How does ShuffleVertexManager (Auto Reduce Parallelism) work

"tez.shuffle-vertex-manager.desired-task-input-size" - Determines the amount of desired task input size per reduce task. Default is around 100 MB.

"tez.shuffle-vertex-manager.min-task-parallelism" - Min task parallelism that ShuffleVertexManager should honor. I.e, if the client has set it as 100, ShuffleVertexManager would not auto-reduce to less than 100 tasks.

"tez.shuffle-vertex-manager.min-src-fraction"
"tez.shuffle-vertex-manager.max-src-fraction"
- Both of these parameters are useful for determining the slow-start behaviour.

In Tez, when a source task generates output, DataMovementEvent (via RPC) is sent out and its payload carry details like outputsize. ShuffleVertexManager keeps aggregating these values from different source tasks and checks periodically on whether it can determine the value for auto-reduce parallelism. If the aggregated data size is less than configured "tez.shuffle-vertex-manager.desired-task-input-size", it waits for stats to be generated from additional source tasks. It is possible that by this time, "tez.shuffle-vertex-manager.min-src-fraction" reaches it limits. But min-src-fraction config is dynamically overriden at runtime as it is better to wait for data from more tasks to determine more accurate value for auto-parallelism. ShuffleVertexManager tries to schedule the task as it crosses the min-src-fraction and finishes scheduling when max-src-fraction limit is reached.

There can be scenarios where auto-reduce computed value is greater than currently configured parallelism of the vertex depending on the amount of data emitted by source tasks. In such cases, existing parallelism is used.

Following method contains details on how parallelism is determined at runtime.
https://github.com/apache/tez/blob/fd75e640396da8d5e1c67ef554d5db1846e08c69/tez-runtime-library/src/main/java/org/apache/tez/dag/library/vertexmanager/ShuffleVertexManager.java#L669

It is also possible for sources to send the per-partition stats along with the DataMovementEvent payload. Retaining all details in the same payload can be fairly expensive. Currently, per-partition details are bucketted into one of the data range (0, 1, 10, 100, 1000 MB) and are stored in RoaringBitMap in the payload. This can be a little noisy, but atleast provides better hints to ShuffleVertexManager. Based on this info, ShuffleVertexManager can schedule the reducer task which would get the maximum amount of data and this can be useful in cases where there are data skews. This can be enabled via **"tez.runtime.report.partition.stats"** (not enabled by default)

ShuffleVertexManager mainly works on scatter-gather sources (When unordered sources are involved, downstream vertex can start processing the data as soon as some data is made available by the source tasks. So downstream tasks can start as soon as the sources start. ShuffleVertexManager would not be able to best job in such cases with limited or no information on the stats for unordered cases).

Real world examples:

Consider Hive which makes use of Tez as an execution engine. Hive sets "tez.shuffle-vertex-manager.desired-task-input-size" and "tez.shuffle-vertex-manager.min-task-parallelism" at the time of creating the DAG. Task-input-size is determined by various factors like the statistics gathered in metastored, amount of data that needs to be sent to a reducer and so on. Min-task-parallelism is determined internally in Hive by couple of other parameters like "hive.tez.max.partition.factor / hive.tez.min.partition.factor" along with data size per reduce task. For instance, assume initial reduce task number is 100 & hive.tez.max.partition.factor=2.0 and hive.tez.min.partition.factor=0.25. In this case, Hive would set the reducers to 200 and the hint to tez for its min-task-parallelism would be 25, so that Tez would not try to auto-reduce below 25 tasks. This serves as a safety net.
Refer:
https://github.com/apache/hive/blob/c76eef2f9db4b6863a9665ed000276d6544309d0/common/src/java/org/apache/hadoop/hive/conf/HiveConf.java#L372
https://github.com/apache/hive/blob/c76eef2f9db4b6863a9665ed000276d6544309d0/common/src/java/org/apache/hadoop/hive/conf/HiveConf.java#L2450
https://github.com/apache/hive/blob/2f73233961e6f527f5460c124cc021a4a9628b01/ql/src/java/org/apache/hadoop/hive/ql/exec/tez/DagUtils.java#L1241

Pig sets "tez.shuffle-vertex-manager.desired-task-input-size" and min/max source fractions for auto-reduce parallelism.
Refer https://github.com/apache/pig/blob/5fe8a4a3aef7ecdc8974575a413d74a2e137aa5f/src/org/apache/pig/backend/hadoop/executionengine/tez/TezDagBuilder.java#L810