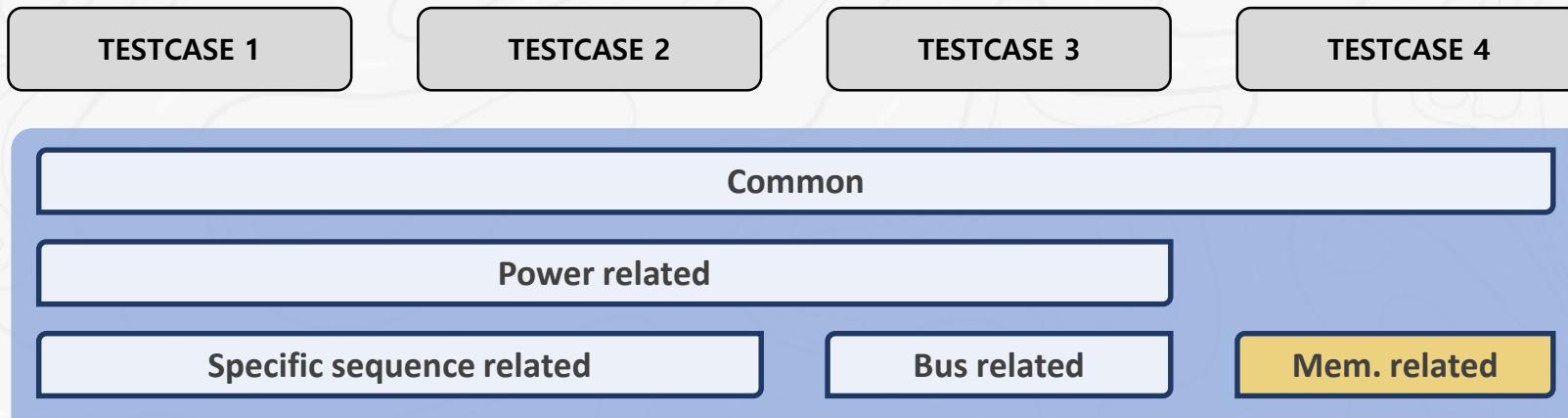
Intro – The growth cost of verification



Problem statement – Resource Usage



Problem statement – Repetitive debugging

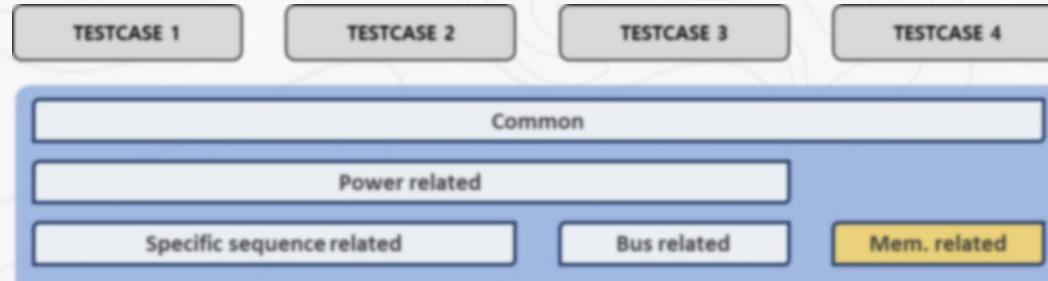


Debugging
the same error
AGAIN ?



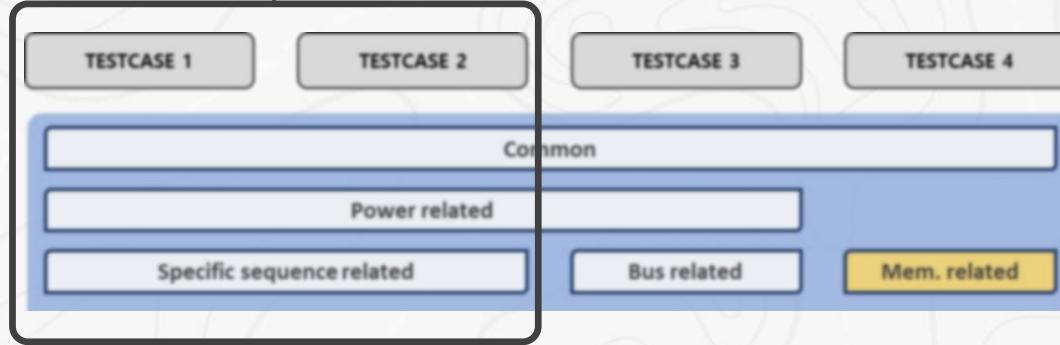
It's too LATE for
debugging
CRITICAL error...

Let's assist verification engineer !



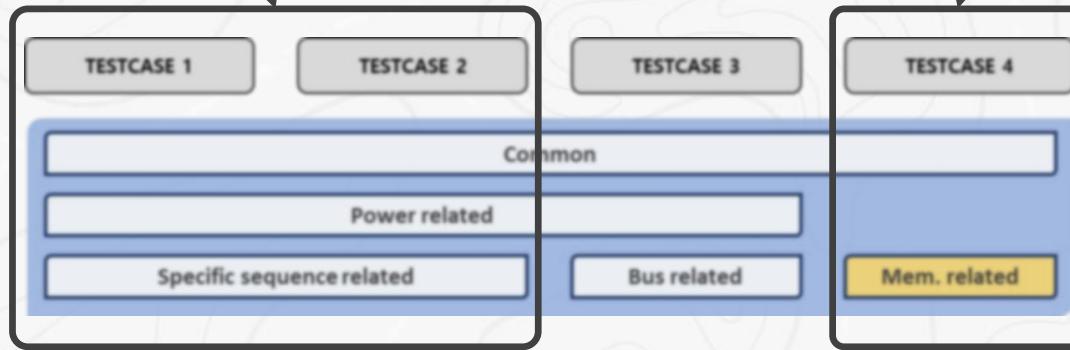
Let's assist verification engineer !

Grouping similar test cases
to prevent repetitive debugging!



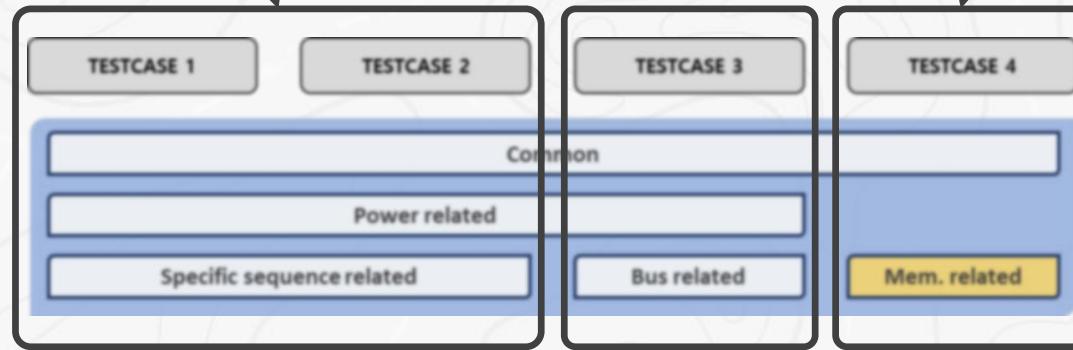
Let's assist verification engineer !

Grouping similar test cases
to prevent repetitive debugging!



Let's assist verification engineer !

Grouping similar test cases
to prevent repetitive debugging!



Prioritize the important test
case or dissimilar test cases!

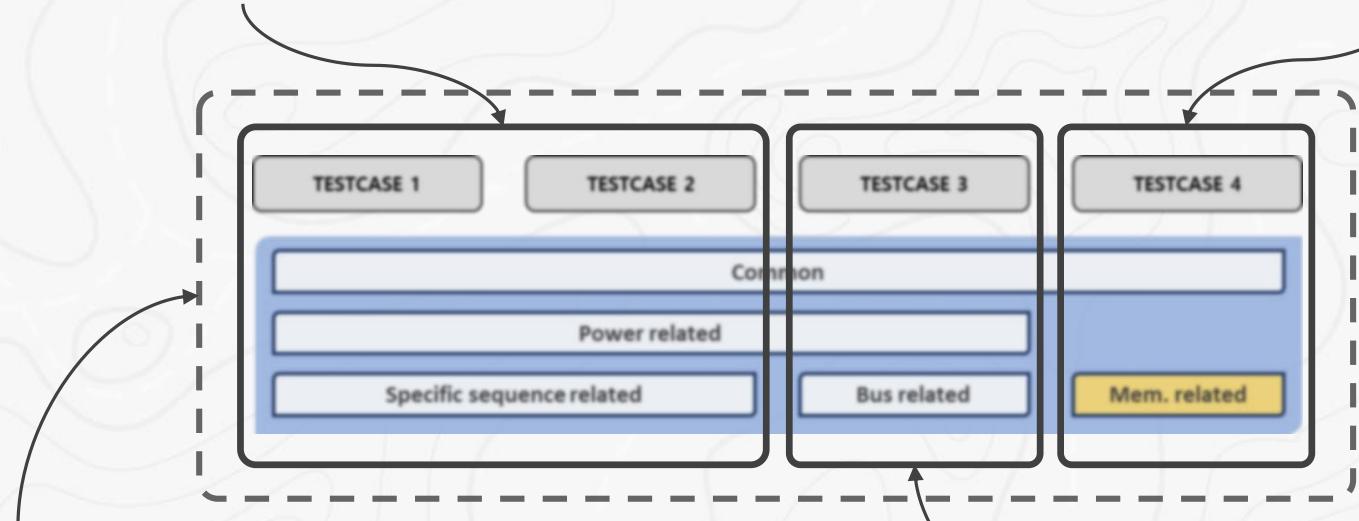


Automatically run low-
prioritized testcases when
resource is idle

Let's assist verification engineer !

Grouping similar test cases
to prevent repetitive debugging!

Prioritize the important test
case or dissimilar test cases!

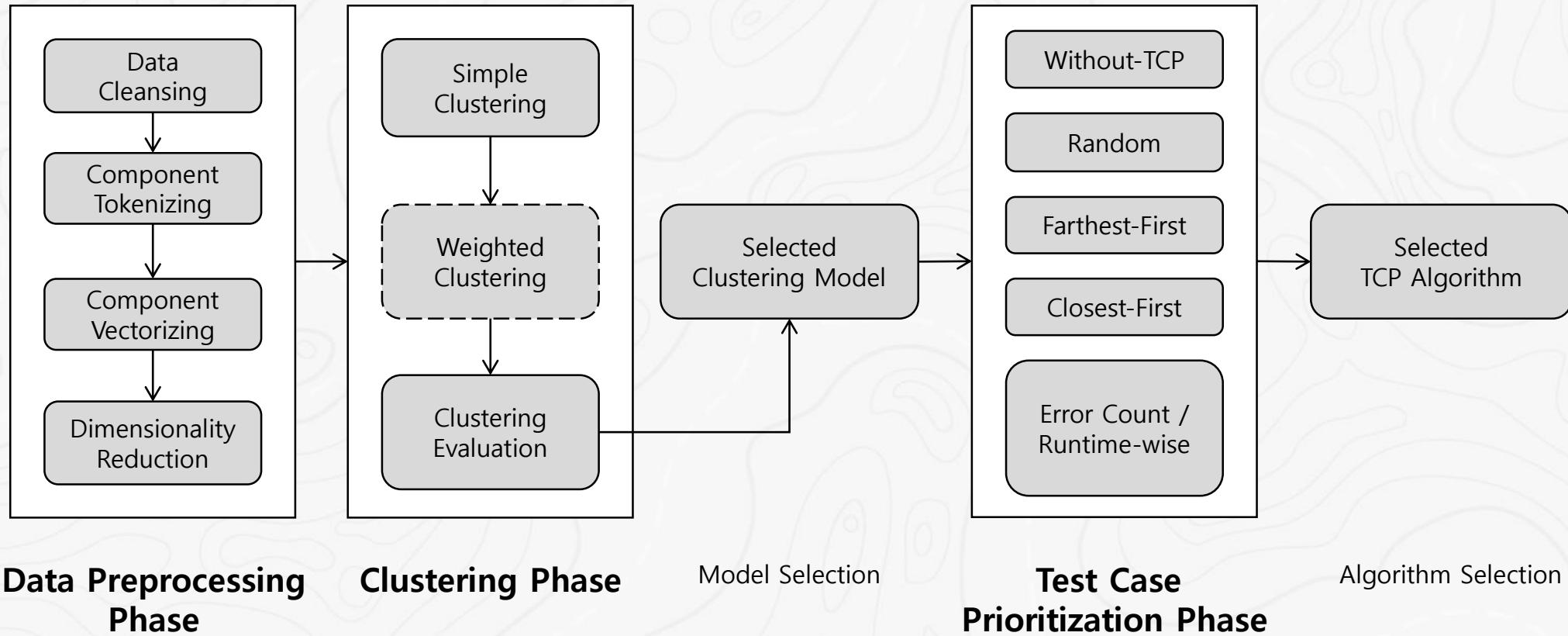


Optimize data preprocessing
and ML processing
by using ***distributed system***.

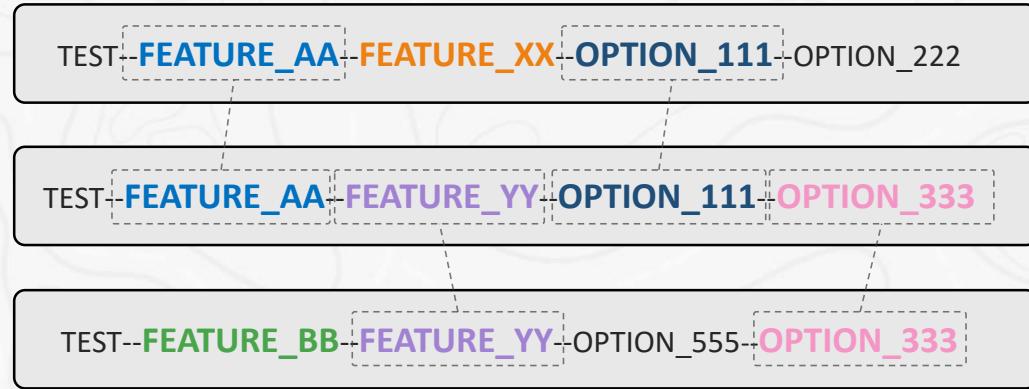


Automatically run low-
prioritized testcases when
resource is idle

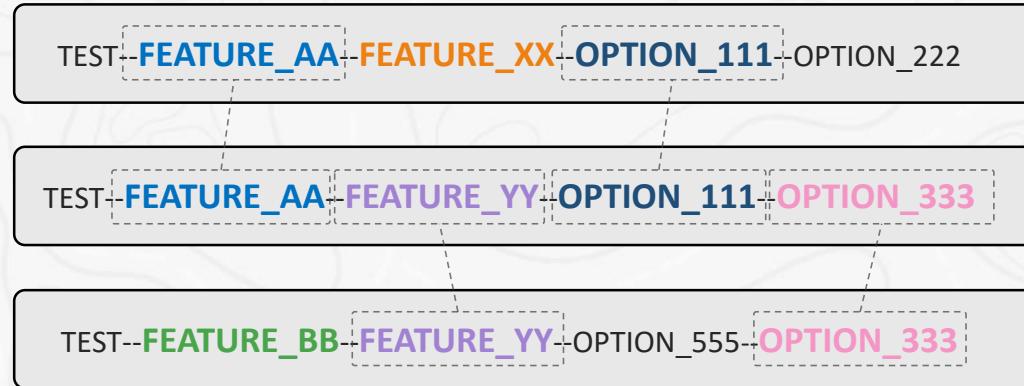
Proposed Algorithm flow



Test name vs. Test log file



Test name vs. Test log file

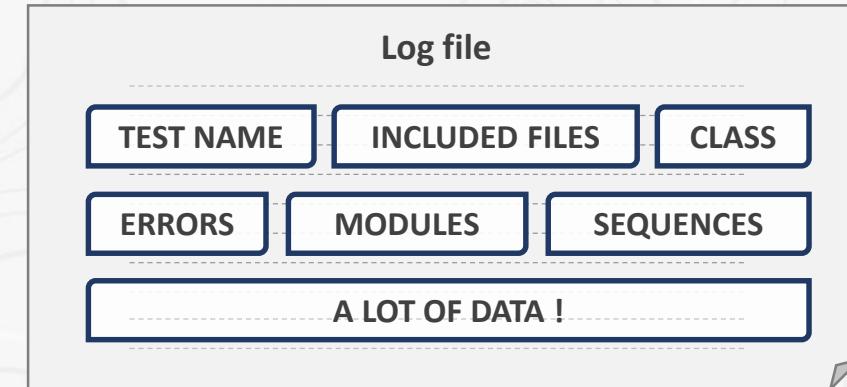
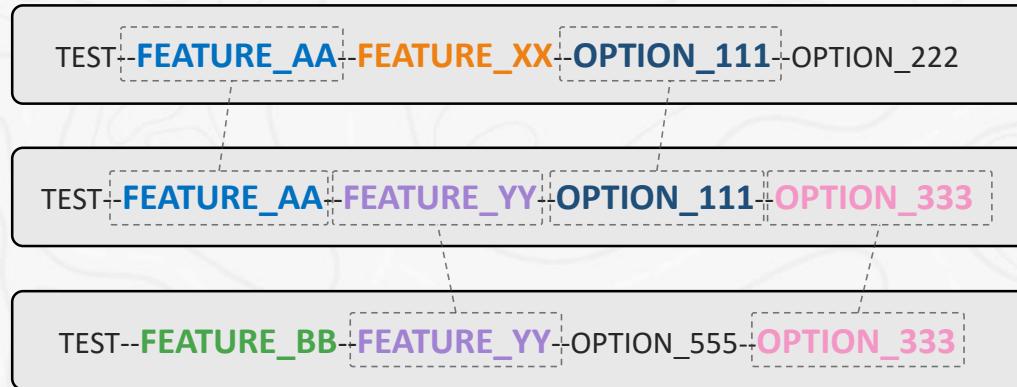


Intuitively grasped similarities

But Lack of information ...

Not always corresponding with actual
behaviors

Test name vs. Test log file

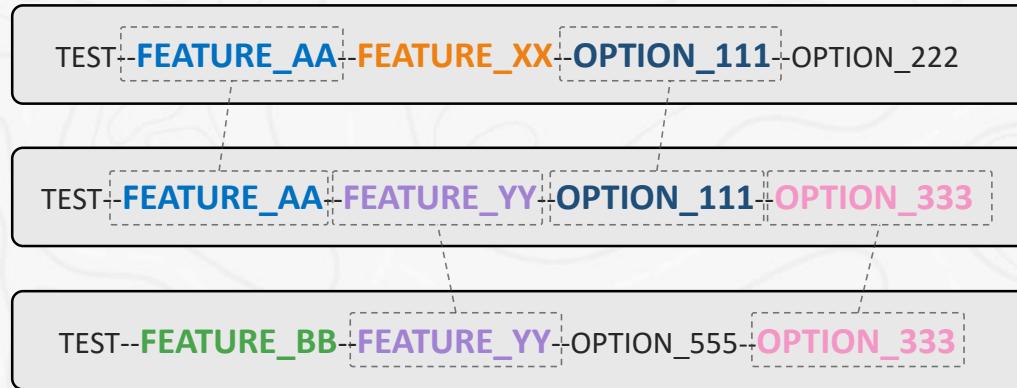


Intuitively grasped similarities

But Lack of information ...

Not always corresponding with actual
behaviors

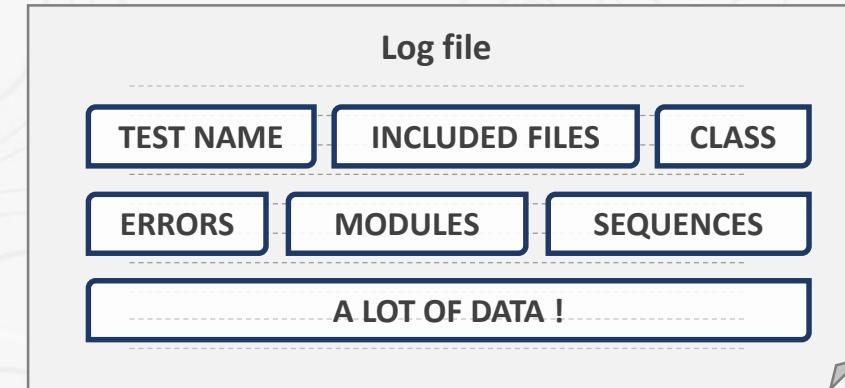
Test name vs. Test log file



Intuitively grasped similarities

But Lack of information ...

Not always corresponding with actual behaviors

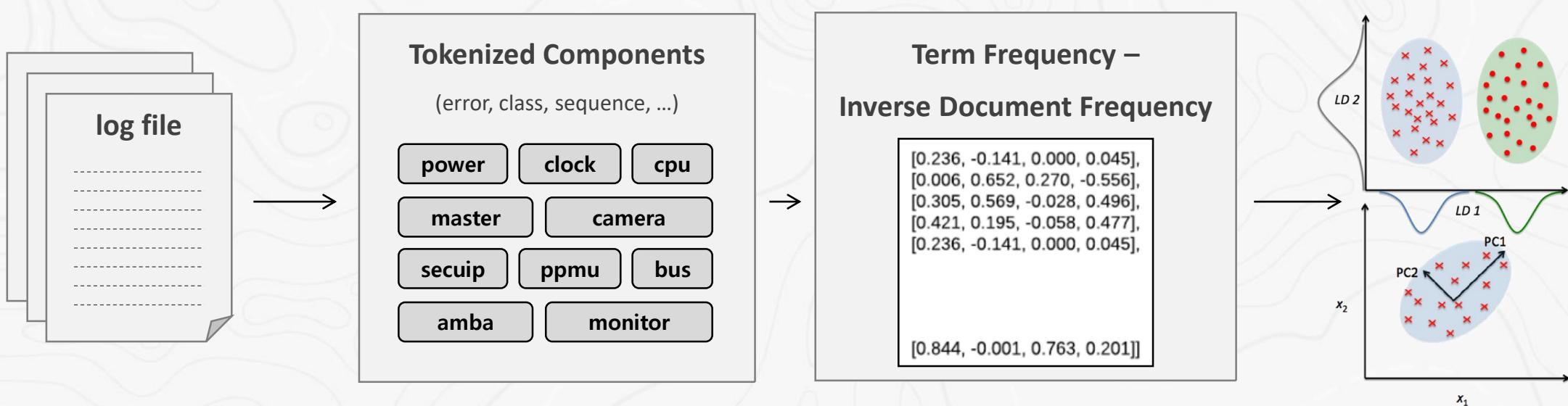


Not only ***Big data*** for test cases

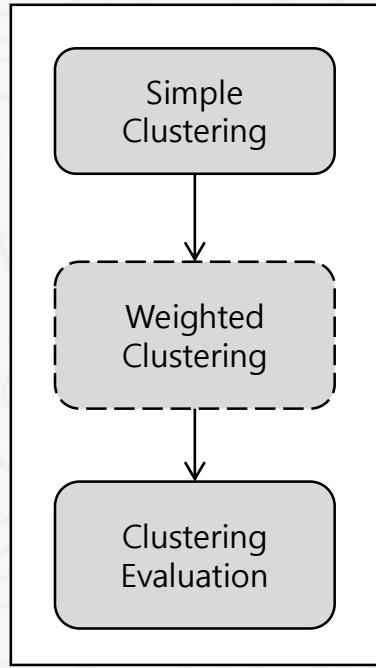
But also Reliable !

Identify unrevealed similarities

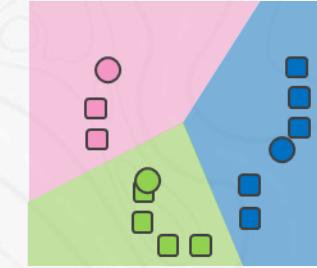
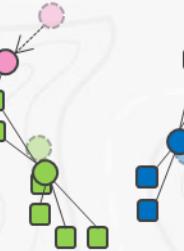
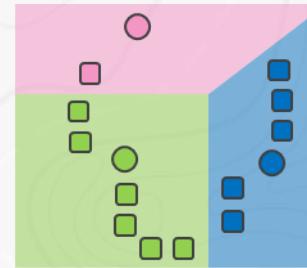
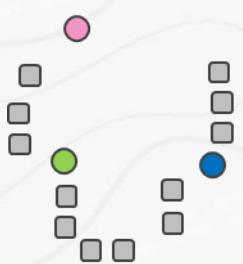
Data preprocessing Phase



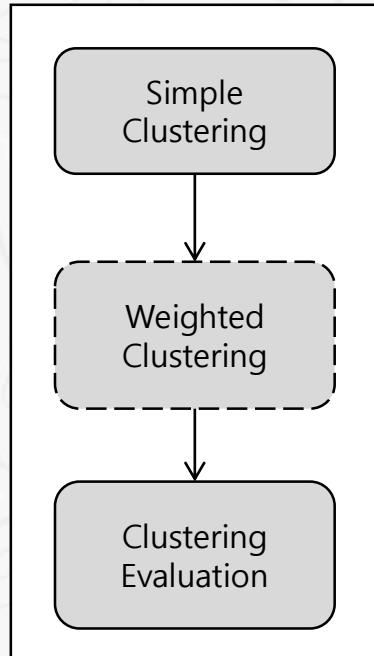
Clustering Phase – KMeans



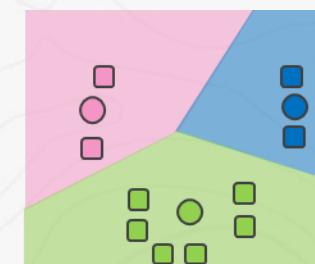
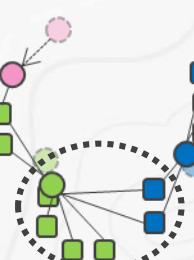
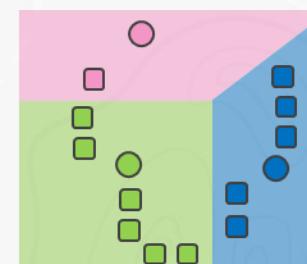
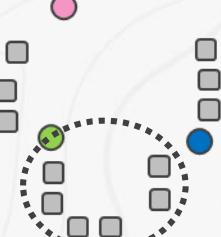
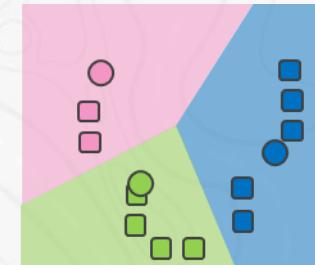
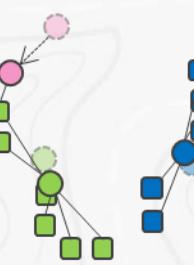
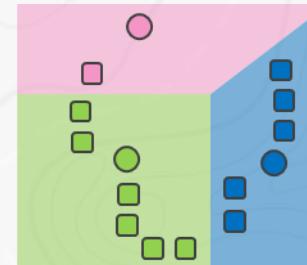
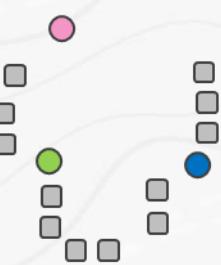
Clustering Phase



Clustering Phase – KMeans



Clustering Phase

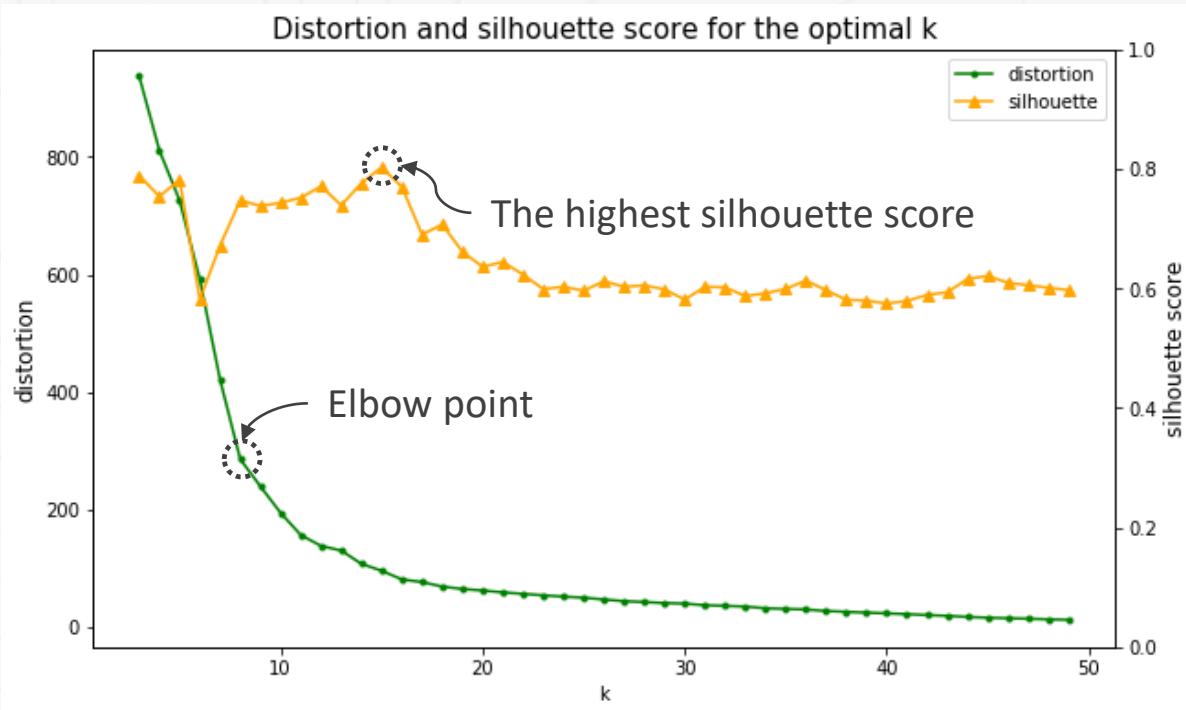


Test Cases with the
important components

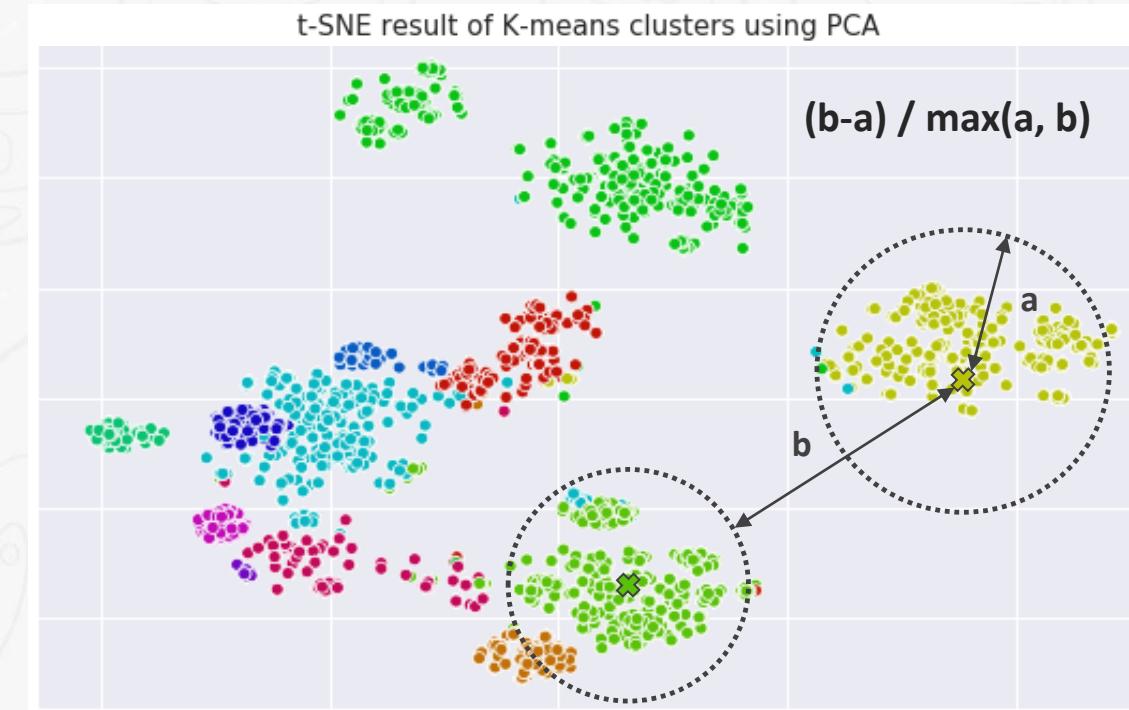
Weighted test cases

Clustering Phase – Measurements

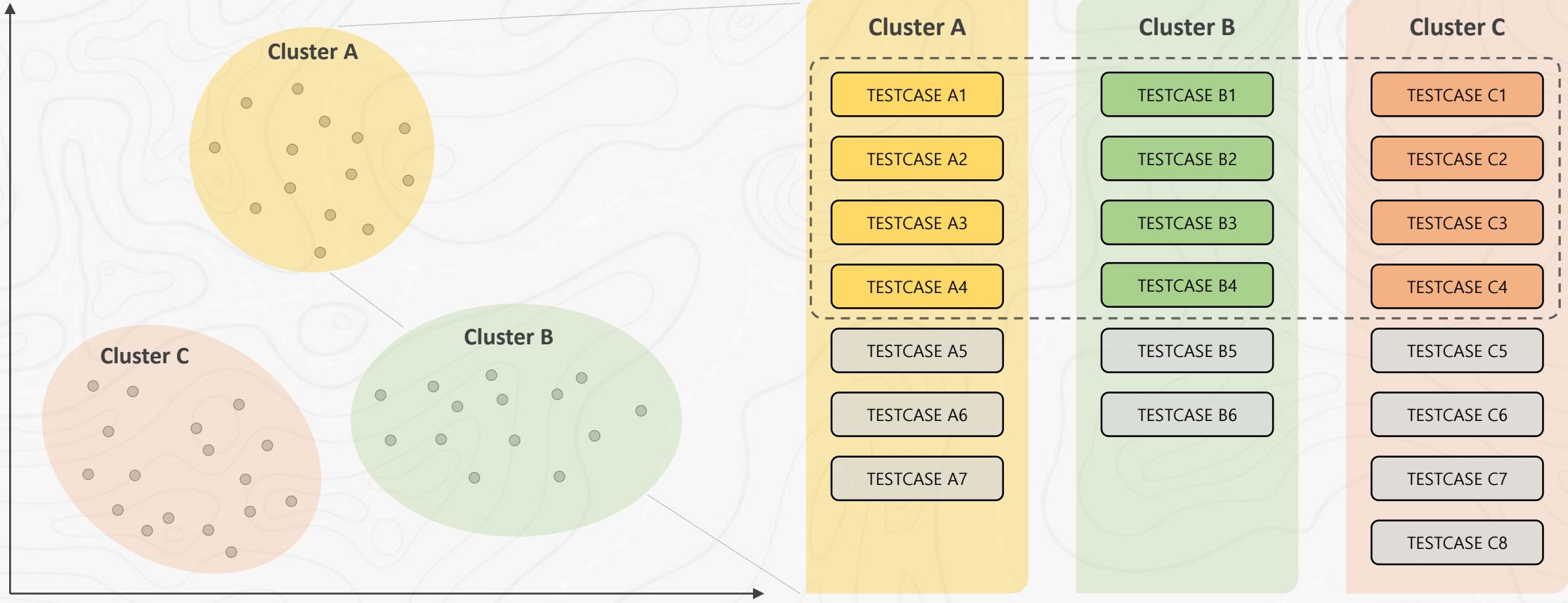
■ Elbow method



■ Silhouette score

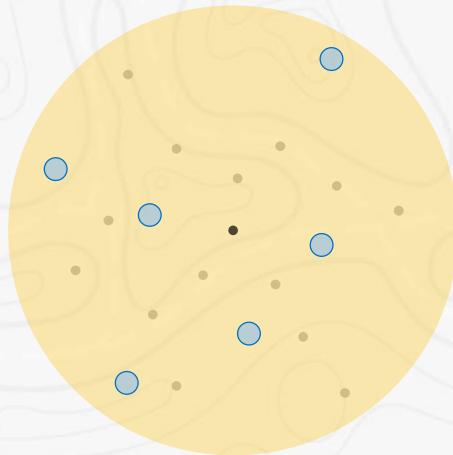


Test Case Prioritization Phase



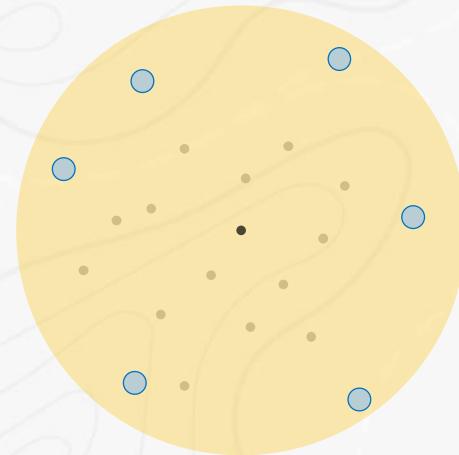
Test Case Prioritization Phase (Cont.)

Random



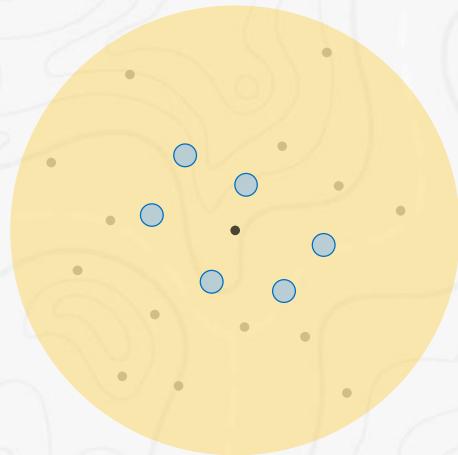
Randomly from clusters

Farthest-First



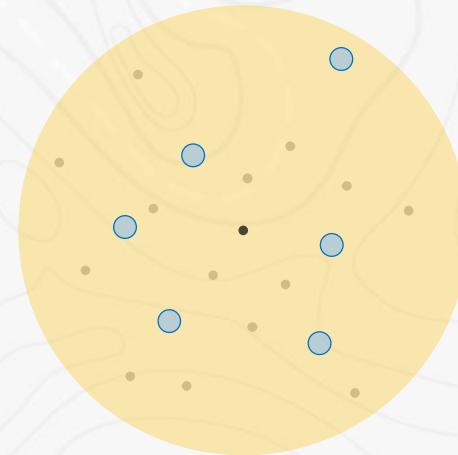
the farthest distance
from the centroid
of each cluster.

Closest-First



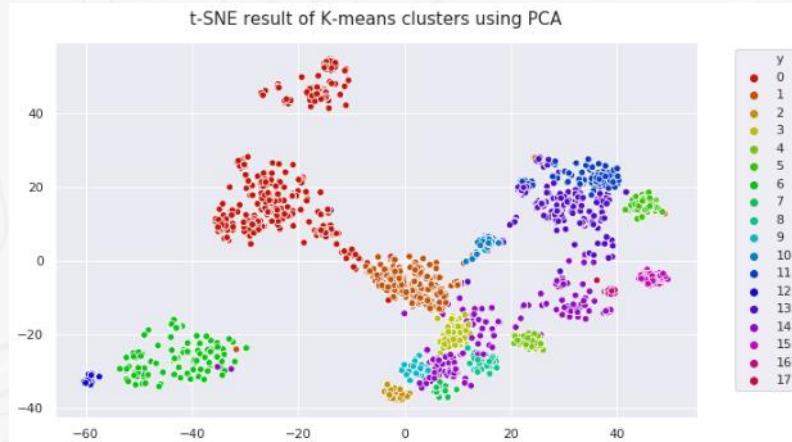
the closest distance
from the centroid
of each cluster.

Error and
runtime-wise



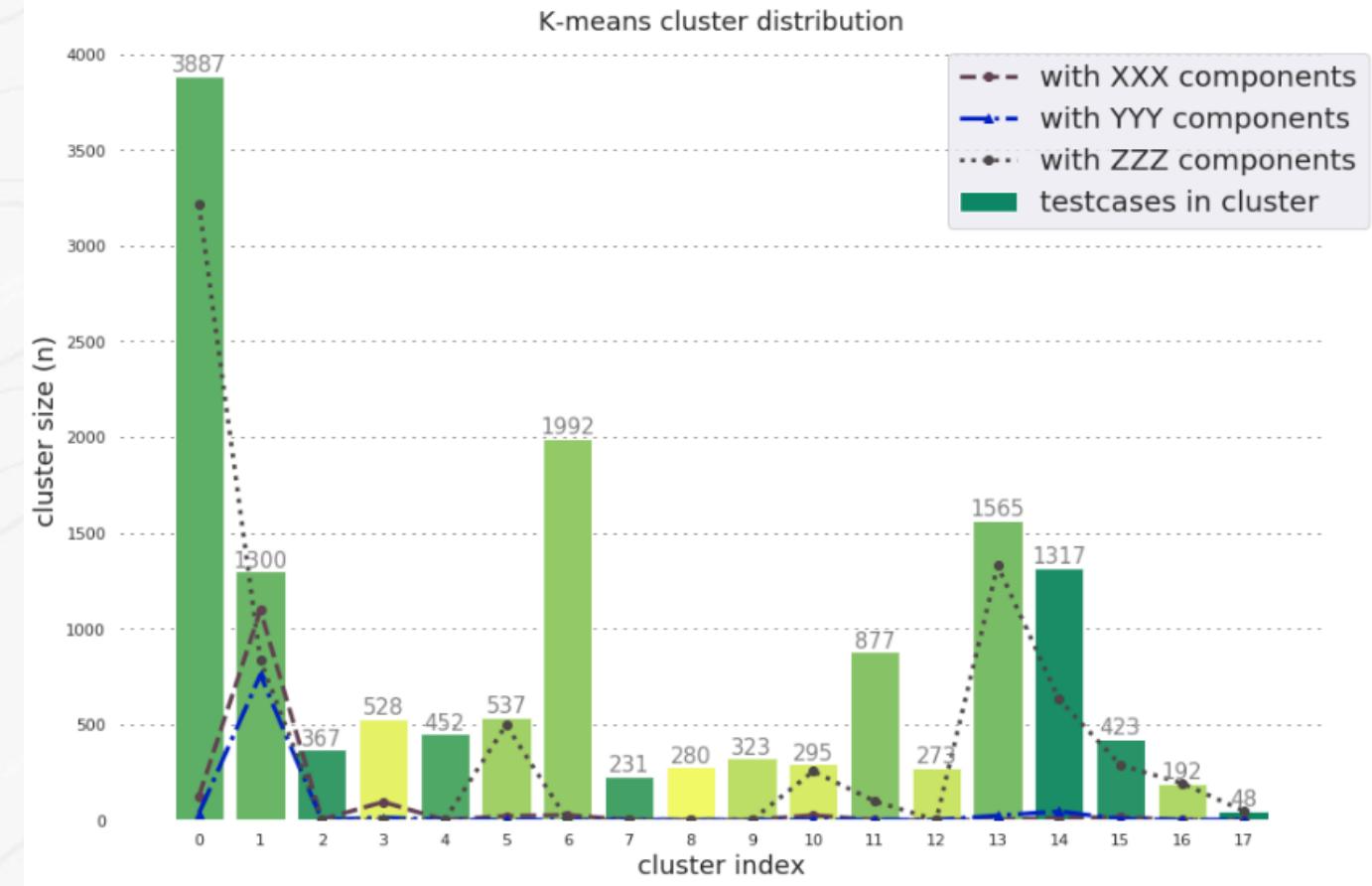
shorter run time and
have more errors
from clusters.

Result – Clustering

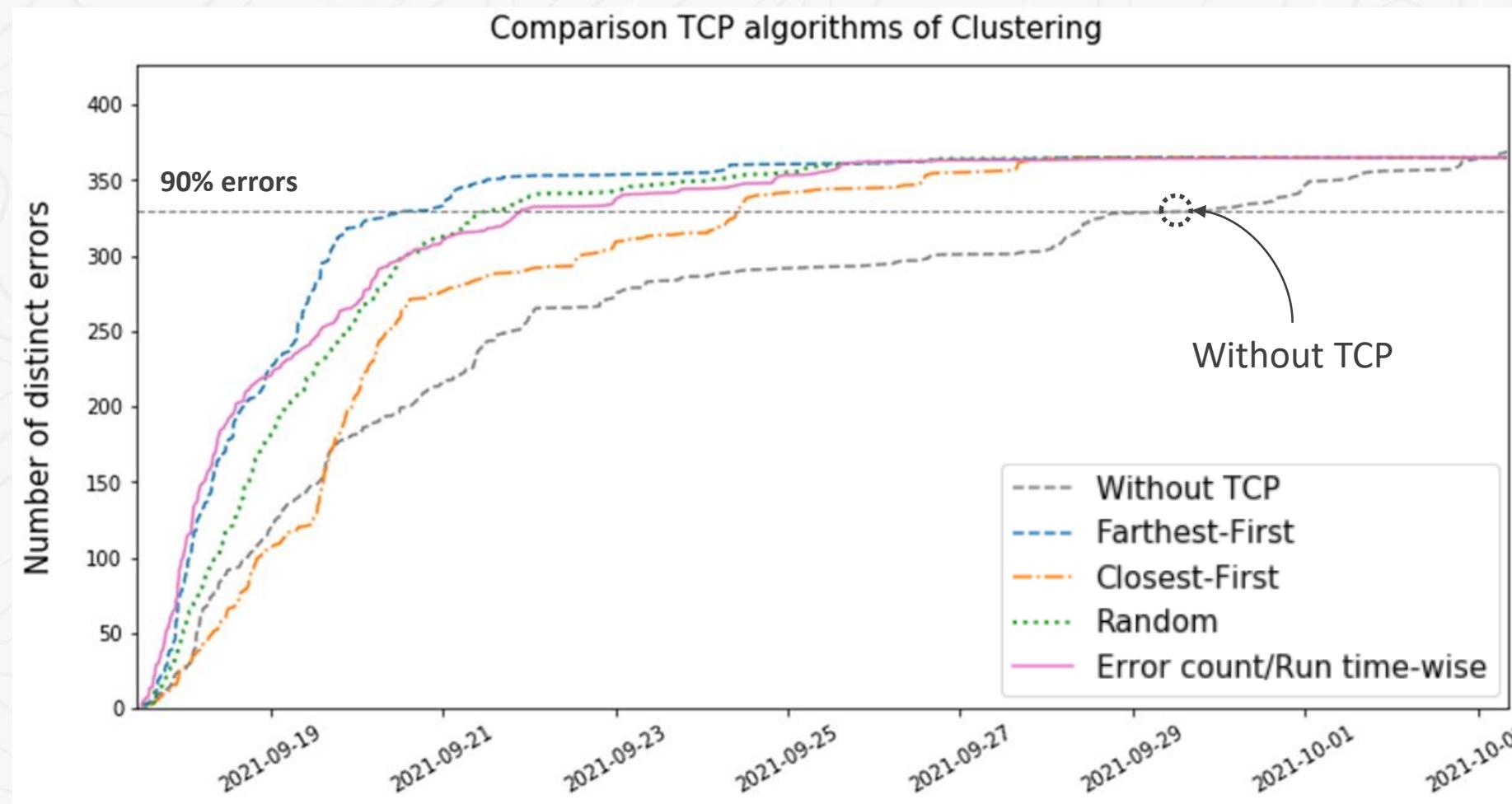


For small set of project

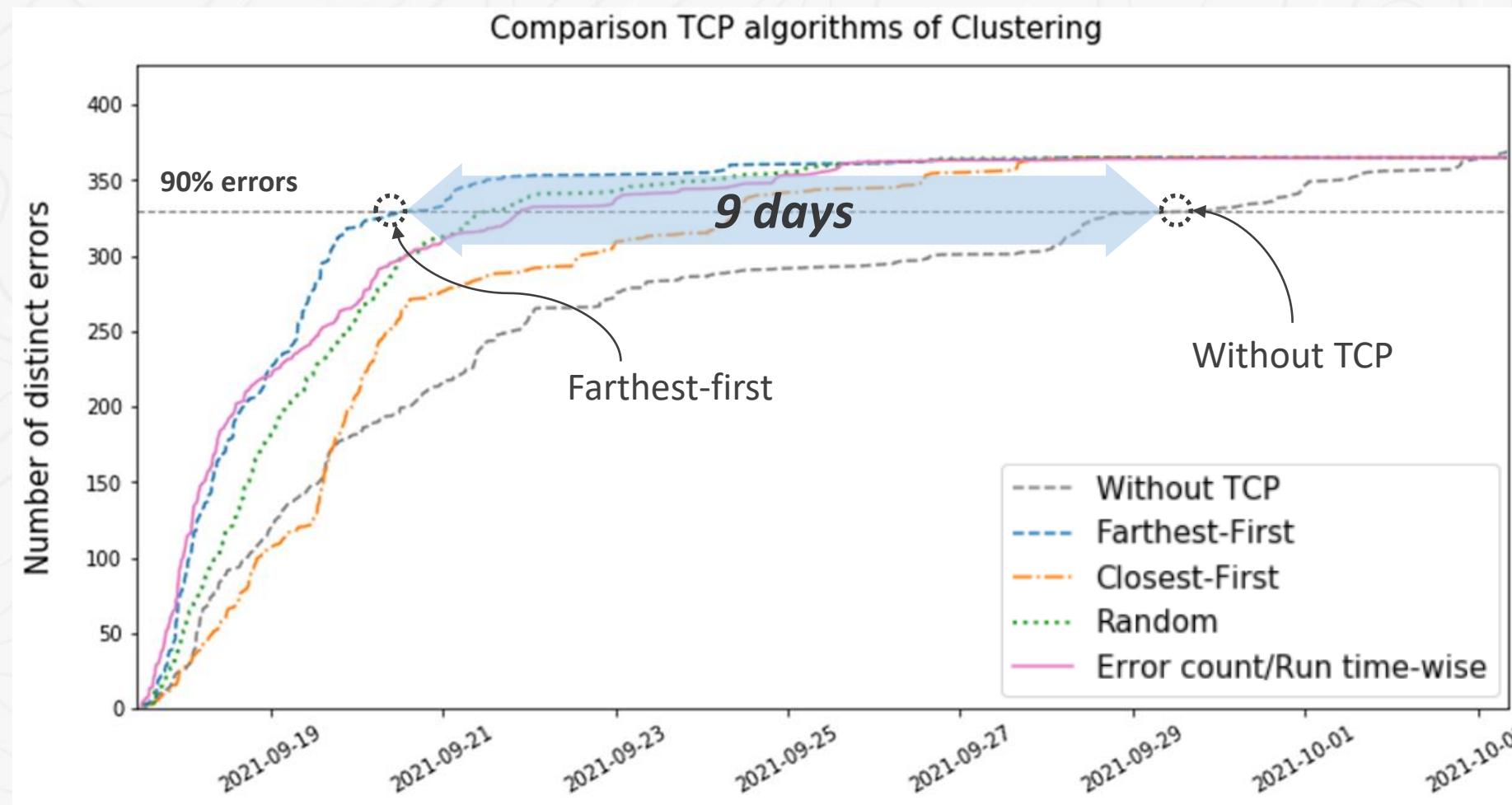
- Number of log files: about 35000
- Number of total components: about 4500
- Optimal K: 18



Result – Test Case Prioritization



Result – Test Case Prioritization



Result – Test Case Prioritization (Cont.)

| Test Case Prioritization method | Time for detecting 90% errors / Total run time (%) | Ratio of test cases for detecting 90% errors (%) |
|---------------------------------|--|--|
| Without TCP | 368.57 hours / 459.52 hours (80.2%) | 100 % |
| Farthest-first | 73.08 hours / 315.37 hours (23.2%) | 76.1 % |
| Closest-first | 166.89 hours / 288.22 hours (57.9%) | 97.7 % |
| Random | 96.7 hours / 325.25 hours (29.7%) | 87.5 % |
| Error-count/Runtime-wise | 106.79 hours / 386.02 hours (27.7%) | 90.6 % |

Conclusion

- Reduced overall verification time and wasted resources
- Efficiently automated design verification flow with machine learning and system

The end

- jjin.choi@samsung.com