**Parallel SLAM for Mobile Robots in the Cloud**

Simultaneous localization and mapping (SLAM) is a very important capability for mobile robots and has been studied extensively in the literature. Computing the position of a robot in an unknown environment amidst measurement errors and simultaneously computing a map of the environment can be a computationally challenging task. SLAM algorithms can use various inputs like distance reading to objects coming from a laser scan, images of the environment etc to do the mapping. We have chosen an algorithm called GMapping[[1](#_ENREF_1), [2](#_ENREF_2)]; which uses distance measurements and implemented a parallel version of this popular algorithm. GMapping is a particle filtering based SLAM algorithm. The algorithm maintains a number of particles each containing a probable map of the environment. The robot uses a laser sensor to find the distances to the objects in its path. As the robot moves through the environment it gets new distance reading as well as new position measurements. According to these new readings the algorithm calculates a weight for each particle depending on how probable that particle is given the readings. Then the algorithm draws particles with replacement from this set according to their weights and this step is called resampling. Resampled particles are used with the next reading. At each reading the algorithm takes the map associated with the particle of highest weight as the correct map. There are serial versions of the algorithm implemented in C++ language. These implementations are not suitable to run in our cloud based distributed streaming computing engine and we had to develop Java version of the algorithm. The computation time of the algorithm depends on the number of particles used, size of the environment and the number of points in the distance reading.

**Overall design of the Application:** The SLAM algorithm is independent from any robot. We are using TurtleBot by Willow Garage as our robot which is a differential drive robot equipped with a Microsoft Kinect sensor as a demonstration. TurtleBot has a ROS driver and a supporting software stack; which can be used to retrieve information like odometry, laser scans etc from the robot as well as controlling it. An application connects to the ROS is deployed on a gateway running in a workstation. This applications converts the ROS messages to a format that suits the cloud application and send transformed data to the application running in the FutureGrid openstack based VMs using the message brokering layer. The application uses the laser scans coming from the IR sensor of the Kinect and Odometry readings of the Turtlebot. The application running in the cloud generates a map according to the current information it receives and send this map back to the workstation running the Gateway. The gateway can save this map, publish it back to ROS for viewing etc.

**Main benefit:** Algorithm can run with many particles with less latency improving the accuracy of the algorithm both because of the number of particles and the increased number of readings it can process.

**Things to improve in the future:** At the moment it is still difficult to develop and scale these applications with the modern distributed stream processing engines due to the complex programming required. Another important area to focus on the scalability is how to schedule the tasks in a dynamic environment of robots. As the robots connect and disconnect from the system the application resources must be rescheduled to get the optimum performance out of the system. Also we observed there are fluctuations in the latency of the application due to various reasons like network, virtualization etc. It would be nice to address these in the future.

**References**

1. Grisetti, G., C. Stachniss, and W. Burgard. *Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling*. in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. 2005. IEEE.

2. Grisetti, G., C. Stachniss, and W. Burgard, *Improved techniques for grid mapping with rao-blackwellized particle filters.* Robotics, IEEE Transactions on, 2007. **23**(1): p. 34-46.