2022
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES

# Optimizing Turnaround Times In A CI Flow Using a Scheduler Implementation

Robert Strong

SAMSUNG

accellera
SYSTEMS INITIATIVE

# Contents

- Problem Statement
- Previous Flow
- Scheduler CI Flow Concept
- Implementation
- Learned Best Practices
- Effects & Results
- Future Improvements
- Questions?

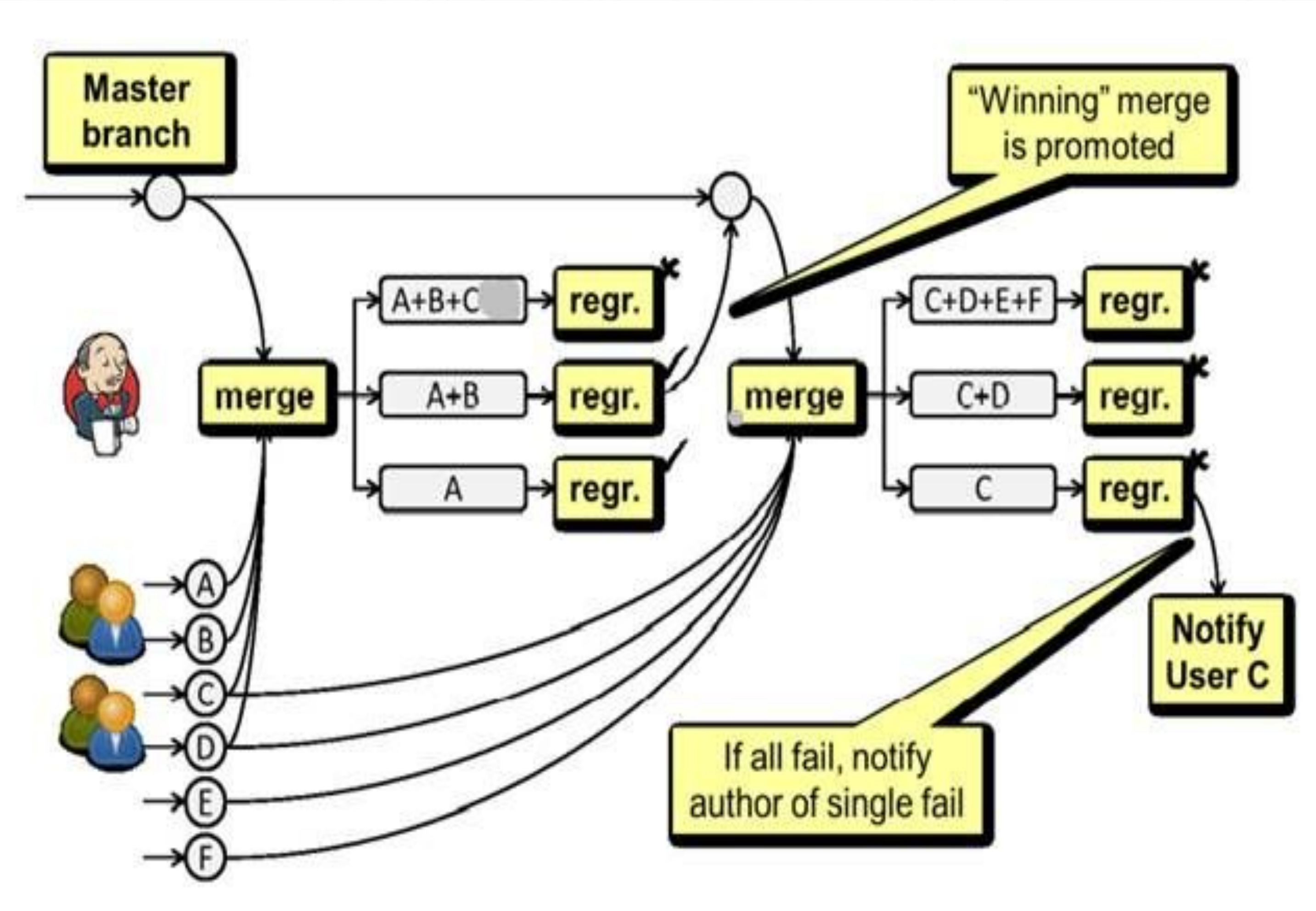
# Problem Statement

- Large number of changesets
- Long turnaround times
  - Accepts & Rejects
- Variable times
- Backlogs during milestones
- Slow in perfect conditions

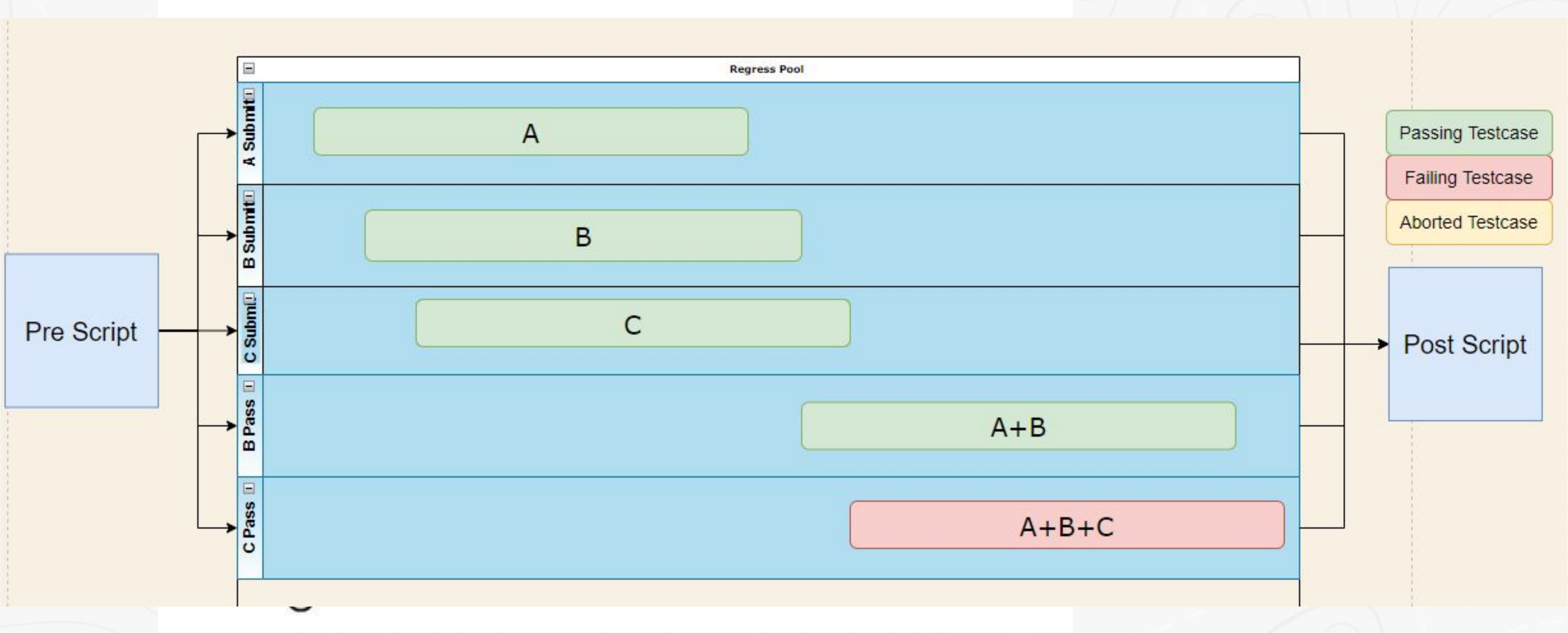
# Previous Flow

- Serial steps
    - Minimize resource usage
- Stage based
    - Large downtimes
    - Human intervention during peaks
- Incremental Improvements
    - Elimination logic
    - More testcases
- Jenkins Jobs

# Previous Scenario #1



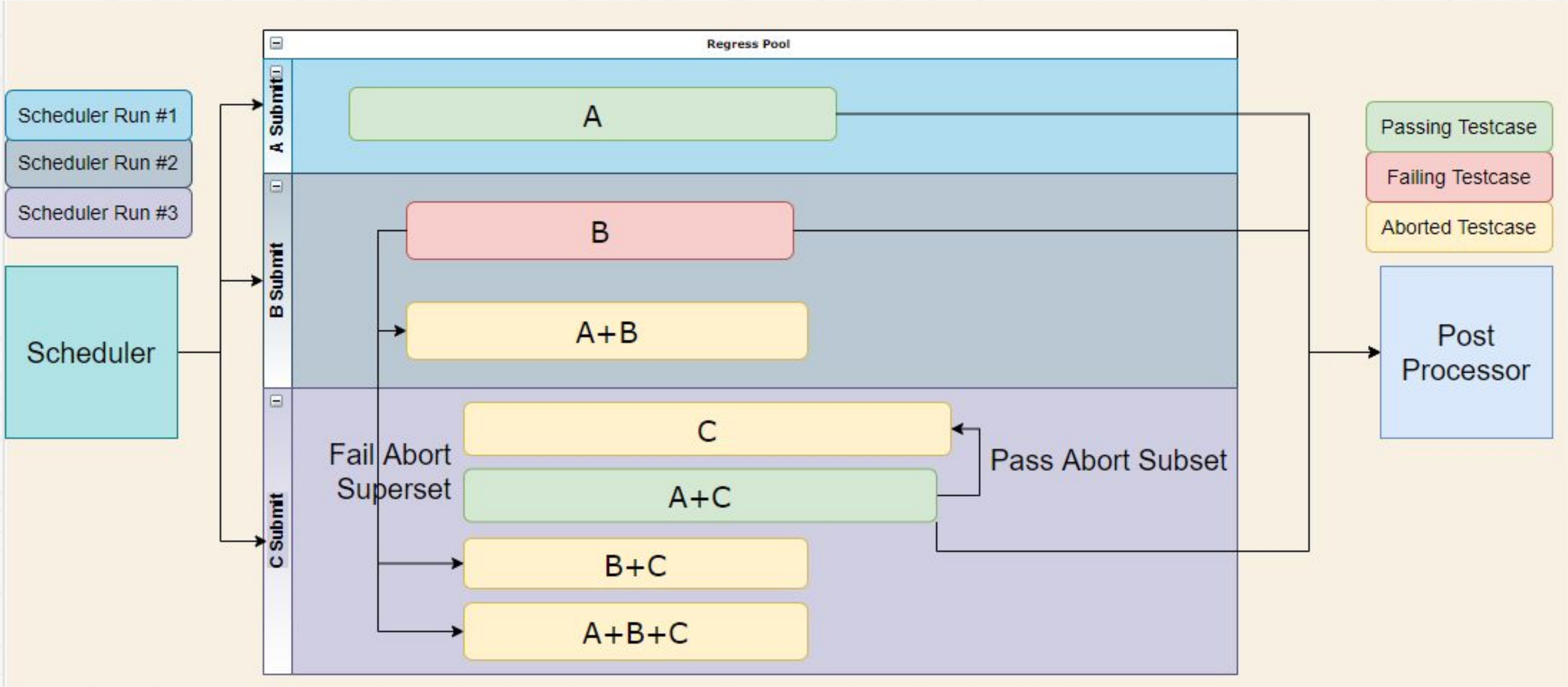
# Previous Scenario #2



# Scheduler CI Flow Concept

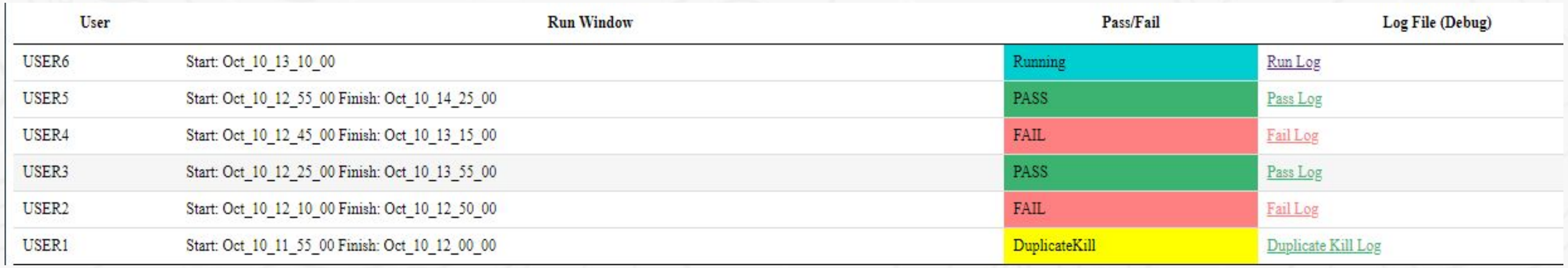
- Run testcases in parallel
  - Testcase – Unique combination of changesets
- Scheduler creates testcases
- Processor reacts to completions
- Goal
  - Turnaround time = Test time

# Scheduler Concept



# Implementation: Individual Scheduler

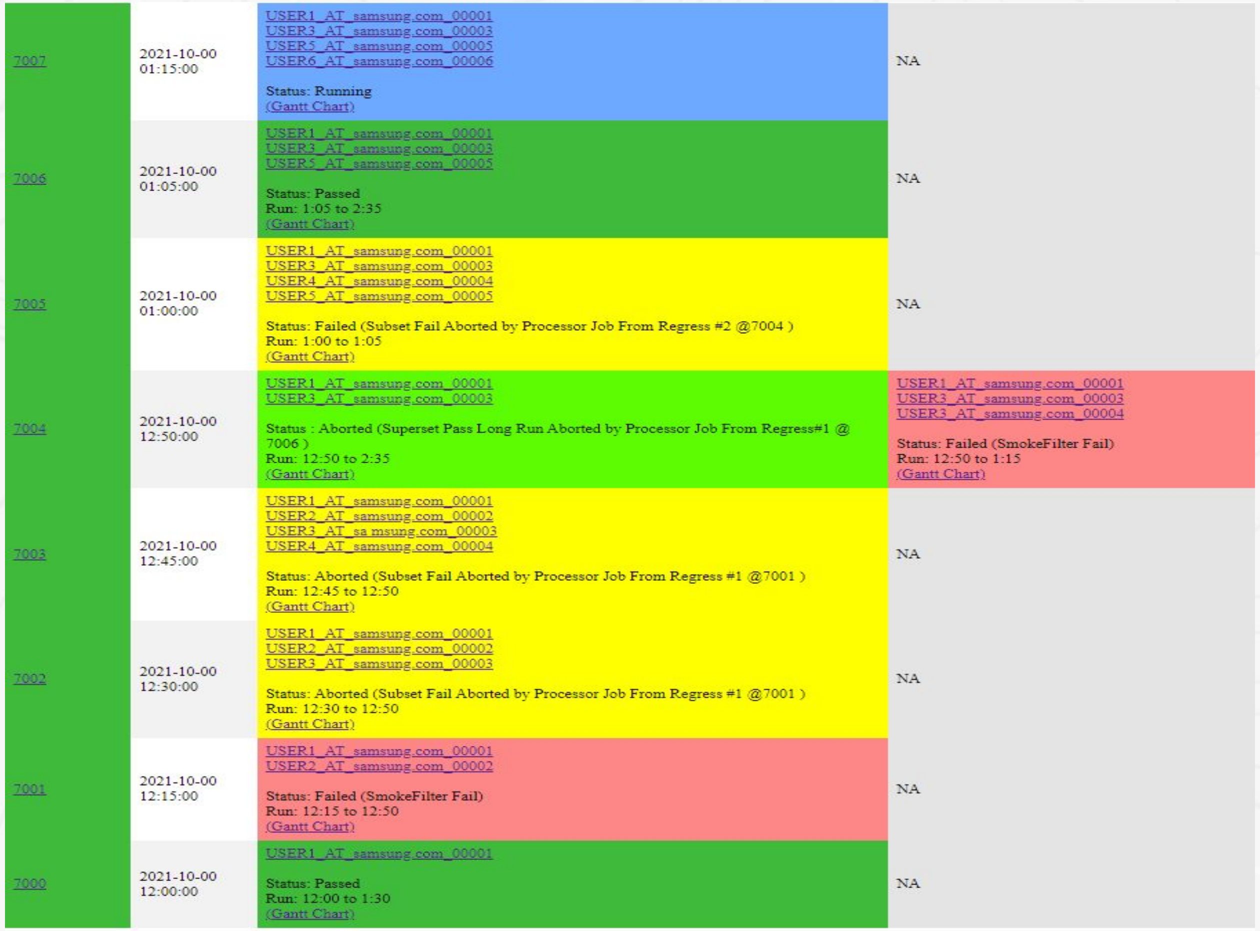
- Individual scheduler grabs user request
  - In house scripts
  - Create immutable copy
  - Launch testcase
  - Ping combined scheduler



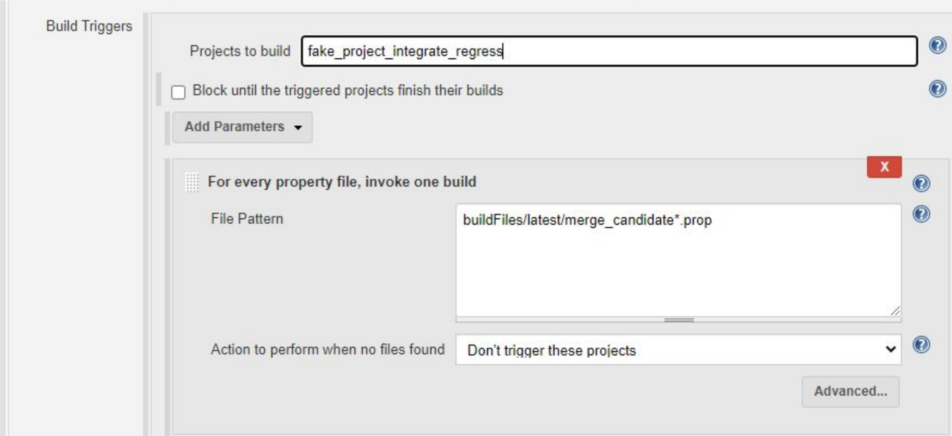
| User | Run Window | Pass/Fail | Log File (Debug) |
|---|---|---|---|
| USER6 | Start: Oct_10_13_10_00 | Running | Run Log |
| USER5 | Start: Oct_10_12_55_00 Finish: Oct_10_14_25_00 | PASS | Pass Log |
| USER4 | Start: Oct_10_12_45_00 Finish: Oct_10_13_15_00 | FAIL | Fail Log |
| USER3 | Start: Oct_10_12_25_00 Finish: Oct_10_13_55_00 | PASS | Pass Log |
| USER2 | Start: Oct_10_12_10_00 Finish: Oct_10_12_50_00 | FAIL | Fail Log |
| USER1 | Start: Oct_10_11_55_00 Finish: Oct_10_12_00_00 | DuplicateKill | Duplicate Kill Log |

# Implementation: Combined Scheduler

- Combined scheduler for testcases > 1
  - Grab state from filesystem
  - Prep workspaces
  - Launch needed testcases
  - Jenkins prop system



| | | | |
|---|---|---|---|
| 7007 | 2021-10-00<br>01:15:00 | USER1_AT_samsung.com_00001<br>USER3_AT_samsung.com_00003<br>USER5_AT_samsung.com_00005<br>USER6_AT_samsung.com_00006<br>Status: Running<br>(Gantt Chart) | NA |
| 7006 | 2021-10-00<br>01:05:00 | USER1_AT_samsung.com_00001<br>USER3_AT_samsung.com_00003<br>USER5_AT_samsung.com_00005<br>Status: Passed<br>Run: 1:05 to 2:35<br>(Gantt Chart) | NA |
| 7005 | 2021-10-00<br>01:00:00 | USER1_AT_samsung.com_00001<br>USER3_AT_samsung.com_00003<br>USER4_AT_samsung.com_00004<br>USER5_AT_samsung.com_00005<br>Status: Failed (Subset Fail Aborted by Processor Job From Regress #2 @7004 )<br>Run: 1:00 to 1:05<br>(Gantt Chart) | NA |
| 7004 | 2021-10-00<br>12:50:00 | USER1_AT_samsung.com_00001<br>USER3_AT_samsung.com_00003<br>Status : Aborted (Superset Pass Long Run Aborted by Processor Job From Regress#1 @ 7006 )<br>Run: 12:50 to 2:35<br>(Gantt Chart) | USER1_AT_samsung.com_00001<br>USER3_AT_samsung.com_00003<br>USER3_AT_samsung.com_00004<br>Status: Failed (SmokeFilter Fail)<br>Run: 12:50 to 1:15<br>(Gantt Chart) |
| 7003 | 2021-10-00<br>12:45:00 | USER1_AT_samsung.com_00001<br>USER2_AT_samsung.com_00002<br>USER3_AT_sa msung.com_00003<br>USER4_AT_samsung.com_00004<br>Status: Aborted (Subset Fail Aborted by Processor Job From Regress #1 @7001 )<br>Run: 12:45 to 12:50<br>(Gantt Chart) | NA |
| 7002 | 2021-10-00<br>12:30:00 | USER1_AT_samsung.com_00001<br>USER2_AT_samsung.com_00002<br>USER3_AT_samsung.com_00003<br>Status: Aborted (Subset Fail Aborted by Processor Job From Regress #1 @7001 )<br>Run: 12:30 to 12:50<br>(Gantt Chart) | NA |
| 7001 | 2021-10-00<br>12:15:00 | USER1_AT_samsung.com_00001<br>USER2_AT_samsung.com_00002<br>Status: Failed (SmokeFilter Fail)<br>Run: 12:15 to 12:50<br>(Gantt Chart) | NA |
| 7000 | 2021-10-00<br>12:00:00 | USER1_AT_samsung.com_00001<br>Status: Passed<br>Run: 12:00 to 1:30<br>(Gantt Chart) | NA |



Build Triggers

Projects to build: fake_project_integrate_regress

☐ Block until the triggered projects finish their builds

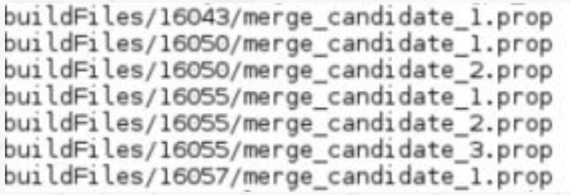
For every property file, invoke one build

File Pattern: buildFiles/latest/merge_candidate*.prop

Action to perform when no files found: Don't trigger these projects

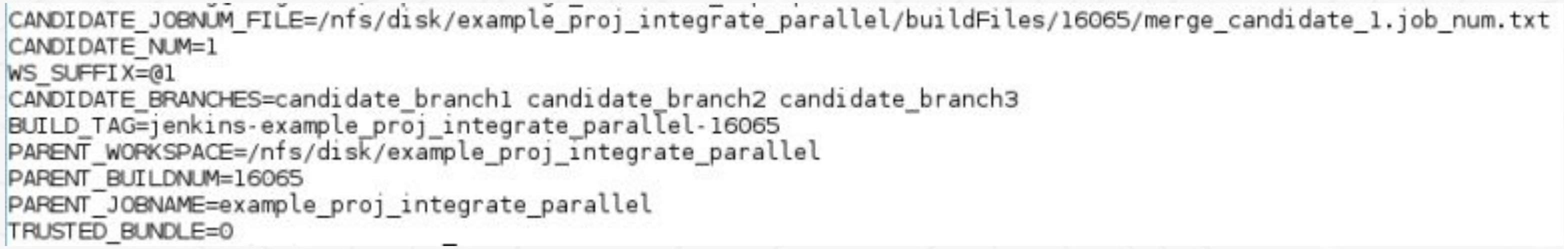
# Implementation: Testcase State & Prop Details

- State saved in filesystem
  - Directory for active testcases
  - Multiple testcases launched/scheduler run
  - Moved to archive after completion
  - DB for pass/fail & metadata



```
buildFiles/16043/merge_candidate_1.prop
buildFiles/16050/merge_candidate_1.prop
buildFiles/16050/merge_candidate_2.prop
buildFiles/16055/merge_candidate_1.prop
buildFiles/16055/merge_candidate_2.prop
buildFiles/16055/merge_candidate_3.prop
buildFiles/16057/merge_candidate_1.prop
```

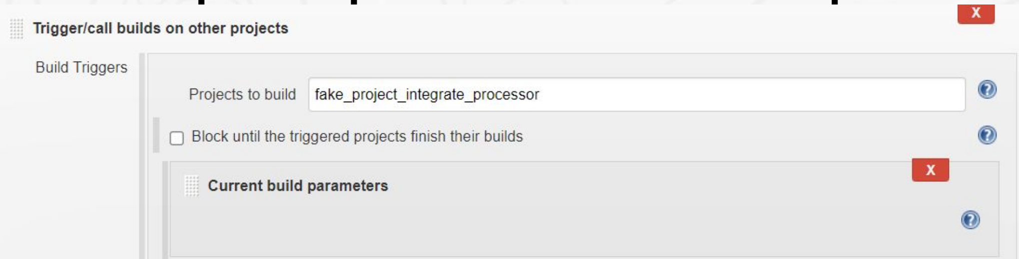
- Prop files w/all info per testcase
  -



```
CANDIDATE_JOBNUM_FILE=/nfs/disk/example_proj_integrate_parallel/buildFiles/16065/merge_candidate_1.job_num.txt
CANDIDATE_NUM=1
WS_SUFFIX=@1
CANDIDATE_BRANCHES=candidate_branch1 candidate_branch2 candidate_branch3
BUILD_TAG=jenkins-example_proj_integrate_parallel-16065
PARENT_WORKSPACE=/nfs/disk/example_proj_integrate_parallel
PARENT_BUILDNUM=16065
PARENT_JOBNAME=example_proj_integrate_parallel
TRUSTED_BUNDLE=0
```

# Implementation Details: Testcases

- Freestyle Jenkins
  - Run test suite
- Serial post-process for complete



Trigger/call builds on other projects

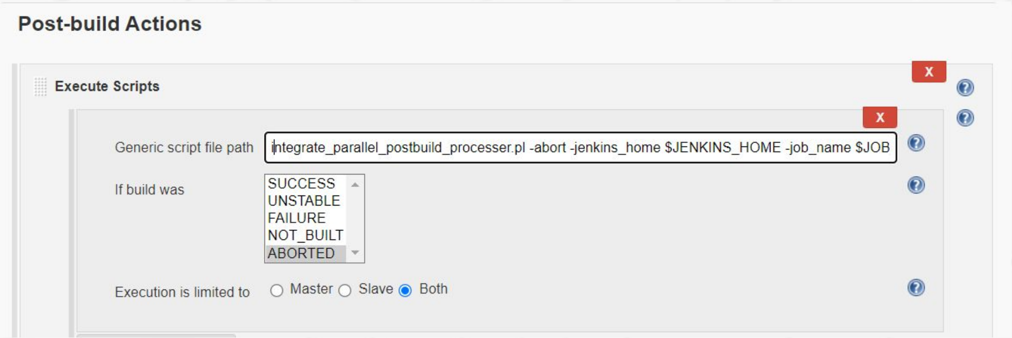
Build Triggers

Projects to build: fake_project_integrate_processor

□ Block until the triggered projects finish their builds

Current build parameters

- Non-serial for aborted



Post-build Actions

Execute Scripts

Generic script file path: integrate_parallel_postbuild_processer.pl -abort -jenkins_home $JENKINS_HOME -job_name $JOB

If build was:
SUCCESS
UNSTABLE
FAILURE
NOT_BUILT
ABORTED

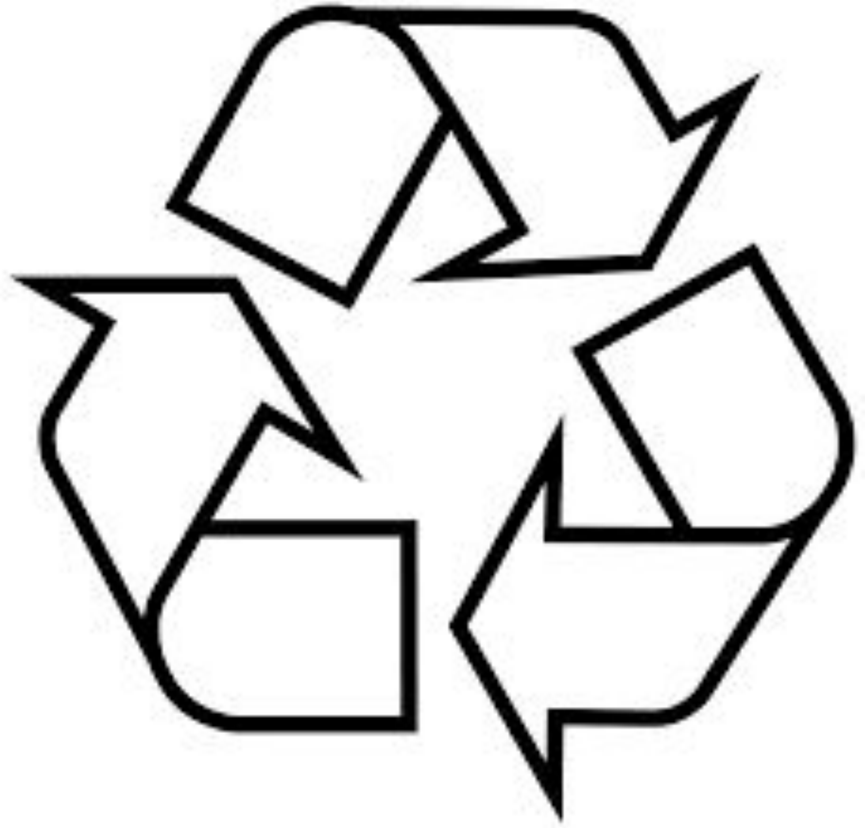
Execution is limited to ○ Master ○ Slave ● Both

# Implementation: Post-processer

- Completed mode only:
  - Changesets to push/remove
  - Kill subset & superset testcases
- Completed & Aborted mode
  - Update filesystem state
  - Recycle Workspace
  - Notify Users

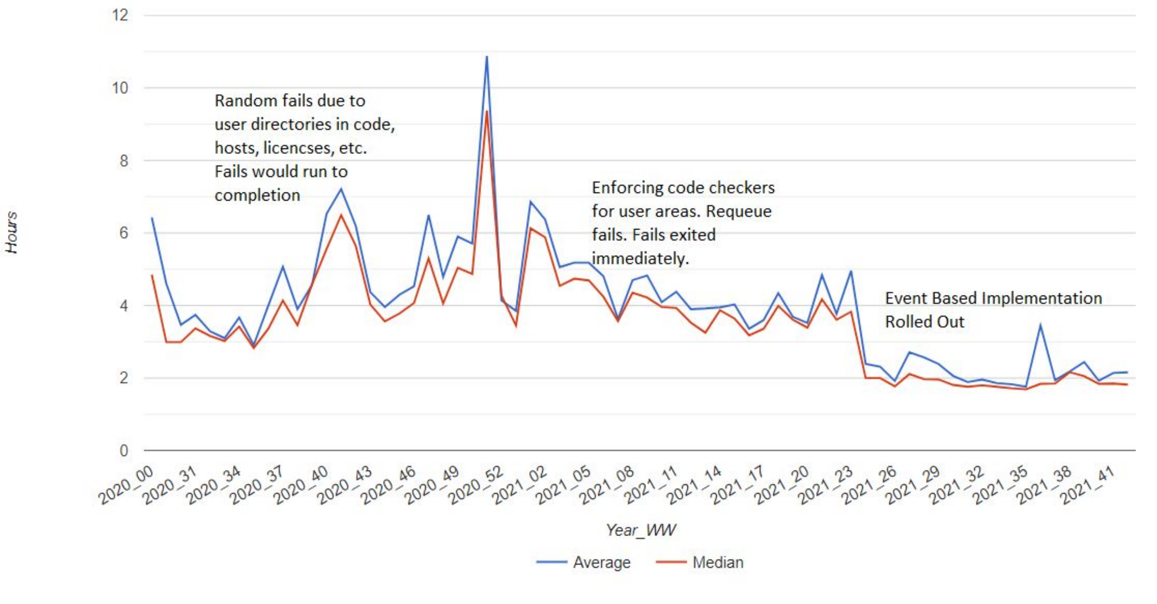
# Best Practices

- Lower costs allows for more testcases
  - Finite resources (batch/license/IO)
- Minimizing the cost of a testcase
  - Exit immediately on fails
  - Block external dependencies
  - Minimize false fails
  - Recycle workspaces
- “Guilt Free” launch & kill testcase



# Effects & Results

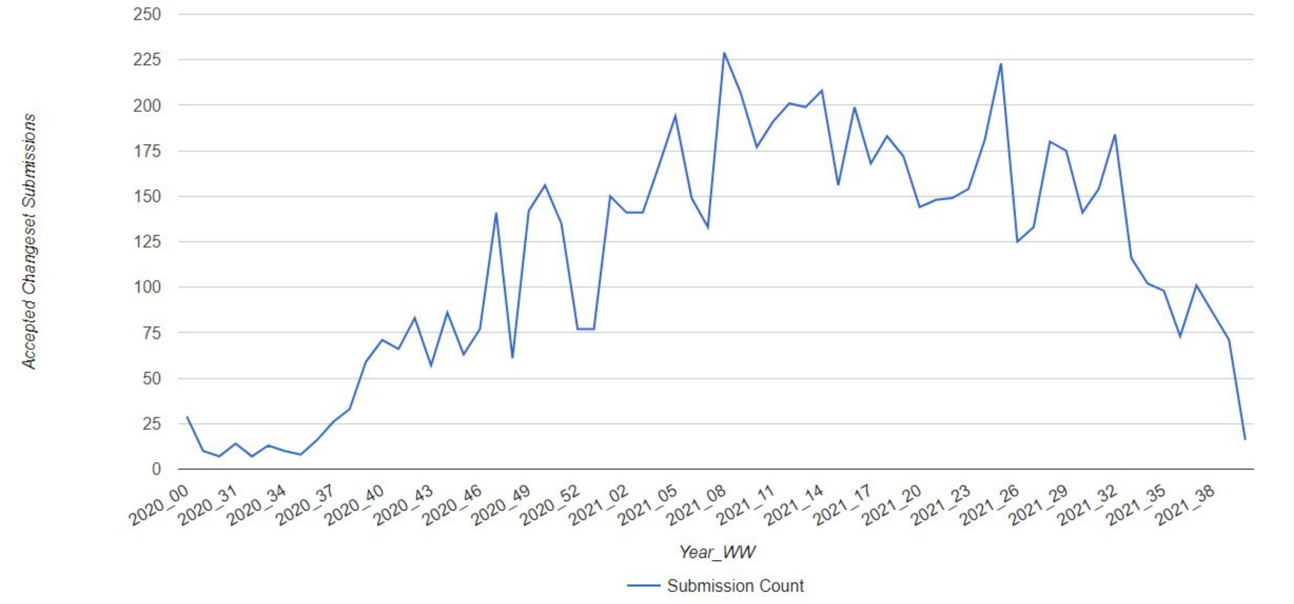
- Fast & stable average & median turnaround times
  - ~15 min of overhead
  - ~50% reduction in average & median times
- Consistent results @ peak submission rate



Random fails due to user directories in code, hosts, licencses, etc. Fails would run to completion

Enforcing code checkers for user areas. Requeue fails. Fails exited immediately.

Event Based Implementation Rolled Out



# Future Improvements

- Batch system interface
- Consolidate schedulers
- Dynamic scheduler
  - Adjust number of testcases
    - Resources
    - # of changesets
    - Pass rate
    - Priority
- Partial Testing
- Data Queries/Storage

# Questions?

- Issues?
- Reliability?
- Admin Interface?