

1 – Admin invokes prepare_scheduled passing the schema as parameter

- 2 Precomputation on GTFS data is done(viz.):
 - a. stop_id_map table is created and populated to map textual stop ids into integer ids
 - b. shortest_time_graph table is created and shortest possible transit time between directly reachable stops are calculated.
 - c. shortest_time_closure table is created and populated using APSP SQL function in pgrouting core library.
- 3 User query invokes scheduled_route SQL function
- 4 Passes to libtransit_routing shared library and scheduled_route c function Is invoked.
- 5 Extracts query parameters from Datum and passes to C++ function, compute_scheduled_route
- 6 compute_scheduled_route instantiates TransitGraph
- 7 compute_scheduled_route instantiates AstarHeuristic
- 8 AstarHeuristic constructor calls fetch_shortest_time
- 9 fetch_shortest_time retrieves shortest time(heuristic) from shortest_time_closure table.
- 10 compute_scheduled_route instantiates AstarVisitor
- 11 compute_scheduled_route calls boost_astar function passing the transit graph object, astar heuristic object and astar visitor object.
- 12 Upon exploration of new vertices, examine_vertex function is triggered.
 13 examine_vertex function requests next set of trip instances from current vertex at the specified time.
- 14 fetch_next_links calls the PL/PGSQL function with the same name through SPI.
- 15 fetch_next_links PL/PGSQL function performs a complex(~7joins) query and fetches the trip instances from that stop at that instant of time(with a limit on waiting time).