

Parallel Game Feature System Summary

System Overview

I've implemented a parallel processing system for generating game-level and publisher-level features for video game market analysis. The system processes data from three main perspectives:

1. Individual game metrics
2. Genre-relative performance metrics
3. Audience overlap metrics

Key Components

Publisher/ParallelPublisher Class

- Handles collection and aggregation of game data for a publisher
- Implemented in both standard (Publisher) and parallel (ParallelPublisher) versions
- Manages parallel processing of game feature calculations for all games produced by the publisher
- Aggregates individual game features into publisher-level metrics

Game Class

- Represents individual games with historical performance data
- Calculates base features like player growth, revenue metrics, etc.
- Manages relationships with similar games through Genre and AudienceOverlap classes
- Key features implemented for serialization in multiprocessing context

Genre/ParallelGenre Class

- Calculates game performance relative to genre peers
- Handles parallel loading and processing of similar games by genre
- Implements similarity scoring based on genre and tag overlap
- Manages broader searches when insufficient peer games are found

AudienceOverlap/ParallelAudienceOverlap Class

- Processes game performance relative to games with similar player bases
- Implements parallel loading of games with audience overlap
- Handles weighted averaging based on overlap scores
- Gracefully handles missing data cases

Parallel Processing Implementation

Key Optimizations

1. Multi-level parallelization:
 - Publisher-level: Processing multiple games concurrently
 - Game-level: Loading similar games in parallel
 - Feature-level: Calculating aggregates concurrently
2. Error Handling:
 - Graceful handling of missing audience overlap data
 - Recovery from API failures
 - Proper handling of games with insufficient peer data
3. Resource Management:
 - Conservative worker count (`max_workers = min(4, CPU_count - 1)`)
 - Rate limiting for API calls
 - Efficient memory usage through proper cleanup

Current Status

Working Features

- Parallel processing of publisher game portfolios
- Concurrent loading of similar games
- Parallel feature calculation
- Error handling for missing data cases

Known Edge Cases

- Games without audience overlap data (e.g., games 71167, 71110)
- Games with insufficient genre peers requiring broader search
- API rate limiting considerations

Data Flow

1. Initialize publisher with game IDs
2. Parallel load of game data
3. Concurrent calculation of base features
4. Parallel processing of genre and audience overlap metrics
5. Aggregation into publisher-level features