System Architecture

* Threading
  + To increase the performance of calculation, threading has been introduced to the system.
  + Separate threads have been created for:
    - Visualization
    - Networking
      * UDP Broadcasting/Listening
      * TCP sending/receiving
    - Simulation
      * Collision Detection
        + Ground-Ball
        + Wall-Ball
        + Ball-Ball
      * Collision Response
* Process Affinity
  + As required from the specification, threads for specific application have been allocated to specific processors. Therefore Processor 1 has been set for Visualization application. Core 2 has been assigned for networking usage. All other remaining cores have been used for Simulation. 3 cores has been assigned to perform 3 different collision detection: Core 4 for the collisions between the ground and balls. Core 5 for the collisions between balls and Core 6 for the collisions between Wall and balls.
* Networking
  + UDP
    - Listener
      * When a new peer tried to connect to the p2p network, the UDP listener would listen to the specified port for Host address in
    - Broadcaster
      * While the peer is hosting TCP network, a thread would be created to broadcast its own Ip address. This would allow other peers connect to the Host without supplied the Host address manually.
  + TCP
    - Host
      * Host the network and exchange data with clients
    - Client
      * Join the network and exchange data with host and other clients
* UML diagrams

Physics implementations

* Collisions detection
* Collisions response

Research Questions

* Visual impact from the perspective of a peer on a non-owned, contended ball when the owner does not update the position fast enough
* As new peers are added into the simulation, how are the balls migrated to improve performance?
* What happens when a ball rolls out of the region of influence? Who is responsible for updating the position of that ball?