

**git\_comments:**

1. `**` Instantiates an {@code ErrorInfo} to cover inconsistent behavior due to FLINK-21376. `**` @param exception The error cause that might be {@code null}. `*` @param timestamp The timestamp the error was noticed. `*` @return a {@code ErrorInfo} containing a generic {@link FlinkException} in case of a missing `*` error cause.
2. once an ErrorInfo is set, never changes
3. `*` True if the original failure was a global failure.
4. `**` Returns an {@code Optional} with the {@link ExecutionVertexID} of the task causing this `*` failure or an empty {@code Optional} if it's a global failure. `**` @return The {@code ExecutionVertexID} of the causing task or an empty {@code Optional} if `*` it's a global failure.
5. `*` Failure reason. {@code @Nullable} because of FLINK-21376.
6. `**` Task vertices to restart to recover from the failure or {@code null} if the failure is not `*` recoverable.
7. `**` The {@link ExecutionVertexID} referring to the {@link ExecutionVertex} the failure is `*` originating from or {@code null} if it's a global failure.
8. fallback in case of a global fail over - no failed state is set and, therefore, no timestamp was taken
9. we have to cancel the task and trigger the restart to have the exception history populated
10. initiate restartable failure
11. assert restarted attempt
12. assert job failure attempt
13. initiate job failure

**git\_commits:**

1. **summary:** [FLINK-21187][runtime] Adds exception history  
**message:** [FLINK-21187][runtime] Adds exception history The initial implementation only collects the actual exception without considering any other failures that might happen concurrently. This closes #14798.

**github\_issues:****github\_issues\_comments:****github\_pulls:**

1. **title:** [FLINK-21187] Provide exception history for root causes  
**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log `*` updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. `*` FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. `*` Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change `*` `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. `*` `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable  
**label:** code-design
2. **title:** [FLINK-21187] Provide exception history for root causes  
**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log `*` updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. `*` FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. `*` Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change `*` `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. `*` `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable
3. **title:** [FLINK-21187] Provide exception history for root causes  
**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042]

(<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

4. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

5. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

6. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

7. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing,

Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

8. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

9. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

10. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

11. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

12. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the

exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

13. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

14. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

15. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

16. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

17. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042]

(<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

18. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

19. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

20. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

21. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no -

Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

22. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

23. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

24. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

25. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

26. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated

accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

27. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

28. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

29. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

30. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

31. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042](https://issues.apache.org/jira/browse/FLINK-6042)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

32. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042](https://issues.apache.org/jira/browse/FLINK-6042)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

33. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042](https://issues.apache.org/jira/browse/FLINK-6042)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

34. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042](https://issues.apache.org/jira/browse/FLINK-6042)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

35. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042](https://issues.apache.org/jira/browse/FLINK-6042)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the



exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

36. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

37. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

38. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

39. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

40. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042]

(<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

41. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

42. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

43. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

44. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing,

Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

45. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

46. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

47. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

48. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

49. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the

exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

50. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

51. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

52. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

53. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* `FailureHandlingResult` get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

54. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface

to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

55. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** documentation

56. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

57. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

58. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing,

Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** test

59. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

60. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

61. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

62. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

63. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure

root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

64. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

65. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

66. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

67. **title:** [FLINK-21187] Provide exception history for root causes

**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable

**label:** code-design

68. **title:** [FLINK-21187] Provide exception history for root causes  
**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable
69. **title:** [FLINK-21187] Provide exception history for root causes  
**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable
70. **title:** [FLINK-21187] Provide exception history for root causes  
**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable
71. **title:** [FLINK-21187] Provide exception history for root causes  
**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? no - If yes, how is the feature documented? not applicable
72. **title:** [FLINK-21187] Provide exception history for root causes  
**body:** ## What is the purpose of the change This PR handles the initial exception history proposal collecting only the root causes. This is part of a major effort of providing a exception history to the Flink web UI ([FLINK-6042] (<https://issues.apache.org/jira/browse/FLINK-6042>)). ## Brief change log \* updated `AccessExecutionGraph` interface to return `ErrorInfo` instead of a stringified version of the failure. \* FailureHandlingResult get a new member identifying the `ExecutionVertex` that failed. \* Added two maps collecting the error infos for each occurring failure root cause. ## Verifying this change \* `ExecutionFailureHandlerTest` and `FailureHandlingResultTest` were updated accordingly. \* `DefaultSchedulerTest#testExceptionHistoryWithRestartableFailure` was introduced to cover the exception history collection. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): no - The public API, i.e., is any changed class annotated with



`@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no -  
 Anything that affects deployment or recovery: JobManager (and its components), Checkpointing,  
 Kubernetes/Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request  
 introduce a new feature? no - If yes, how is the feature documented? not applicable

### github\_pulls\_comments:

1. Thanks a lot for your contribution to the Apache Flink project. I'm the @flinkbot. I help the community to review your pull request. We will use this comment to track the progress of the review. ## Automated Checks Last check on commit e962af6d8df77a725defeb92537837e6f685292f (Fri Feb 19 07:34:36 UTC 2021) \*\*Warnings:\*\* \* No documentation files were touched! Remember to keep the Flink docs up to date! <sub>Mention the bot in a comment to re-run the automated checks.</sub> ## Review Progress \* â “ 1. The [description] looks good. \* â “ 2. There is [consensus] that the contribution should go into to Flink. \* â “ 3. Needs [attention] from. \* â “ 4. The change fits into the overall [architecture]. \* â “ 5. Overall code [quality] is good. Please see the [Pull Request Review Guide] (<https://flink.apache.org/contributing/reviewing-prs.html>) for a full explanation of the review process.<details> The Bot is tracking the review progress through labels. Labels are applied according to the order of the review items. For consensus, approval by a Flink committer or PMC member is required <summary>Bot commands</summary> The @flinkbot bot supports the following commands: - `@flinkbot approve description` to approve one or more aspects (aspects: `description`, `consensus`, `architecture` and `quality`) - `@flinkbot approve all` to approve all aspects - `@flinkbot approve-until architecture` to approve everything until `architecture` - `@flinkbot attention @username1 [@username2 ..]` to require somebody's attention - `@flinkbot disapprove architecture` to remove an approval you gave earlier </details>
2. <!-- Meta data { "version" : 1, "metaDataEntries" : [ { "hash" : "9a9204577129b7145846210351ee8754928287a3", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=12620", "triggerID" : "9a9204577129b7145846210351ee8754928287a3", "triggerType" : "PUSH" }, { "hash" : "22d7f6d27419e0f58b738b10e3a05a64ef2f24dd", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=12621", "triggerID" : "22d7f6d27419e0f58b738b10e3a05a64ef2f24dd", "triggerType" : "PUSH" }, { "hash" : "b8f6541c7820d1eea9c29c915690070cfebe837b", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=12644", "triggerID" : "b8f6541c7820d1eea9c29c915690070cfebe837b", "triggerType" : "PUSH" }, { "hash" : "f6278de3bae76474c390b6251551e68862e3841f", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=12710", "triggerID" : "f6278de3bae76474c390b6251551e68862e3841f", "triggerType" : "PUSH" }, { "hash" : "0cde4fdbc8d036cdb44884a0974fbf1cf31966c5", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=12912", "triggerID" : "0cde4fdbc8d036cdb44884a0974fbf1cf31966c5", "triggerType" : "PUSH" }, { "hash" : "211a68e9532b0bf10581388b395e474123ec6ad5", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=12944", "triggerID" : "211a68e9532b0bf10581388b395e474123ec6ad5", "triggerType" : "PUSH" }, { "hash" : "ab763747a3acaff1d72dfe9cd9b9dbb3ed0896d70", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=12951", "triggerID" : "ab763747a3acaff1d72dfe9cd9b9dbb3ed0896d70", "triggerType" : "PUSH" }, { "hash" : "5910b1a876d28886a8b5f87c09e67d75d2a45cd3", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=12962", "triggerID" : "5910b1a876d28886a8b5f87c09e67d75d2a45cd3", "triggerType" : "PUSH" }, { "hash" : "4c8bc864776e0923683d1e60a1da38a12e9a52d0", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13005", "triggerID" : "4c8bc864776e0923683d1e60a1da38a12e9a52d0", "triggerType" : "PUSH" }, { "hash" : "af8de65a4905ae47ab704a75acfd1b2e897d915", "status" : "UNKNOWN", "url" : "TBD", "triggerID" : "af8de65a4905ae47ab704a75acfd1b2e897d915", "triggerType" : "PUSH" }, { "hash" : "772e076ed9d773e16713236c942f4e30b659d6eb", "status" : "UNKNOWN", "url" : "TBD", "triggerID" : "772e076ed9d773e16713236c942f4e30b659d6eb", "triggerType" : "PUSH" }, { "hash" : "95f80b654417779460301440b21d1c4d1b9476e3", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13276", "triggerID" : "95f80b654417779460301440b21d1c4d1b9476e3", "triggerType" : "PUSH" }, { "hash" : "06115e6dc53dfd654db8ee33449352d8f43abd89", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13284", "triggerID" : "06115e6dc53dfd654db8ee33449352d8f43abd89", "triggerType" : "PUSH" }, { "hash" : "354eb574c98bbc5fdaedabc9f3ca7e4acfaa3746", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13285", "triggerID" : "354eb574c98bbc5fdaedabc9f3ca7e4acfaa3746", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url" : "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/\_build/results?buildId=13321", "triggerID" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "triggerType" : "PUSH" }, { "hash" : "94d299053103c1bec5dfefaf4089ed0b31c403bb", "status" : "DELETED", "url"

```
"b895dddd2a1833c10dd04f2823dd19e497fc882c", "status": "DELETED", "url": "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/_build/results?buildId=13337", "triggerID":  
"b895dddd2a1833c10dd04f2823dd19e497fc882c", "triggerType": "PUSH" }, { "hash":  
"8e732bfb2bddc38ec7422f482dcda4be3d296408", "status": "DELETED", "url": "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/_build/results?buildId=13413", "triggerID":  
"8e732bfb2bddc38ec7422f482dcda4be3d296408", "triggerType": "PUSH" }, { "hash":  
"854b645872b5596d2cc618c65dac0228c616eec2", "status": "DELETED", "url": "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/_build/results?buildId=13478", "triggerID":  
"854b645872b5596d2cc618c65dac0228c616eec2", "triggerType": "PUSH" }, { "hash":  
"e962af6d8df77a725defeb92537837e6f685292f", "status": "DELETED", "url": "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/_build/results?buildId=13487", "triggerID":  
"e962af6d8df77a725defeb92537837e6f685292f", "triggerType": "PUSH" }, { "hash":  
"f05d87ce3393dca5b8df30919f439b8c272e982e", "status": "SUCCESS", "url": "https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/_build/results?buildId=13503", "triggerID":  
"f05d87ce3393dca5b8df30919f439b8c272e982e", "triggerType": "PUSH" } ] --> ## CI report: *  
af8de65a4905ae47ab704a75acfd1b2e897d915 UNKNOWN * 772e076ed9d773e16713236c942f4e30b659d6eb  
UNKNOWN * f05d87ce3393dca5b8df30919f439b8c272e982e Azure: [SUCCESS](https://dev.azure.com/apache-flink/98463496-1af2-4620-8eab-a2ecc1a2e6fe/_build/results?buildId=13503) <details> <summary>Bot  
commands</summary> The @flinkbot bot supports the following commands: - @flinkbot run travis` re-run the last  
Travis build - @flinkbot run azure` re-run the last Azure build </details>
```

3. **body:** Thanks @tillrohrmann for reviewing the PR. I addressed your changes. I removed code that would become only necessary when adding the support for the exception history collecting concurrent failures to be more consistent.  
**label:** code-design
4. @pnwojski @AHeise @rkhachatryan Could one of you guys have a look at the `Task`-related change I did in [211a68e](https://github.com/apache/flink/pull/14798/commits/211a68e9532b0bf10581388b395e474123ec6ad5)?
5. @rkhachatryan thanks for looking into it. There was no production code path I ran into that let me start this investigation. I ran into test issues after adding the invariant check to `TaskExecutionState`. That's where I started to check the code to see whether I could rely on the invariant to be true all the time. The only place where the `failureCause` is not set even though it's transitioning into `FAILED` state is [in the fall back case] (https://github.com/apache/flink/pull/14798/commits/af0bf68b965c9adaab61f42a99ef3278d27aed32#diff-b84174e55cb1999d99ad60cdeded7be20ff4978472bfc785c5a77b6270f47b56R856).
6. The test failures reported by Azure seem to be related to [FLINK-21277](https://issues.apache.org/jira/browse/FLINK-21277). I'm going to rebase and squash the related commits together after @rkhachatryan gave his final ok
7. @tillrohrmann I addressed all comments, squashed the changes and rebased the branch.
8. Thanks for the update. I'll merge this PR once AZP has completed.
9. There seems to be some test instability in the fine-grained source data now with the invariant being added. The [Azure build attached to this PR](https://dev.azure.com/apache-flink/apache-flink/\_build/results?buildId=13005&view=logs&j=a57e0635-3fad-5b08-57c7-a4142d7d6fa9&t=5360d54c-8d94-5d85-304e-a89267eb785a) succeeded. [Another Azure build on the same commit] (https://dev.azure.com/mapohl/flink/\_build/results?buildId=221&view=logs&j=cc649950-03e9-5fae-8326-2f1ad744b536&t=51cab6ca-669f-5dc0-221d-1e4f7dc4fc85&l=5837) failed in `TaskExecutorSlotLifetimeTest`. The test failure is caused by the newly introduced Precondition. To me, that looks like an issue in the fine-grained resource management code or some other parts of the code which we should investigate. I'd prefer to create a Jira issue covering this instead of relaxing the state change conditions again and work around it in the exception history. I still think that the invariant makes sense and we shouldn't relax it.
10. Ok, then let's first fix this instability before merging this PR @XComp. Ping me once the instability has been fixed.
11. The test instability is caused by the change adding the requirement that `failureCause` need to be set if the `executionState` is `FAILED`. The failing test runs a task but doesn't wait for the task to be finished. The `TaskExecutor` triggers the cancellation of the `Task` through [Task.failExternally] (https://github.com/XComp/flink/blob/354eb574c98bbc5fdaedabc9f3ca7e4acfaa3746/flink-runtime/src/main/java/org/apache/flink/runtime/taskmanager/Task.java#L1129) passing in a `FlinkException: Disconnect from JobManager responsible for [...]`. This will trigger the `Task` to transition into `FAILED`. A context switch happens after the state is set but before the `failureCause` is set. Then, the `Task` finishes in the main thread, cleans up and calls [notifyFinalState] (https://github.com/XComp/flink/blob/354eb574c98bbc5fdaedabc9f3ca7e4acfaa3746/flink-runtime/src/main/java/org/apache/flink/runtime/taskmanager/Task.java#L904) as part of this which initializes a `TaskExecutionState` and runs into the `IllegalStateException` since the `failureCause` is still not set. The solution would be to do the state change and `failureCause` atomically. But I'm not sure whether this would be a performance problem. We would have to switch from [compareAndSet] (https://github.com/XComp/flink/blob/354eb574c98bbc5fdaedabc9f3ca7e4acfaa3746/flink-runtime/src/main/java/org/apache/flink/runtime/taskmanager/Task.java#L1064) to a normal lock, I guess. Alternatively, we could remove the invariant again and handle the `null` in the exception history. @tillrohrmann what are your thoughts on that?
12. I looked through the code (supported by @AHeise): The race condition really only kicks in when cancelling/failing the task because that's when the `failureCause` becomes relevant. So, instead of synchronizing the [state transition] (https://github.com/XComp/flink/blob/5781449f38c1e36c1a2952518f9e30761d915f04/flink-

runtime/src/main/java/org/apache/flink/runtime/taskmanager/Task.java#L1052) we could add synchronize blocks for the [cancellation of a task] (<https://github.com/XComp/flink/blob/5781449f38c1e36c1a2952518f9e30761d915f04/flink-runtime/src/main/java/org/apache/flink/runtime/taskmanager/Task.java#L1130>) and while [handling failure handling during the normal Invokable execution] (<https://github.com/XComp/flink/blob/5781449f38c1e36c1a2952518f9e30761d915f04/flink-runtime/src/main/java/org/apache/flink/runtime/taskmanager/Task.java#L828>). The synchronization will only cover the state transition and setting the `failureCause`. Cancelling the corresponding task would be moved out of the synchronization block.

13. I created [FLINK-21376](<https://issues.apache.org/jira/browse/FLINK-21376>) to cover the `failureCause`/`executionState` synchronization. The code changes of FLINK-21187 will also handle cases where no `failureCause` is set for now.
14. I addressed all changes, squashed and rebased the branch.

#### github\_pulls\_reviews:

1. **body:** If this method returns `null`, then I would suggest to return an `Optional<ErrorInfo>`. This makes the contract clearer.  
**label:** code-design
2. Same here.
3. I guess that this field is `@Nullable`, right?
4. `@Nullable` missing.
5. **body:** I think wrapping `getFailureCause` in a `SerializedThrowable` is not necessary because that's what `ErrorInfo` does in the constructor.  
**label:** code-design
6. **body:** with `Optional<ErrorInfo>` this might look a bit nicer.  
**label:** code-design
7. I think `@Nullable` annotation is missing here.
8. I think `@Nullable` is missing here.
9. Explanation missing what `null` value means.
10. `@Nullable` is missing.
11. `@Nullable` is missing.
12. Alternatively you can introduce a `isLocalFailure()` method and let this method fail if `failingExecutionVertexId == null`. That way you enforce the contract that you can only ask for the failing execution vertex id if it is a local failure.
13. Why did you introduce a second `thenRunAsync`?
14. **body:** I think it is nicer if you have something like `failureHandlingResult.isGlobalFailure()`. `failureHandlingResult.getExecutionVertexIdOfFailedTask() == null` is not very expressive.  
**label:** code-design
15. One could use `FlinkMatchers.containsCause`.
16. `System.currentTimeMillis()` can be susceptible to clock resets. Hence, it can cause some test instabilities. Unfortunately, `Execution` also uses this to determine the state timestamp.
17. **body:** How would this log line look like? Wouldn't `failure.getMessage` be duplicated? Maybe also say "Archive global failure...".  
**label:** code-design
18. **body:** For what do we need the maps here? Wouldn't a single `List<ErrorInfo>` be good enough here? Is this kind of a preparation for supporting to capture the exceptions of multiple `Executions`?  
**label:** code-design
19. **body:** I switched to returning an `Optional` to make this more explicit.  
**label:** code-design
20. **body:** You're right. That was still some leftover of my initial work considering multiple exceptions per failure cause. I'm gonna refactor it to use a List here in order to be more consistent.  
**label:** code-design
21. Ah, thanks for the hint. I didn't know about `FlinkMatchers`, yet.
22. I removed this check. This if was a relict from a previous code version.
23. That's a good question. I did it when experimenting for convenience and didn't reiterate over it again. ðŸ‘ I refactored it now.
24. Correct me if I'm wrong here: Another way to measure the time would be `System.nanoTime()` since it ensures monotonicity. But using this in the test would require `Execution` to use `System.nanoTime()` as well which would not work as `System.nanoTime` cannot be used to compare timestamps between different JVMs [[1]] (<https://www.javaadvent.com/2019/12/measuring-time-from-java-to-kernel-and-back.html>)?!
25. **body:** nit: That the throwable is wrapped in a `SerializedThrowable` should be an implementation detail of the `ErrorInfo`. I think it is not necessary to mention it here.  
**label:** code-design
26. `` suggestion return Optional.ofNullable(failureInfo); ``

27. Does it make sense to change `failureCause` to be of type `ErrorInfo`? If yes, then we could save recreating this object multiple times.
28. ```suggestion return Optional.ofNullable(failingExecutionVertexId); ```
29. I think we should check here that we are indeed in the `FAILED` state. What can also happen is that the user cancels the job in between.
30. **body:** I would suggest to take the `ExecutionGraph.getStatusTimestamp`. That way the timestamp will be consistent.  
**label:** code-design
31. Is this correct? Why does CI passes with this assertion if it is wrong?
32. Yes, we are actually interested in a timestamp and not some measure of passed time. But for testing purposes a manually controlled `Clock` can help to avoid instabilities cause by `System.currentTimeMillis`. The production code could then still use `System.currentTimeMillis`.
33. Yes, it does. I actually had it like that in a first version of the code but changed it for the sake of consistency after realizing that it's not `Serializable` (which was a problem in `ArchivedExecution`). I will revert it.
34. Oh man, good catch. `ArchiveExecutionGraphTest` triggered a global failure which only set the `failureInfo` on the `ExecutionGraph` level. I switched to `SchedulerBase.updateTaskExecutionState` which also sets the `failureInfo` on an `Execution` level
35. Here, I don't understand fully: You're referring to the case where a failure happens, the user cancels the job while the failure handling is done and the `failJob` method might be called while being in a `CANCELED`/`CANCELLING` state? For this case, I would still think that we should archive the exception because the users intervention happened after the exception happened.
36. Thanks for clarification.
37. I mean the case that a failure happens, the job goes into the `FAILING` state and tries to cancel the tasks and now the user cancels the job because it takes too long for him. Then the job will go into the `CANCELING` state which will result to the `CANCELED` state once all tasks have terminated. I think you are right that we should still record the failure cause.
38. I think we can't update `failureCause` until we successfully transitioned to `FAILED` state.
39. That's actually a good point. I will correct that.
40. The comment seems to be wrong now.
41. Can't we say that if `t == null`, then we create an unknown failure cause? `new FlinkException("Unknown cause for Execution failure. This might be caused by FLINK-21376")`?
42. **body:** The JavaDoc seems wrong. If I am not mistaken, then we will also set this field if the failure is recoverable, right?  
**label:** documentation
43. Why do we have to ask the execution for the failure cause? If we are only interested in the root cause, doesn't `failureHandlingResult` contain all the required information?
44. I think `archiveFromFailureHandlingResult` can also be called when handling a global failover. In this case `executionOptional` would be empty and as a consequence we won't record a failure cause.
45. **body:** I think we also need a test for global failover.  
**label:** test
46. Because of the timestamp in case of local failure: The `Execution` fails but does not necessarily trigger the `ExecutionGraph` to switch to fail. Hence, no timestamp is record for this kind of failure.
47. **body:** But looking into that again, I realize that there is, indeed, the `failureInfo` in `ExecutionGraph` that is set during `[handleTaskFailure]` (<https://github.com/XComp/flink/blob/8e732bfb2bddc38ec7422f482dcda4be3d296408/flink-runtime/src/main/java/org/apache/flink/runtime/scheduler/DefaultScheduler.java#L218>) and `[handleGlobalFailure]` (<https://github.com/XComp/flink/blob/8e732bfb2bddc38ec7422f482dcda4be3d296408/flink-runtime/src/main/java/org/apache/flink/runtime/scheduler/DefaultScheduler.java#L236>). This is used to forward the current root cause of the `ExecutionGraph` to the Web UI. I could have used that. But the drawback is that it's using `System.currentTimeMillis`. Instead, I'm gonna work on removing the `ExecutionGraph.failureInfo` (and related methods) as part of FLINK-21190. It becomes redundant after we enable to exception history since it's just the exception history's last entry.  
**label:** code-design
48. You're right. I remember looking into that. Back then, I mistook the task as only focusing on "task failures". That was a misunderstanding. I added the test for global failover and the test.
49. So theoretically we would have to add the timestamp to the `FailureHandlingResult` or the `ErrorInfo` if we didn't want to look up the `Execution` again, right?
50. Shall we make these methods `final` since we don't want them to be overridden.
51. I think `getError` can return `null`. This needs to be handled properly.
52. **body:** Maybe it is a bit clearer to use `failureHandlingResult.isGlobalFailure()` because at the moment the contract is that `executionVertexIdOfFailedTask == null` iff it's a global failure.  
**label:** code-design
53. `@Nullable` is missing. One way to find nullability problems is to turn on IntelliJ's inspections. There are couple of those for null checks. This helps me quite a lot.
54. **body:** I'm hesitant to change that: IMHO, it makes the code harder to read and easier to misbehave. The current version of the code is straight forward: Use the `FAILED` timestamp provided the corresponding `Execution` if we have the corresponding `ExecutionAttemptId`. Additionally, your change request might cause unwanted behavior if we

decide to introduce global failures due to some local failure. I consider the fact that the global failure does not have a causing `Execution` an implementation detail which we don't have to expose here. @tillrohrmann Is that reasonable?

**label:** code-design

55. Yes, that would be possible if would would like to relax the constraint that the exception history does not provide the timestamp the `Execution` was marked as failed but an earlier timestamp. But to be fair, we do that already for the fallback of a global failover in `archiveFromFailureHandlingResult`.
56. Fair enough. Then let's not change it.
57. Ok, I think we don't have to change it right now. Maybe we do this sometime in the future.
58. I added this remark to FLINK-21190

#### jira\_issues:

1. **summary:** RootException history implementation in DefaultScheduler  
**description:**
2. **summary:** RootException history implementation in DefaultScheduler  
**description:**
3. **summary:** RootException history implementation in DefaultScheduler  
**description:**
4. **summary:** RootException history implementation in DefaultScheduler  
**description:**

#### jira\_issues\_comments:

1. Error identified in [Azure CI run|https://dev.azure.com/mapohl/flink/\_build/results?buildId=221&view=logs&j=cc649950-03e9-5fae-8326-2f1ad744b536&t=51cab6ca-669f-5dc0-221d-1e4f7dc4fc85&l=5837] could be reproduced: {noformat} 2021-02-09T17:43:25.8951082Z [ERROR] Tests run: 1, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 2.296 s <<< FAILURE! - in org.apache.flink.runtime.taskexecutor.TaskExecutorSlotLifetimeTest 2021-02-09T17:43:25.8951992Z [ERROR] testUserCodeClassLoaderIsBoundToSlot(org.apache.flink.runtime.taskexecutor.TaskExecutorSlotLifetimeTest) Time elapsed: 0.777 s <<< FAILURE! 2021-02-09T17:43:25.8952650Z java.lang.AssertionError: The TestingFatalErrorHandler caught an exception. 2021-02-09T17:43:25.8953244Z at org.apache.flink.runtime.util.TestingFatalErrorHandlerResource.after(TestingFatalErrorHandlerResource.java:78) 2021-02-09T17:43:25.8953928Z at org.apache.flink.runtime.util.TestingFatalErrorHandlerResource.access\$300(TestingFatalErrorHandlerResource.java:33) 2021-02-09T17:43:25.8954639Z at org.apache.flink.runtime.util.TestingFatalErrorHandlerResource\$1.evaluate(TestingFatalErrorHandlerResource.java:57) 2021-02-09T17:43:25.8955234Z at org.junit.rules.TestWatcher\$1.evaluate(TestWatcher.java:55) 2021-02-09T17:43:25.8955710Z at org.junit.rules.RunRules.evaluate(RunRules.java:20) 2021-02-09T17:43:25.8956163Z at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:325) 2021-02-09T17:43:25.8956692Z at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78) 2021-02-09T17:43:25.8957265Z at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57) 2021-02-09T17:43:25.8957789Z at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:290) 2021-02-09T17:43:25.8958262Z at org.junit.runners.ParentRunner\$1.schedule(ParentRunner.java:71) 2021-02-09T17:43:25.8958764Z at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288) 2021-02-09T17:43:25.8959262Z at org.junit.runners.ParentRunner.access\$000(ParentRunner.java:58) 2021-02-09T17:43:25.8959758Z at org.junit.runners.ParentRunner\$2.evaluate(ParentRunner.java:268) 2021-02-09T17:43:25.8960248Z at org.junit.rules.ExternalResource\$1.evaluate(ExternalResource.java:48) 2021-02-09T17:43:25.8960742Z at org.junit.rules.RunRules.evaluate(RunRules.java:20) 2021-02-09T17:43:25.8961200Z at org.junit.runners.ParentRunner.run(ParentRunner.java:363) 2021-02-09T17:43:25.8961719Z at org.apache.maven.surefire.junit4.JUnit4Provider.execute(JUnit4Provider.java:365) 2021-02-09T17:43:25.8962293Z at org.apache.maven.surefire.junit4.JUnit4Provider.executeWithRerun(JUnit4Provider.java:273) 2021-02-09T17:43:25.8962896Z at org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:238) 2021-02-09T17:43:25.8963477Z at org.apache.maven.surefire.junit4.JUnit4Provider.invoke(JUnit4Provider.java:159) 2021-02-09T17:43:25.8964085Z at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:384) 2021-02-09T17:43:25.8964702Z at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:345) 2021-02-09T17:43:25.8965259Z at org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:126) 2021-02-09T17:43:25.8965973Z at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:418) 2021-02-09T17:43:25.8966537Z Caused by: java.lang.IllegalArgumentException: A FAILED execution state should always provide a cause. 2021-02-09T17:43:25.8967100Z at org.apache.flink.util.Preconditions.checkArgument(Preconditions.java:138) 2021-02-09T17:43:25.8967726Z at org.apache.flink.runtime.taskmanager.TaskExecutionState.<init>(TaskExecutionState.java:111) 2021-02-09T17:43:25.8968333Z at org.apache.flink.runtime.taskmanager.TaskExecutionState.<init>(TaskExecutionState.java:84) 2021-02-09T17:43:25.8968903Z at org.apache.flink.runtime.taskmanager.Task.notifyFinalState(Task.java:1005) 2021-02-09T17:43:25.8969415Z at org.apache.flink.runtime.taskmanager.Task.doRun(Task.java:897) 2021-02-09T17:43:25.8969891Z at

org.apache.flink.runtime.taskmanager.Task.run(Task.java:565) 2021-02-09T17:43:25.8970326Z at  
java.lang.Thread.run(Thread.java:748) {noformat}  
2. Fixed via c77a686c195d1742c276f4a9e75899c8b85377bb