Educational real-time scheduling framework on distributed mobile environments

Mariano Larios Gómez

Benemérita Universidad Autónoma de Puebla

Universidad Autónoma de Tlaxcala

Contact Information:
Department of Computer Science
Meritorious Autonomous University of Puebla
Puebla-México
14 South and San Claudio Avenue

Phone: +5222295500
Email: mariano.larios@correo.buap.mx

VIRTUAL EDITION | GUADALAJARA, MEXICO
WORKSHOPS: OCTOBER 4-5, 2021
CONFERENCE: OCTOBER, 6-8, 2021
TUTORIALS: SEPTEMBER 27- OCTOBER 1 AND OCTOBER 11-15, 2021

Abstract

This work describes the implementation of a user graphical interface, named JPeer, for an embedded software; this shows the use of a P2P network that allows a supercomputer the allocation of its resources optimally among the different nodes connected to it. The peers in this project are represented as mobile devices and with the use of JNI (Java Native interface), with this it is possible to communicate Java and C++ peers created, accordingly, message passing would therefore be possible among different programming languages and operating systems. We applied several P2P nets with multiple peers in a node of LNS (supercomputing laboratory) in Southeast Mexico. The understanding of distributed and real time system algorithms can represent a difficulty due to the abstraction and difficult learning. In the meantime, the framework implementation represents a mobile distributed system environment, where the user can manage the nodes in a simple, easy and transparent way, as well as visualize how each node executes its processes, becomes a very useful and didactic tool.

Introduction

In a mobile distributed system (MDS), the nodes can communicate with neighbors by means of a suitable configuration, which allows us a multicast, in a transmission range from transmitter to receiver within a neighborhood **L**. The formalism communication in this MDS configuration neighborhood describes a dynamic topology, which makes it difficult to establish and maintain a communication path between the nodes $v_i \to v_j$, taking into account that $i \neq j$ and $v_i, v_j \in V$, where **V** is the set of nodes or peers in a mobile network, presumably remote because of the unwillingness to communicate in a given time. Assuming that the solution of using a related cyclic graph, reduces these errors in planning and communication among peers (1).

$$G =< P, R, J, A > \;\; p_i \in P,\; r_k \in R \; and \; j_l \in J \qquad (1)$$

It should be noted that 92% of the peers achieved their deadline established a priori. From 980 peers, a constant with respect to time is observed. The interpretation of this statement indicates that the scheduler has a brief improvement with respect to the deadline of the peer´s processes. After a processing time, 98.9% of processes reaching out their deadline are achieved. The remaining 8% achieved a considerable improvement in the range of 1.8s to 2 seconds, and it keeps up with this behavior.
A metric is also proposed (2), to compare different algorithms based on theorems of process scheduling on uni-processors and multiprocessors, by measuring the computation time required for the determination of a planner that satisfies the partial order and resource constraints.

$$M_c = \frac{1}{n}\sum_{i=1}^{n-1} N_{p_i}(t_c) - N_{p_{(i+1)}}(t_c) \qquad (2)$$

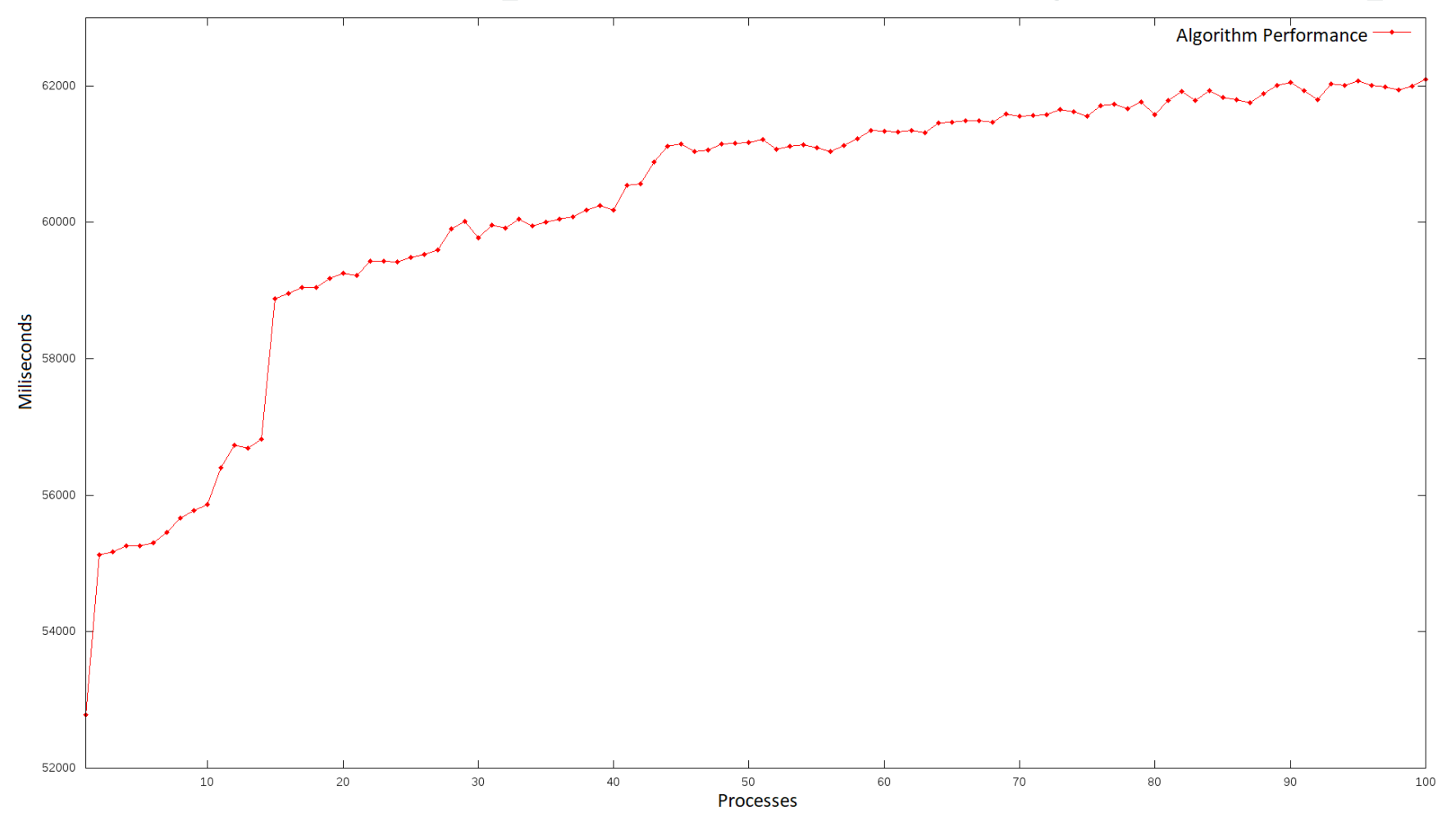
$$t_c = max(f_i) - min(a_i) \qquad (3)$$

Materials and Methods

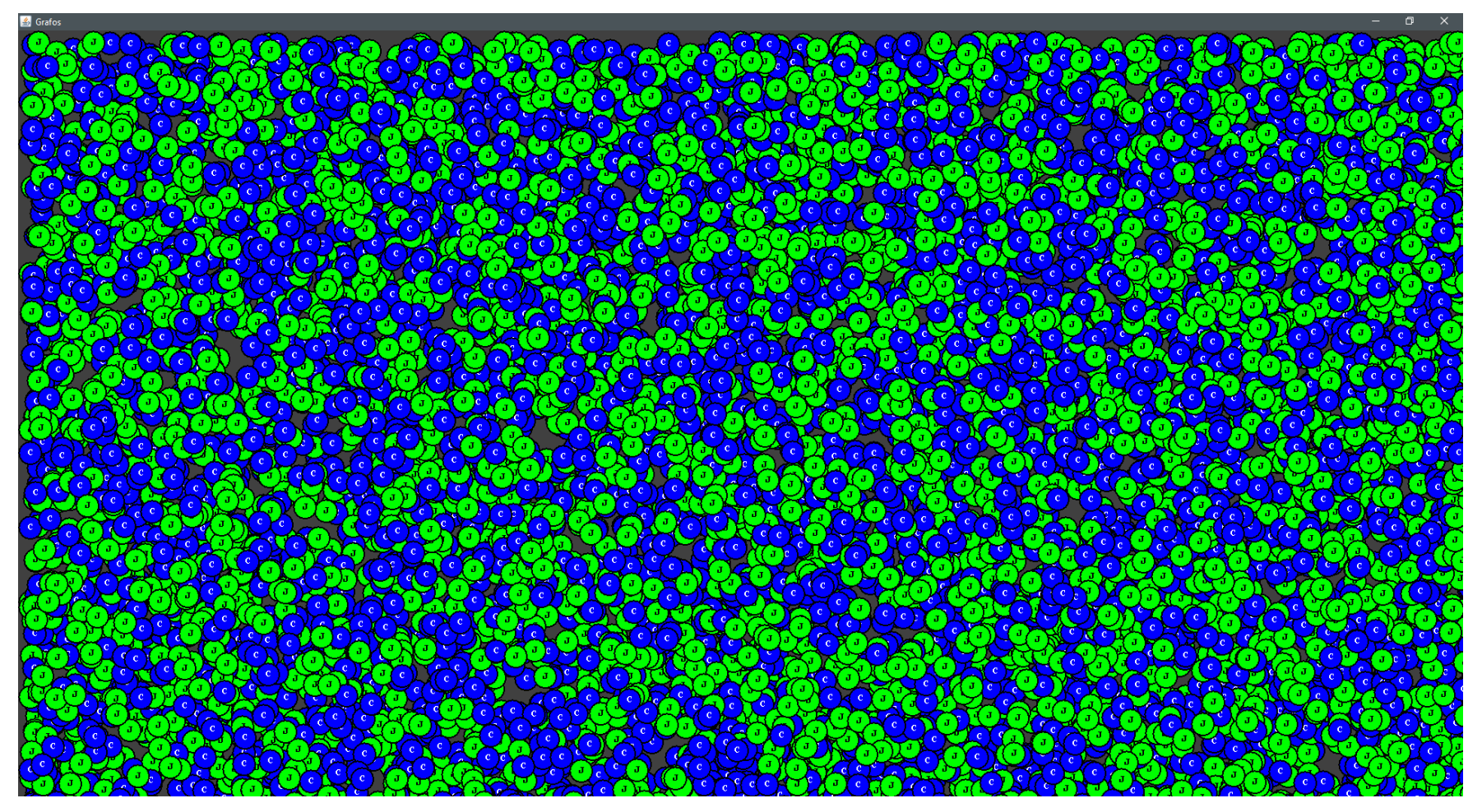
Observe that the total completion time as defined in equation (3) considers the time of the task that took longer to finish and the time of the task that started first.

1 Analysis for the resource and services

The start of the executions of the tasks from 1 to 100 processes. The first processes enter the deadline in good conditions and executions were performed on a node assigned by the National Laboratory of Southeast Mexico Supercomputer (LNS-BUAP). The kernel has basic resources of an operating system, being ideal for the execution of the algorithms of planning. These instances were carried out with a tool for the administration of the cloud of computation, several tests were performed with at least 1,000 instances of peers and a handling of 1,000 processes.



We can see the saturation of peers under tow different programming languages for the evaluation of frameworks with respect to the limits of resources and communication services through messages. The peers in green are communicated with the peers in blue with the help of JNI. This native interface was very useful to obtain the real data that the peers throw in C ++ and be able to communicate with the peers under Java, as well as being visible in a user graphical environment with the JVM.



Listing 1: RTPeerinC.jni

```
#include <jni.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
JNIEXPORT void JNICALL Java_DelayTime
(JNIEnv * env, jobject jobj,jstring i){
 int ID;
 time_t rawtime;
 struct tm *timeinfo;
 time (&rawtime);
 timeinfo=localtime(&rawtime);
 const char *str=(*env)->
 GetStringUTFChars(env, i, 0);
 printf("Dispositivo:\%d Group:
 Processes:\%s Date:\%s\n",
 ID++,str,
 asctime(timeinfo));
}
```

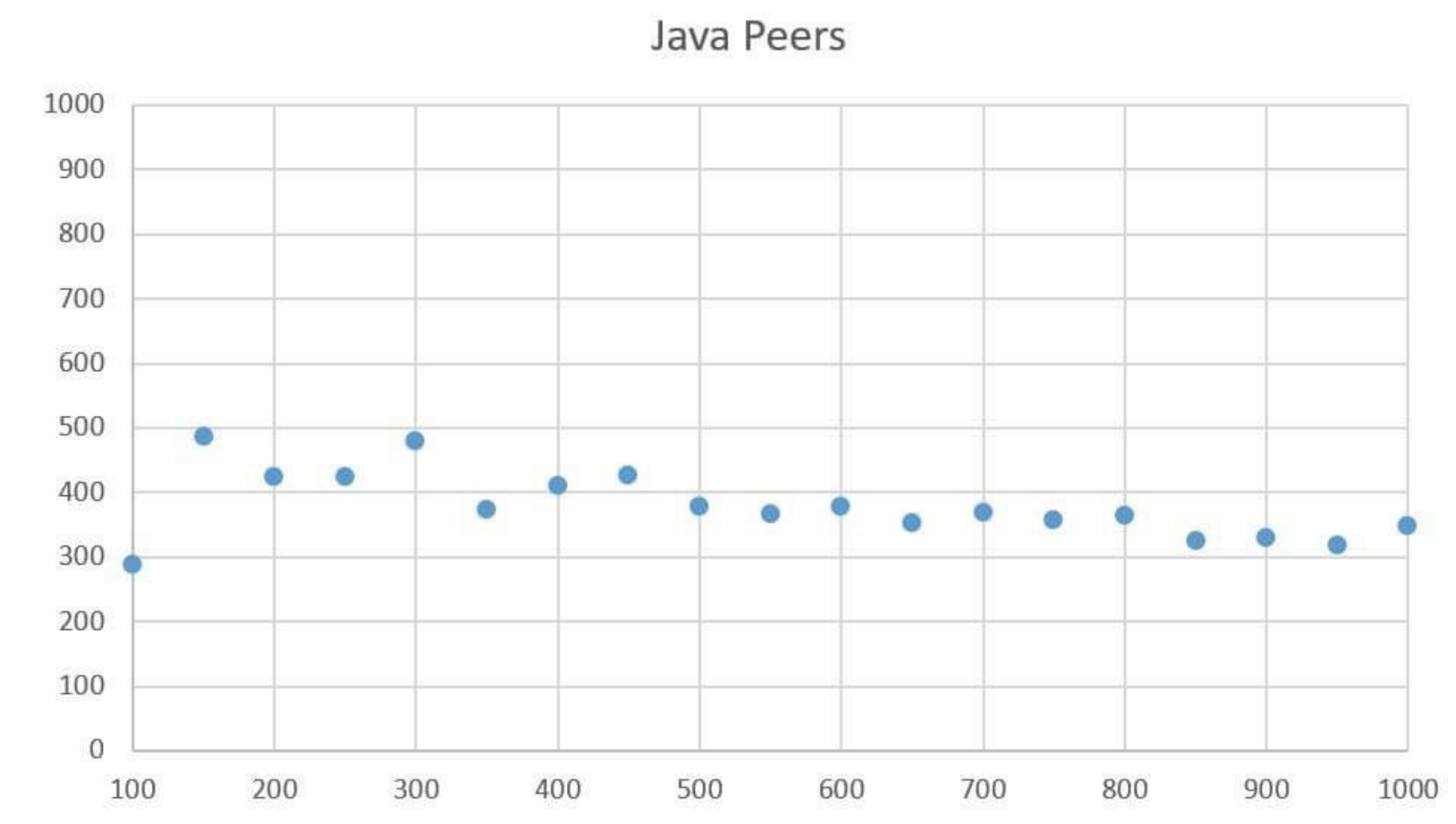
Results

The main functionality of JPeer Framework is the communication between the virtual mobile devices through an information flow between two layers, the graphic layer and the low level layer. For this reason, a multiplatform with JNI and the supercomputing was implemented. In JPeer, the information exchanged between the nodes includes the identifier, the execution time (start and final) and the quantum assigned by the system to the process. The code shown in ((Listing 2)) correspond to the function that requests for time in the java native interface and assign the respective peer to the node.

Listing 2: PeerJavaandC.jni

```
#include<jni.h>
#include<stdio.h>
#include "MainWindow.h"
#include<String.h>
//Implementation of native method
//JNI class
JNIEXPORT jstring JNICALL
      Java_MainWindow_PeerInC
      (JNIEnv *env,jobject thisObj){
  char msg[60]="C++";
  jstring result=(*)NewStringUTF(msg);
      return result;
}
```

For representing the JPeer Framework, a driver capable of using real resources was implemented as a DLL file. When the JNI option is selected in the contextual menu, the driver requests the ID processes of the connected nodes from the supercomputer working on real time. Then a command line terminal is used to verify that the exchange of messages between the connected nodes was correct. Because a personal computer did not have enough resources for testing the creation a lot of mobile devices, there was the need to use a supercomputer, whose characteristics are described next: The Cuetlaxcoapan supercomputer of the LNS [11], is composed of a standard calculation cluster with Intel Xeon processors and a cluster with Intel Xeon Phi Knights Landing processors with 228 Thin calculation nodes (5472 total cores). Each node contains 2 Intel Xeon E5-2680 v3 processors (Haswell) at 2.5 GHz, 12 cores per processor / 24 total cores, 128 GB of DDR4 memory at 2133 MHz, 2 Gigabit Ethernet network interfaces and an InfiniBand FDR 56 Gbps net-work interface. Storage of 1.2 PB of total space for disk storage. Finally, there is a Gigabit Ethernet network for managing the hardware of the supercomputer and the provisioning of software to the nodes.
The implementation of a user interface that allows to represent any paradigm of distributed system was achieved. The networks P2P on between them, and the information in-side each node correspond to the resources requested to the system. The interface is centered in graphs edges and relations between the networks.



We show the balance of the load of the cluster in the time the use of the same time while the number of companions grows, that is to say, with 100 pairs the use of the computer of 288 per second, while with 10000 it is of 347 of 6 to 7 seconds. The start and launch time is approximately 13.4 seconds.

Conclusions

The graphical user interface implemented in this project allows us to visually represent one or more P2P networks in a mobile distributed system, to show an easy and transparent interaction between various nodes. A metric was used to measure the communication time between peers in one neighborhood **L**. The metric is justified by the response time in its deadlines of each process for low-level communication with the native interface JNI. For this reason, it is possible to visualize how a multilanguages P2P network is executed through the exchange of information between the nodes by means of labels inserted in each node. Until now, each node can execute a single process, but it is planned to modify the functions to allow the generation of more peers. Knowing that there are adequate languages for the area of parallelism, concurrency among others, it is also necessary to adapt languages with native characteristics and take advantage of their important characteristics. This project proposes a framework for educational use, as well as scientific and technological use.

Acknowledgements

The authors thankfully acknowledge computer resources, technical advice and support provided by Laboratorio Nacional de Supercómputo del Sureste de México (LNS), a member of the CONACYT national laboratories, with project No. 202103070C