Who We Are

Bill Taggart
Software Engineer

Tim Gerber
Database Administrator
How Garmin Connect Manages and Analyzes 6 Billion Miles of Fitness GPS Data
Program Agenda

- Garmin – The Company
- Overview of Garmin’s use of Oracle Spatial
- Segments and Leader Boards
- Challenges and their Solutions
- Benefits
- Q&A
Garmin is the global leader in satellite navigation.
Since 1989, Garmin has designed, manufactured, marketed and sold navigation, communication and information devices and applications, most of which are enabled by GPS technology.
Garmin’s products serve automotive, aviation, marine, outdoor, sports and fitness industries.
Garmin has approximately 10,000 associates in 35 offices worldwide.
What is Garmin Connect

- Web based portal to upload, track, and analyze a user's outdoor, wellness and fitness activities.
- Allows users to interact/connect with each other around their outdoor/fitness data.
- Track and plan your current and future activities.
- The 6 billion miles mentioned in this presentation come from Garmin Connect user activities.
• Fitness customers wanted a way to challenge one another.
• Segments provide a way to compete along a stretch of road or trail.
• Leaderboards rank user activities on a given challenge segment.
• Oracle Spatial and Graph plus Linear Referencing System are used to match user activities to segments and extract the elapsed time for ranking.
• Segments are creating from either a running or cycling activity
Garmin Segments and Leaderboards
Defining Segments from User Activities
Garmin Segments and Leaderboards
Ranking Users Activities
Meet the Edge 1000
The file sent to an Edge 1000 is like one drop from a distillery.

There is a lot of energy spent behind the scenes to produce it.

Oracle Spatial drives the process.
Garmin Segments and Leaderboards
Matching with Oracle Spatial and Graph

Criteria:
- Start
- End
- Direction
- Path
Garmin Segments and Leaderboards
Segment Matching Challenges

GPS Variance:
- Buildings
- Trees
- Canyons
- Satellite drift
- Equipment
• SDO_ANYINTERACT casts a broad net for possible match candidates.

• SDO_GEOM.SDO_BUFFER used to ensure that the match candidate stays within an acceptable range along the segment.
• Needed a way to measure elapsed time.
• LRS time measure provides a way to obtain the time entered & exited for the match candidate.
• Time comparison tests directionality.
Proof of concept challenges and testing

- Define data type (Oracle LRS w/time measure)
- Spatially index data
- Load test data
- Discovered some interesting activities
DEFINE USER_SDO_GEOM_METADATA

INSERT INTO USER_SDO_GEOM_METADATA(
    table_name, column_name, diminfo, srid
) VALUES (  
'SEGMENT', 'POLYLINE',
SDO_DIM_ARRAY(  
SDO_DIM_ELEMENT('Longitude', -180, 180, 0.05),
SDO_DIM_ELEMENT('Latitude', -90, 90, 0.05),
SDO_DIM_ELEMENT('Time', 0, 0, 0.05)), 8307)
new JGeometry(
    3302, // Line string; 3D (x, y, m), 3rd is dimension
    8307, // SRID for latitude/longitude (8307)
    {1,2,1}; // offset, line, straight,
    SDO.ordinates(geom)
);
SELECT COUNT(*) FROM GEO_ACTIVITY a WHERE SDO_GEOM.VALIDATE_GEOMETRY ( a.polyline, .005) = 'FALSE';
UPDATE GEO_ACTIVITY a
SET a.polyline =
SDO_UTIL.REMOVE_DUPLICATE_VERTICES
( a.polyline, .005)
Example of creating a 2D Line & MapViewer

INSERT INTO SPATIAL_LEARNING
(SPATIAL_NAME, SPATIAL_DATA)
VALUES
('Kansas Speedway Track',
SDO_GEOMETRY(
 2002, -- two dimensional line
8307, -- SRID for WGS-84 longitude/latitude
NULL,
SDO_ELEM_INFO_ARRAY(1, 2, 1),
SDO_ORDINATE_ARRAY(
-94.838234424591,39.11655560213989,
-94.8394682407379,39.11618478057385,
-94.8402192592621,39.11574920989278,
-94.8404874801636,39.1153254080791,
-94.8400046825409,39.114819198615976,
-94.8388562896729,39.11417171516305,
-94.8353912830533,39.11296502557993,
-94.8324944972992,39.11248823034213,
-94.8288466930699,39.11187604407223,
-94.824501544348,39.11176004407223,
-94.8188152313232,39.1116700186508,
-94.813558101654,39.1115581760269,
-94.808676481247,39.11156994894302,
-94.8044385910034,39.111611154134025,
-94.8295569705963,39.111711223783246,
-94.8294071292877,39.111934908367736,
-94.8290383157654,39.1122368831305,
-94.8287057876580,39.11257562599992,
-94.8284711768341,39.11292382118056,
-94.8269605484009,39.11707357185418,
-94.82670485973358,39.117479704535015,
-94.82674241065979,39.118021211134604)));
User paused multisport device at a triathlon in one city and then resumed it at home.
We found a few instances of bad GPS data.
Indoor treadmill activity with GPS enabled.
Popular segment that is a loop and one activity frequently crosses the segment multiple times.
In order to handle the following data issues, we build a process to scrub all of our activities (6 billion miles worth):
- One activity crossing a segment multiple times (make sure they get credit for each match to the segment)
- User usage issues
- Separate data structure to contain the scrubbed activities
- Define scrubber rules
- Complete initial load of 300 million historical activities
Activity Scrubbing Filters

Point Filter Class Diagram

- **ValidLatLonFilter**
  - double: MAX_LAT
  - double: MAX_LON
  - boolean: isValid(TrackContext)

- **TimestampFilter**
  - boolean: isValid(TrackContext)

- **MinimumDistanceFilter**
  - double: maxDistanceMeters
  - boolean: isValid(TrackContext)

- **MinimumSpeedFilter**
  - double: minMetersPerSecond
  - boolean: isValid(TrackContext)

- **MaximumElapsedTimeFilter**
  - double: maxElapsedTime
  - boolean: isValid(TrackContext)

- **MaximumDistanceFilter**
  - double: maxDistanceMeters
  - boolean: isValid(TrackContext)

- **MaximumSpeedFilter**
  - double: maxMetersPerSecond
  - boolean: isValid(TrackContext)

---

**PointCloud**

- Coordinate[]: track
- List<Coordinate>**: validatedTracks
- int: pointIndex
- int: trackIndex
- Coordinate: previousPoint
- Coordinate: currentPoint
- Coordinate: nextPoint
- double: accumulatedDistance

---

**Garmin**
• Segments give users a way to compete with each other.
• Oracle Spatial and Graph allows us to match user activities to segments.
• Oracle LRS allows us to measure the elapsed time of each segment match (used to build the segment leaderboard).
• The difficult part was provisioning clean data and overcoming differences in GPS data.
• Today we have over 125K segments with 52+ million and growing segment matches
Two Exadata Half-Racks

- One \(\frac{1}{2}\) rack for Connect production database.

- One \(\frac{1}{2}\) rack for all non-production Connect databases and production Data Guard standby database.
• Running Oracle 11.2.0.3

• Ran the same spatial query on a commodity server and on Exadata -> Result was 20X faster on Exadata.

• We have even further performance gains by applying some of the backported Oracle 12C Spatial Patches.
  • Patches numbers: 13950749, 16512844, 18907724
The Garmin Connect database is 50 TB and growing at a pace of 2 TB per month.
• Of this 50 TB, 5.5 TB is spatial data used for segments and leaderboards.
• Putting this data into a single non-partitioned table is unrealistic.
• To get the best performance, we needed to partition the tables.
Our Partitioning Strategy

• All of our queries will be based around activity type (i.e. running and cycling) and date.
• Spatial indexing allows for partitioning but not sub-partitioning.
• Therefore we chose to use a composite key of activity_type, date with range partitioning.
• Tested global indexes (both traditional and Spatial Indexes).

• Saw 5x improvement when using local spatial indexes.

• CREATE INDEX LRS_GEO_ACTIVITY_IDX ON GEO_ACTIVITY ("POLYLINE")
  INDEXTYPE IS "MDSYS"."SPATIAL_INDEX"  PARAMETERS
  ('SDO_INDX_DIMS=2') LOCAL PARALLEL 6 ;
Running tests with different degrees of parallelism helped drive out the optimal number of parallel processes.
The Performance Impact

- Iteratively return rows as they are produced instead of in a batch after all processing of the table function's input is completed.

- Execution of a table function can be parallelized, and returned rows can be streamed directly to the next process without intermediate staging.
Parts of our segment matching processing is utilizing a PL/SQL package.
To help performance within the package, we have taken advantage of parallel pipelined functions.

```sql
get_sections_for_segment (segment_pk_v NUMBER,
                        buffer_dist    NUMBER,
                        interval_in_meters NUMBER,
                        max_skip_percent  NUMBER,
                        source_table_cursor IN for_segment_cursor_type)
RETURN activity_segment_table_type DETERMINISTIC
PIPELINED PARALLEL_ENABLE
(PARTITION source_table_cursor BY HASH (geo_activity_pk))
```
• How to test a new functionality without any true baseline?
• How to create real life data for testing?
• How to manage the required resources needed by the Segments feature and providing the rest of the application sufficient resources?
We wrote a java program that created test data from segments from real life data and also created segments from known popular rides/runs.

To test performance of matching this test data, we created a PL/SQL program that walked through each segment and recorded the run time to a table.

Test, test, and test again before migrating to production.
- Evaluate AWR reports
- Second option is to use Real Application Testing (RAT) but some of the recommendations for using RAT can be difficult to accomplish.
• Oracle Spatial queries can be CPU intensive

• How to manage the required resources needed by the Segments feature and providing the rest of the application sufficient resources?
• DB Resource Manager is instrumental in allowing us to prioritize work from the application including the Segments feature.

• DBRM allows us to prioritize our CPU cycles and we can easily change this priority based on application peak times and low times.
Each feature of our application connects to the database using separate DB Services.

```
srvtl add service -d <DB> -s <DB Service Name> -r <DB Inst1> -a <DB Inst2> -l primary
```

Then DBRM is implemented to prioritize the work based off the DB Service Names allowing different parts of the application to have priority over others.
• DBRM for Intra DB Resource Management
  • [http://docs.oracle.com/cd/E11882_01/server.112/e25494/dbrm.htm#g1021210](http://docs.oracle.com/cd/E11882_01/server.112/e25494/dbrm.htm#g1021210)

• CPU/Instance Caging
• A DBRM plan should be enabled.

```
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN='DEFAULT_PLAN';
ALTER SYSTEM SET cpu_count=12 scope=both sid='*';
OR
ALTER SYSTEM SET cpu_count=12 scope=both sid='CONCTPRD1';
```
How do you process, analyze and load 300+ million fitness activities in a minimal time frame all while new activities (approximately 1 million per day) are being created?
The Data Load Challenge
Possible Options

- Transportable Tablespaces
- Partition Swapping
- Export/Import of table
- Custom Process
• We were able to utilize several queues (using Oracle Advanced Queuing) to place current activities being created in one queue and the historical activities in another queue.
• Processed both queues continuously until spatial data for all activities was created.
• We were able to process, analyze, load all 300+ Million fitness activities in the database in under 20 days.
The Data Load Challenge

The Load Process
Benefits
The Results are In!

• Provides “clean” versions of our data.
• Able to provide users results within seconds.
• Allows real time features for Garmin products.
• Enables additional reporting information about Garmin products.
OVERVIEW
• Allow users to compete for fastest time climbing a hill or sprinting down a straightaway and rank the leaders.
• Users define the starting line, path, and finish line for each competitive “segment”
• Match a user’s fitness activities to segments and rank the activity

CHALLENGES / OPPORTUNITIES
• Be able to match activities to segments within seconds
• Initial loading for 300+ million activities
• Data quality

SOLUTIONS
• Oracle Exadata Machine (Half Rack)
• Oracle Database 11g Enterprise Edition
  • Spatial and Graph
  • Partitioning
  • Parallel Pipelined Functions

RESULTS
• Stores and simplifies processing of more than 6 billion miles of user activities in a 50TB database
• Able to process and match activities to segments in seconds and able to match new segments to 5+ TB of activity data in minutes
• Allows for additional real time features on Garmin devices (Edge 1000)
• Enables additional reporting features about the use of Garmin Fitness and Wellness products