Distributed robotics simulator with Unreal Engine

Larry Ng (Rapyuta Robotics)
Yu Okamoto (Rapyuta Robotics)

https://rclue.readthedocs.io
https://rapyutasimulationplugins.readthedocs.io
Rapyuta Robotics is the pioneer in cloud robotics

Select investors

Goldman Sachs

GLP

YASKAWA

SONY
Why use Unreal Engine (UE)

Many tools have already been developed by the Game Industry

- Photorealistic rendering
  - data generation for machine learning
  - demo/marketing
- Game assets are available. [Epic marketplace](https://www.epicgames.com/fortnite/en-US/home).
- Blueprint Editor
- Can get talent from Game Industry
- Physics engine

FF IIV Remake
https://www.jp.square-enix.com/ffvii_remake/about/index.html

Fortnite
rclUE is a ROS 2 client library for UE4

- ROS 2 Nodes are implemented as UE Actors
- Allows for Publishers/Subscribers etc to be defined inside UE4
- Supports: Ubuntu, ROS 2 Foxy, UE 4
- ROS components can be accessed from Blueprint
- no bridge node
Unreal Networking: Replication and RPC

Replication:
Copy State from Server to Clients every tick

RPC:
Call a function in server from the client (and Vice Versa)

Client
1. Player Controller receives Player Input
2. Player Controller moves local Actor

Server
1. Server Side Player Controller applies new state to Server side Actor

Other clients
1. Server Actor State is replicated to all other Clients
Our Use of Unreal Networking

Replication: Duplicate Position of Server’s Robots to all other Clients

RPC: Move Server’s copy of the Robot

Client
1. Robot ROS Node receives ROS Topic
2. Robot moves based on cmd_vel
3. Pose and Velocity sent to Network Controller

Server
1. Server Side Network Controller moves relevant robot to Pose

Other clients
1. Server’s Robot’s Pose are replicated to other clients
Distributed Simulator architecture

Goal: Simulating large number of robots
- Separate Viewer from server
  - GPU is only in the Viewer client, reduces costs significantly
- Separate Robot software from server
  - ROS 2 Node only exist in client
  - Sensor simulation at client
- No delay between Robot app and simulator ROS 2 Node
  - However, delays can occur when syncing a large number of clients to the server
Standalone vs distributed simulation
## Standalone vs Distributed Simulation - 10 robots

- **Standalone Instance specs**

<table>
<thead>
<tr>
<th>Type</th>
<th>CPU (Cores)</th>
<th>RAM (GB)</th>
<th>GPU (Yes/No)</th>
<th>Instances</th>
<th>Cost (¥/month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone</td>
<td>8</td>
<td>56</td>
<td>Yes</td>
<td>1</td>
<td>~$550</td>
</tr>
</tbody>
</table>

- **Distributed Simulation Instance specs (5 robots per Robot Client)**

<table>
<thead>
<tr>
<th>Type</th>
<th>CPU (Cores)</th>
<th>RAM (GB)</th>
<th>GPU (Yes/No)</th>
<th>Instances</th>
<th>Cost (¥/month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>2</td>
<td>4</td>
<td>No</td>
<td>1</td>
<td>~$35</td>
</tr>
<tr>
<td>Robot Client</td>
<td>2*2</td>
<td>4*2</td>
<td>No</td>
<td>1*2</td>
<td>~$35*2</td>
</tr>
<tr>
<td>Viewer Client</td>
<td>2</td>
<td>4</td>
<td>Yes</td>
<td>1</td>
<td>~$200</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td><strong>8</strong></td>
<td><strong>16</strong></td>
<td><strong>Yes</strong></td>
<td><strong>4</strong></td>
<td><strong>~$305</strong></td>
</tr>
</tbody>
</table>
Demo

link: https://roscondemo-kqdbp.ep-r.io/
backup video: https://drive.google.com/file/d/18XCkvZkeP_FZLiUbffVMac_EE5yiEcuy/view?usp=sharing
For Further Information
Check out our documentation!

Or!

Check us out at our booth

https://rclue.readthedocs.io
https://rapyutasimulationplugins.readthedocs.io
Thank you!

We’re Hiring!

hiring-tyo_sig@rapyuta-robotics.com
For Further Information

Check out our documentation at:

https://rclue.readthedocs.io
https://rapyutasimulationplugins.readthedocs.io

Or!

Check us out at our booth
## Standalone vs Distributed Simulation - 10 robots

- **Standalone Instance specs**

<table>
<thead>
<tr>
<th>Type</th>
<th>CPU (Cores)</th>
<th>RAM (GB)</th>
<th>GPU (Yes/No)</th>
<th>Instances</th>
<th>Cost (¥/month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone</td>
<td>8</td>
<td>56</td>
<td>Yes</td>
<td>1</td>
<td>~¥83,000</td>
</tr>
</tbody>
</table>

- **Distributed Simulation Instance specs (5 robots per Robot Client)**

<table>
<thead>
<tr>
<th>Type</th>
<th>CPU (Cores)</th>
<th>RAM (GB)</th>
<th>GPU (Yes/No)</th>
<th>Instances</th>
<th>Cost (¥/month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>2</td>
<td>4</td>
<td>No</td>
<td>1</td>
<td>~¥5,000</td>
</tr>
<tr>
<td>Robot Client</td>
<td>0.4*10</td>
<td>0.8*10</td>
<td>No</td>
<td>1*2</td>
<td>~¥5,000*2</td>
</tr>
<tr>
<td>Viewer Client</td>
<td>2</td>
<td>4</td>
<td>Yes</td>
<td>1</td>
<td>~¥30,000</td>
</tr>
</tbody>
</table>
Unreal Network: Replication and RPCs

1. Robot Client 1 receives velocity input $u$.
2. Locally update state.
3. Robot state and timestamp are sent to Server from Robot Client 1.
5. Server state and timestamp are sent to Robot Client 2.

**Equations**

- **Robot Client 1**
  - Robot 1 (Controlled in this client)
  - $x_{c1}(t_{c1} + dt_{c1}) = x_{c1}(t_{c1}) + u(t_{c1}) dt_{c1}$
  - $v_{c1}(t_{c1} + dt_{c1}) = u(t_{c1})$

- **Server**
  - $x_s(t_s + dt_s) = x_{c1}(t_{c1}) + u(t_{c1}) (dt_s + t_s - t_{c1})$
  - $v_s(t_s + dt_s) = u(t_{c1})$

- **Robot Client 2**
  - Robot 1
  - $x_{c2}(t_{c2}) = x_s(t_s)$
Why use Unreal Engine (UE)

- Photorealistic rendering
  - data generation for machine learning
  - demo/marketing
- Game assets are available. Epic marketplace.
- Physics engine: PhysX*
- Editor:
  - GUI → easy/intuitive to edit environment.
  - Blueprint → Visual scripting. Creating logic from GUI
  - Behaviour Tree → Human/non robot simulation

*UE5 uses Chaos physics engine
*List of Unreal Engine games
More About Pick Assist Robots
Another Thing About Pick Assist Robots
A Bit More About Pick Assist Robots
Also About Pick Assist Robots
One More Thing..
And Finally
Rapyuta Robotics is a **spin-off from ETH Zürich**, which has produced many famous researchers including Einstein.
ラピュタロボティクス株式会社は、アインシュタインをはじめ、数々の著名研究者を輩出しているチューリッヒ工科大学（ETH Zürich）からスピンオフした大学発ベンチャー企業です。
Index

- A
- B
- C
- D
List

- A
- B
- C
- D
A great testimonial about Rapyuta Robotics’ cool and amazing pick assist robots.

By Someone
Thank you!
ロボットを便利で身近に