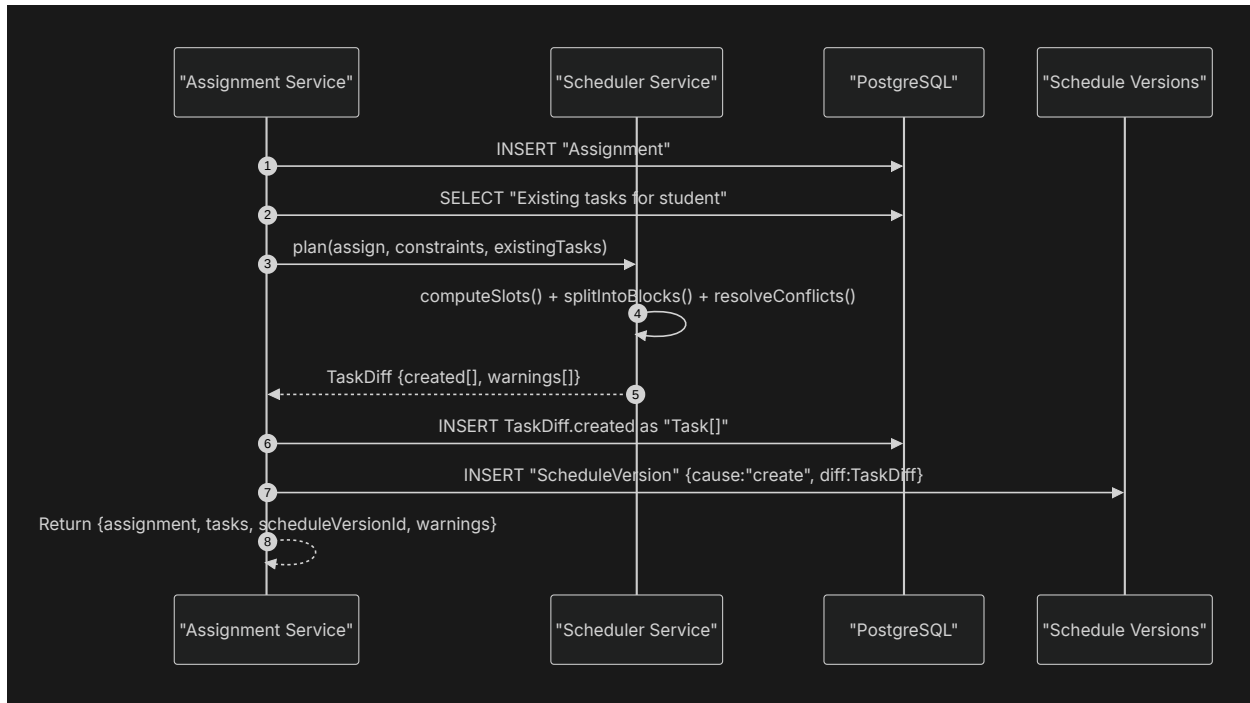


scheduler-Seq-Diagram

1) Plan new assignment (create → schedule → persist)

```
sequenceDiagram
    autonumber
    participant AS as "Assignment Service"
    participant SCH as "Scheduler Service"
    participant DB as "PostgreSQL"
    participant SV as "Schedule Versions"

    AS->>DB: INSERT "Assignment"
    AS->>DB: SELECT "Existing tasks for student"
    AS->>SCH: plan(assign, constraints, existingTasks)
    SCH->>SCH: computeSlots() + splitIntoBlocks() + resolveConflicts()
    SCH->>AS: TaskDiff {created[], warnings[]}
    AS->>DB: INSERT TaskDiff.created as "Task[]"
    AS->>SV: INSERT "ScheduleVersion" {cause:"create", diff:TaskDiff}
    AS->>AS: Return {assignment, tasks, scheduleVersionId, warnings}
```



2) Replan on update (due date or estimate changes)

sequenceDiagram

autonumber

participant AS as "Assignment Service"

participant SCH as "Scheduler Service"

participant DB as "PostgreSQL"

participant SV as "Schedule Versions"

AS→>DB: SELECT "Assignment + completed vs future tasks"

AS→>AS: remaining = new_est - actual_logged

AS→>DB: SELECT "Other tasks that may conflict"

AS→>SCH: replanRemaining(remaining, constraints, futureTasks, otherTasks)

SCH→>SCH: minimizeChanges() + honorCaps() + resolveConflicts()

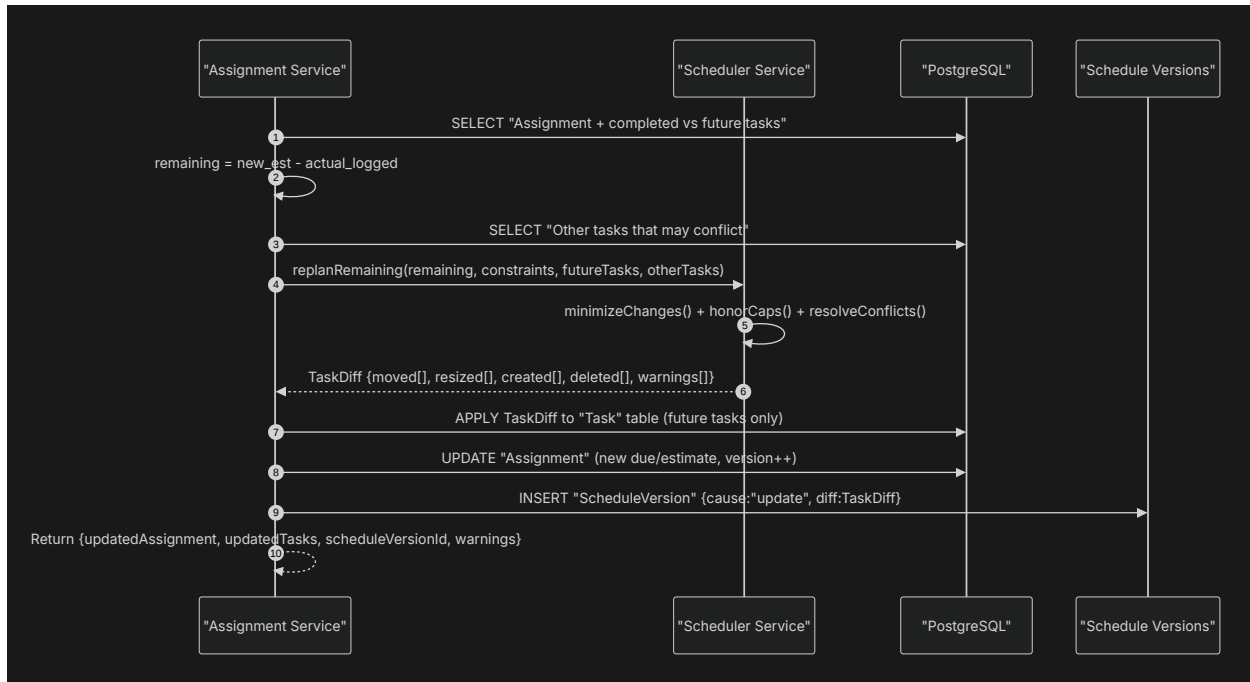
SCH→>AS: TaskDiff {moved[], resized[], created[], deleted[], warnings[]}

AS→>DB: APPLY TaskDiff to "Task" table (future tasks only)

AS→>DB: UPDATE "Assignment" (new due/estimate, version++)

AS→>SV: INSERT "ScheduleVersion" {cause:"update", diff:TaskDiff}

AS→>AS: Return {updatedAssignment, updatedTasks, scheduleVersionId, warnings}



3) Adapt on overrun (execution feedback → heal plan)

sequenceDiagram

autonumber

participant TS as "Telemetry / Timer"

participant AS as "Assignment Service"

participant SCH as "Scheduler Service"

participant DB as "PostgreSQL"

participant SV as "Schedule Versions"

TS→>DB: INSERT "WorkSession" for task

TS→>AS: notifyOverrun(taskId, actual > planned)

AS→>DB: SELECT "Task + nearby tasks and caps"

AS→>SCH: adapt(strategy, taskContext, constraints)

SCH→>SCH: choose(extend_now | add_follow_up | shift_others)

SCH→>AS: TaskDiff {resized[] | created[] | moved[], warnings[]}

AS→>DB: APPLY TaskDiff to "Task" table (future tasks only)
 AS→>SV: INSERT "ScheduleVersion" {cause:"slip", diff:TaskDiff}
 AS→>TS: ack with updated plan summary

