



## Performance Analysis based on Persival/AVL Codebase

Protobuf vs. Flatbuf (full tables) vs. Flatbuf (partial structs)

2023-02-22

# Goal

- See if structs significantly improve performance compared to tables-only flatbuffers
- Only use structs for
  - basic OSI fields that will most probably not change in future
  - OSI fields that usually imply performance loss (e.g. Reflection)

## Adaption of Persival/AVL Codebase (feature/flatbuffers\_examples)

[https://github.com/OpenSimulationInterface/osi-sensor-model-packaging/tree/feature/flatbuffers\\_examples](https://github.com/OpenSimulationInterface/osi-sensor-model-packaging/tree/feature/flatbuffers_examples)

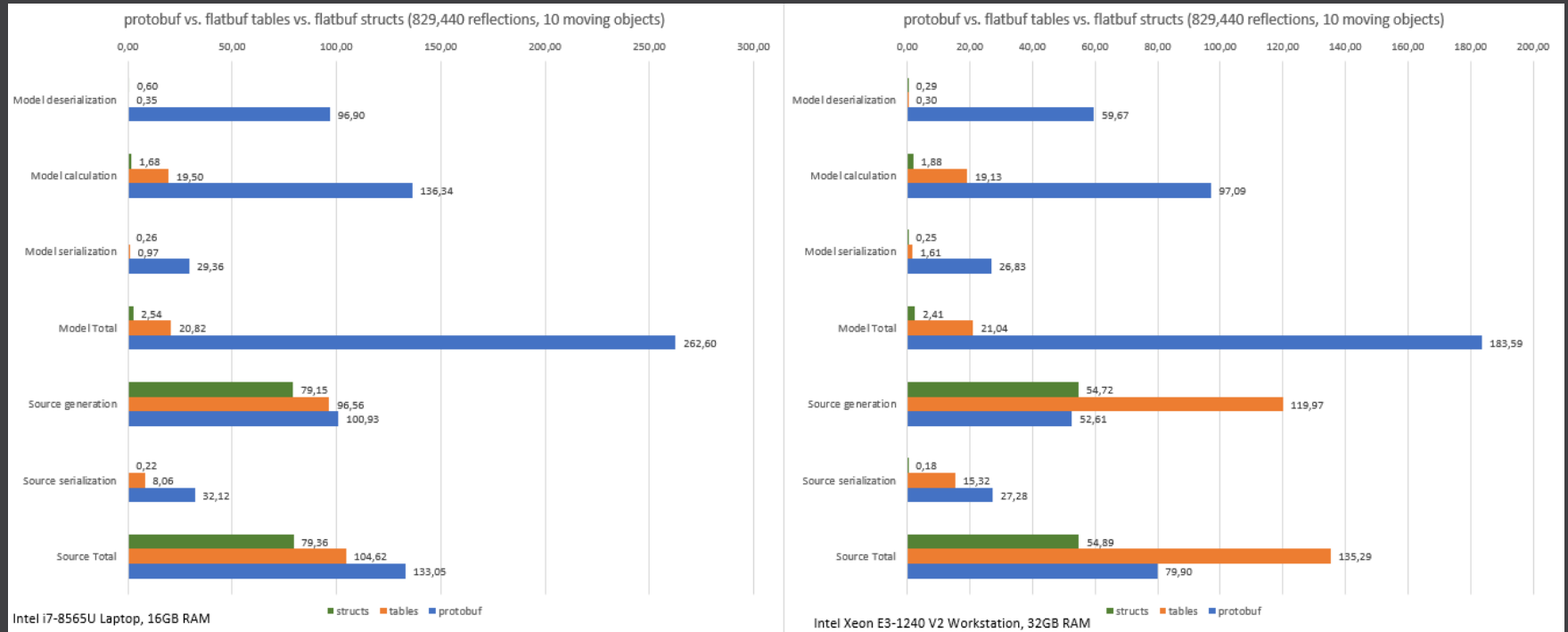
- .fbs schema transformation from tables to structs
  - osi\_common:
    - Vector3d
    - Vector2d
    - Timestamp
    - Dimension3d
    - Orientation3d
    - Identifier
    - MountingPosition
    - Spherical3d
  - osi\_sensorview:
    - Reflection

```
#ifndef USE_STRUCTS
    auto dimension = & osi3::Dimension3d(5.0, 2.0, 1.5);
    auto position = & osi3::Vector3d(source_x_offsets[i % 10]+time*x_speed, source_y_offsets[i % 10]+sin(time/x_speed)*0.25, 0.0);
    auto velocity = & osi3::Vector3d(x_speed, cos(time/x_speed)*0.25/x_speed, 0.0);
    auto acceleration = & osi3::Vector3d(0.0, -sin(time/x_speed)*0.25/(x_speed*x_speed), 0.0);
    auto orientation = & osi3::Orientation3d(0.0, 0.0, 0.0);
    auto orientation_rate = & osi3::Orientation3d(0.0, 0.0, 0.0);
#else
    auto dimension = osi3::CreateDimension3d(builder, 5.0, 2.0, 1.5);
    auto position = osi3::CreateVector3d(builder, source_x_offsets[i % 10]+time*x_speed, source_y_offsets[i % 10]+sin(time/x_speed)*0.25, 0.0);
    auto velocity = osi3::CreateVector3d(builder, x_speed, cos(time/x_speed)*0.25/x_speed, 0.0);
    auto acceleration = osi3::CreateVector3d(builder, 0.0, -sin(time/x_speed)*0.25/(x_speed*x_speed), 0.0);
    auto orientation = osi3::CreateOrientation3d(builder, 0.0, 0.0, 0.0);
    auto orientation_rate = osi3::CreateOrientation3d(builder, 0.0, 0.0, 0.0);
#endif
```

```
double attenuation = 1.0;
#ifdef USE_STRUCTS
    std::vector<osi3::LidarSensorView::Reflection> reflection_vector;
    double signal_strength = max_emitted_signal_strength_in_dB + 10 * std::log10(attenuation) - 10 * std::log10(rays_per_beam);
    double time_of_flight = 0.0;
    double doppler_shift = 0.0;
    auto normal_to_surface = osi3::Vector3d(0.0, 0.0, 0.0);
    auto object_id = osi3::Identifier(0);
    auto ray = osi3::LidarSensorView::Reflection(signal_strength, time_of_flight, doppler_shift, normal_to_surface, object_id);
    reflection_vector.push_back(ray);
#else
    osi3::LidarSensorView::ReflectionBuilder reflection_builder(builder);
    reflection_builder.add_time_of_flight(const_distance * 2.0 / speed_of_light);
    reflection_builder.add_signal_strength(max_emitted_signal_strength_in_dB + 10 * std::log10(attenuation) - 10 * std::log10(rays_per_beam));
    auto ray = reflection_builder.Finish();
    reflection_vector.push_back(ray);
#endif
```

# Benchmarks (Protobuf vs. Flatbuf structs vs. Flatbuf tables)

(Average over 1000 time steps, in ms)



# Benchmarks (Protobuf vs. Flatbuf structs vs. Flatbuf tables)

(Average over 1000 time steps, in ms)

