A stage is a set of parallel tasks all computing the same function that need to run as part

of a Spark job, where all the tasks have the same shuffle dependencies. Each DAG of tasks run

by the scheduler is split up into stages at the boundaries where shuffle occurs, and then the

DAGScheduler runs these stages in topological order.

Each Stage can either be a shuffle map stage, in which case its tasks' results are input for

other stage(s), or a result stage, in which case its tasks directly compute a Spark action

(e.g. count(), save(), etc) by running a function on an RDD. For shuffle map stages, we also

track the nodes that each output partition is on.

Each Stage also has a firstJobId, identifying the job that first submitted the stage. When FIFO

scheduling is used, this allows Stages from earlier jobs to be computed first or recovered

faster on failure.

Finally, a single stage can be re-executed in multiple attempts due to fault recovery. In that

case, the Stage object will track multiple StageInfo objects to pass to listeners or the web UI.

The latest one will be accessible through latestInfo.