dJay: Enabling High-density Multi-tenancy for Cloud Gaming Servers with Dynamic Cost-Benefit GPU Load Balancing

Sergey Grizan, David Chu, Alec Wolman, Roger Wattenhofer

Density Optimization Problem

\[
\max \sum Q(p_i)
\]

Challenges:
- Predicting GPU_TIME(p_i)
- Predicting Q(p_i)

... for arbitrary scenes
... for all graphics settings

Impacted by:
- Player churn: join/leave
- Scene change: outdoors/small room

System Architecture

Online Profiling
- dJay: concurrent execution
  - Session #18 (master instance)
  - Session #28 (slave instance)

per-frame rendering setting, frame-time:
  - SHD: Active Object List (AOL)

dJayT: utility profiler

Online Utility Maximization
- Session #13
- Session #14
- Session #15
- Session #16

per-frame: utility estimation

GPU_TIME(p_i) < 33ms

Key Insight: Tunable Visual QoS

System Architecture

Online Profiling
- dJayT: concurrent execution
  - Session #18 (master instance)
  - Session #28 (slave instance)

per-frame: rendering setting, frame-time:
  - SHD: Active Object List (AOL)

dJayT: utility profiler

Online Utility Maximization
- Session #13
- Session #14
- Session #15
- Session #16

per-frame: utility estimation

GPU_TIME(p_i) < 33ms

Key Insight: Tunable Visual QoS

Utility Estimating: Cost and Benefit

Solving for Optimal Settings

Building Object Tables

Optimal Settings

Building Object Tables

Object Tables by Clustering

Increasing multi-tenancy

Increasing scene complexity

Responsiveness

Results on Fable 3