CS554 Project Ideas

OS:DistOS – Light-weight Distributed Operating Systems

Overview

Programming paradigms are different as we go from one node (multi-threading/processing) to multi-nodes (workflows, MapReduce, MPI). Legacy applications which run in a single node have to be rewritten to be able to run in a distributed environment across machine boundaries. This project aims at supporting legacy applications in a distributed system environment. As an overview, this project will modify a light-weight operating system (e.g. Minix or LUbuntu) to span multiple nodes. Mosix is a distributed OS, and could be seen as a good example of what you need to implement (unfortunately, Mosix is not open source for you to modify directly). You will likely have to modify core system calls or programs (e.g. fork, pthread, top, virtual memory, CPU scheduler, memory management, etc) to allow them to operate across machine boundaries. You should have taken a rigorous OS class before attempting this project, ideally an OS class where you designed many OS components from scratch. Ideally, you should be able to run micro-benchmarks to showcase the extensions you made. A big plus would be of you can take relatively complex applications (e.g. a web server, or a relational database) and show its functionality and performance.

Relevant Systems and Reading Material

- Minix: http://www.minix3.org/
- Mosix: www.mosix.org/

Preferred/Required Skills

- Programming language choice: C/C++
- Skills/knowledge: OS

Performance Metrics

Legacy support for multi-process and/or multi-threaded applications, support for remote memory, application performance; experiments are expected to be conducted on the Amazon EC2 cloud with up to 16 VM instances

Project Mentor

Ioan Raicu, iraicu@cs.iit.edu