# Analysis of Scheduling Algorithms on a Parallel Programming Context

**Student: Peter Ngugi Nyumu**
**Supervisor: Prof. Afredo Goldman**
Institute of Mathematics and Statistics
University of São Paulo
pnyumu@linux.ime.usp.br

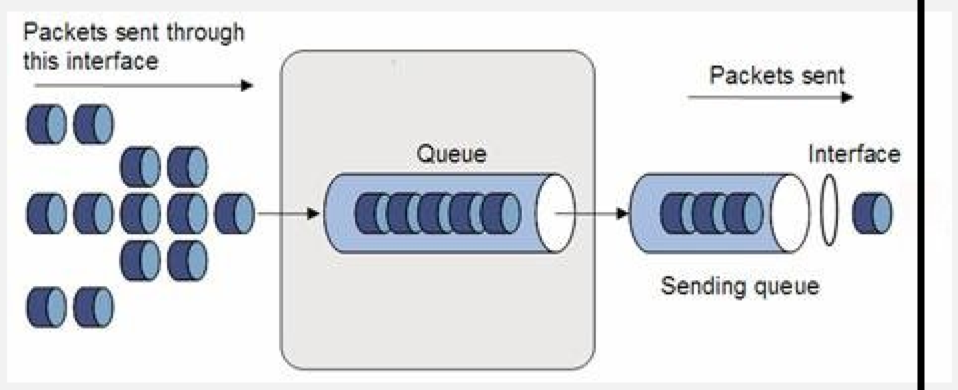
## 1. Introduction

In this working we define diferent kind of computing, and study diferent kind of algorithms, analysing them in the context of parallel computing. The tasks are processed in sequential form.

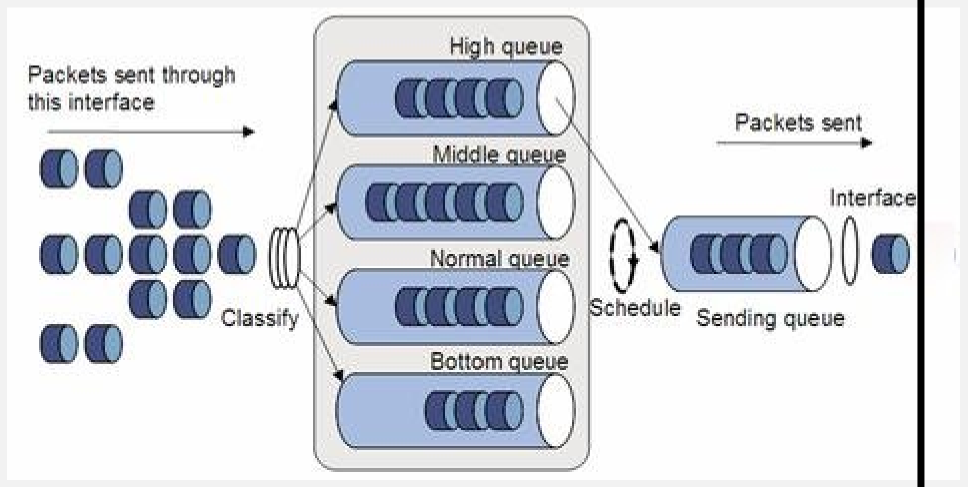
## 2. The Operating System Scheduler

An operating system (OS) is an interface between hardware and user which is responsible for the management and coordination of activities and the sharing of the resources of the computer that acts as a host for computing applications run on the machine. This interface uses diferent types of algorithms.

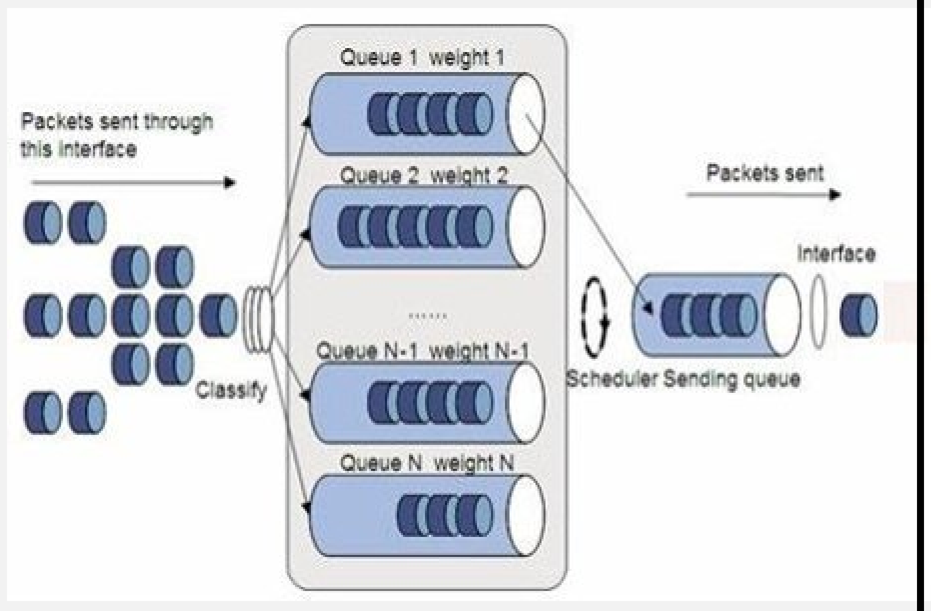
### 2.1 First Come First Served Algorithm



### 2.2 Priority Queue Algorithm



### 2.3 Weighted Fair Queue Algorithm



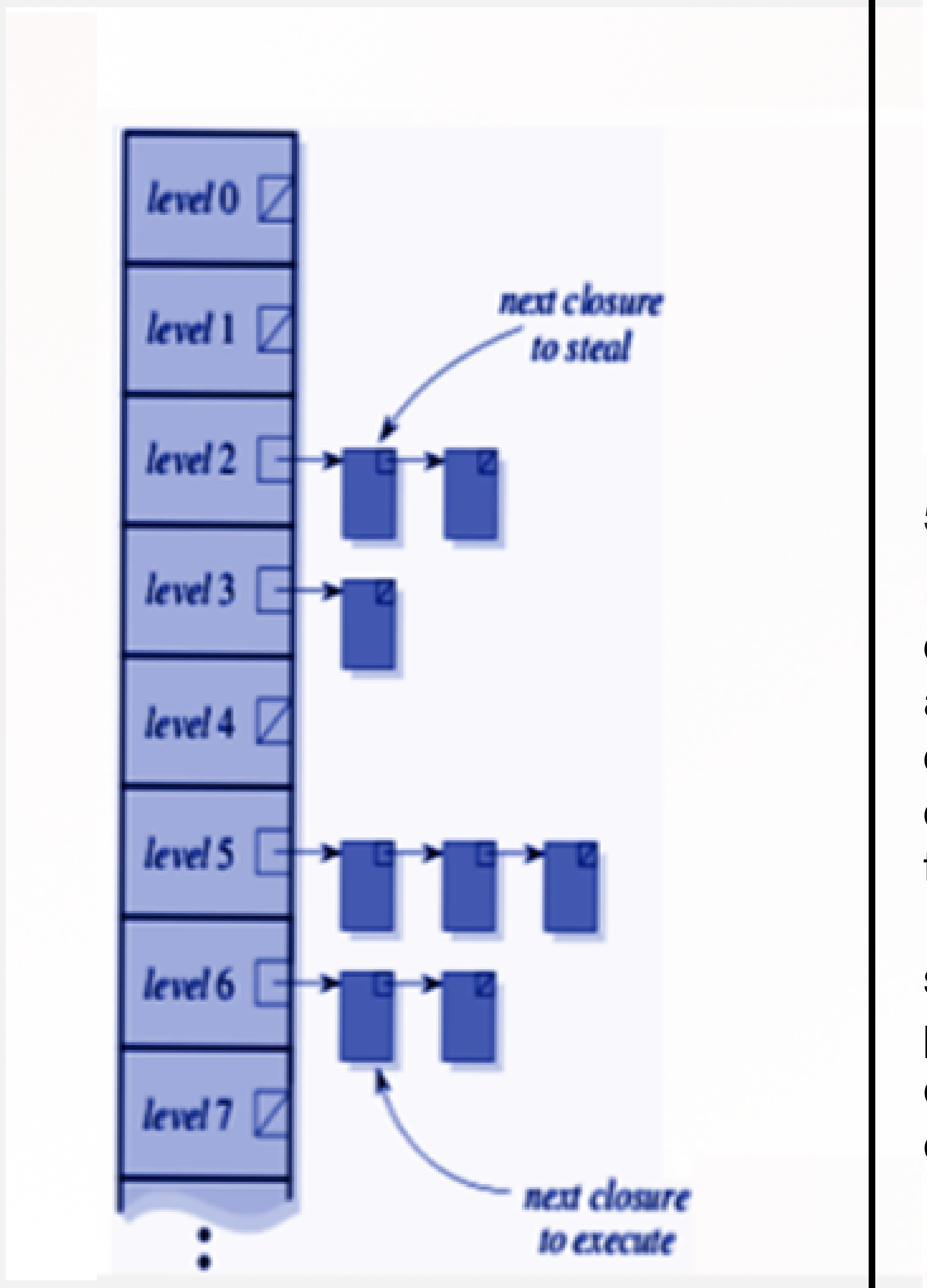
Other algorithms include;

- Round robin
- weighted round robin
- Fair Queueing
- Start-time fair queuing
- Self-clocked fair queing
- Surplus Round robin
- Gang algorithm
- Credit based fair

## 3. Parallel Computing

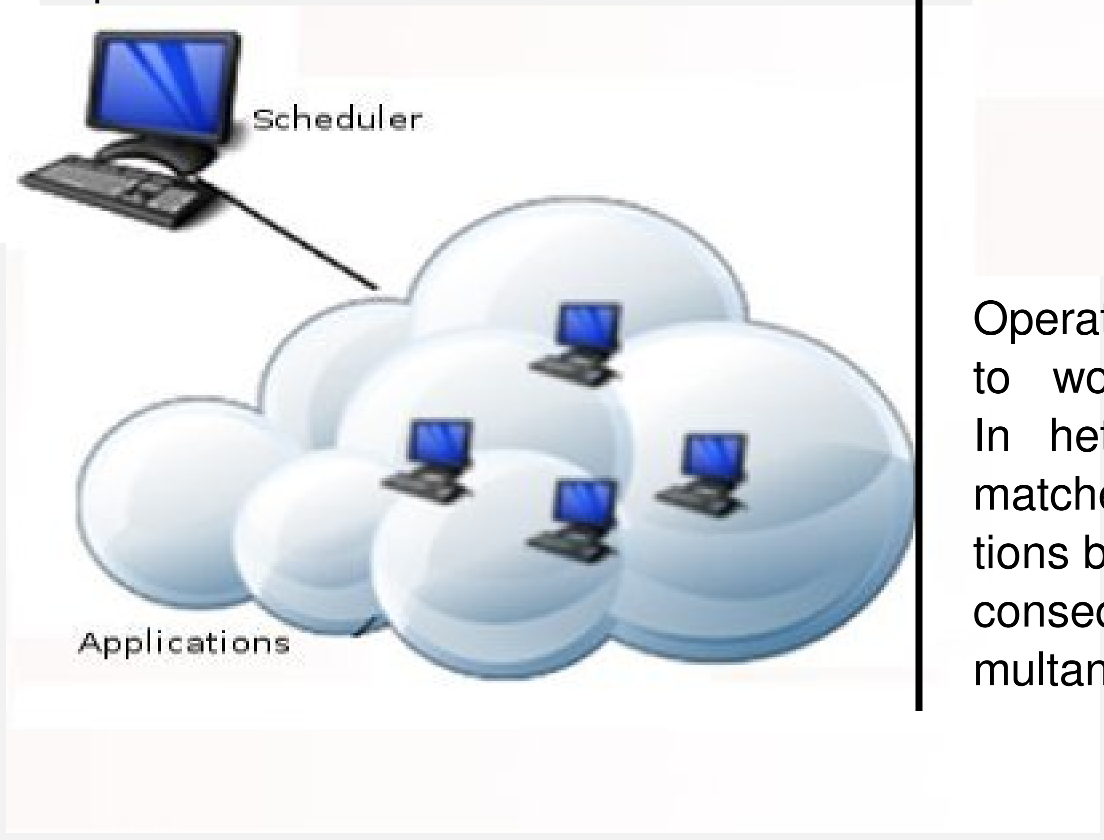
Parallel computing is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel"). Work-stealing algorithm is an algorithm used with Cilk and Kaapi scheduler in Parallel computing.

### 3.1 Work-Stealing Algorithm



## 4. Cluster Computing

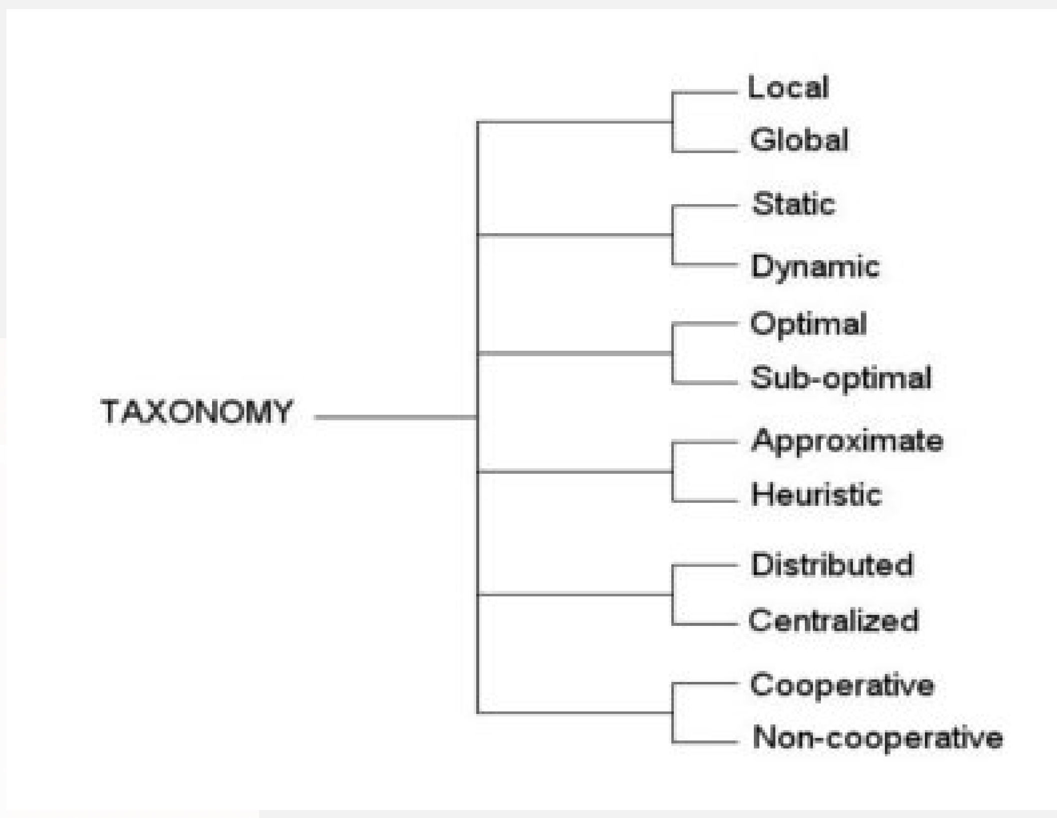
A computer cluster is a group of linked computers, working together closely so that in many respects they form a single computer.



## 5. Grid Computing

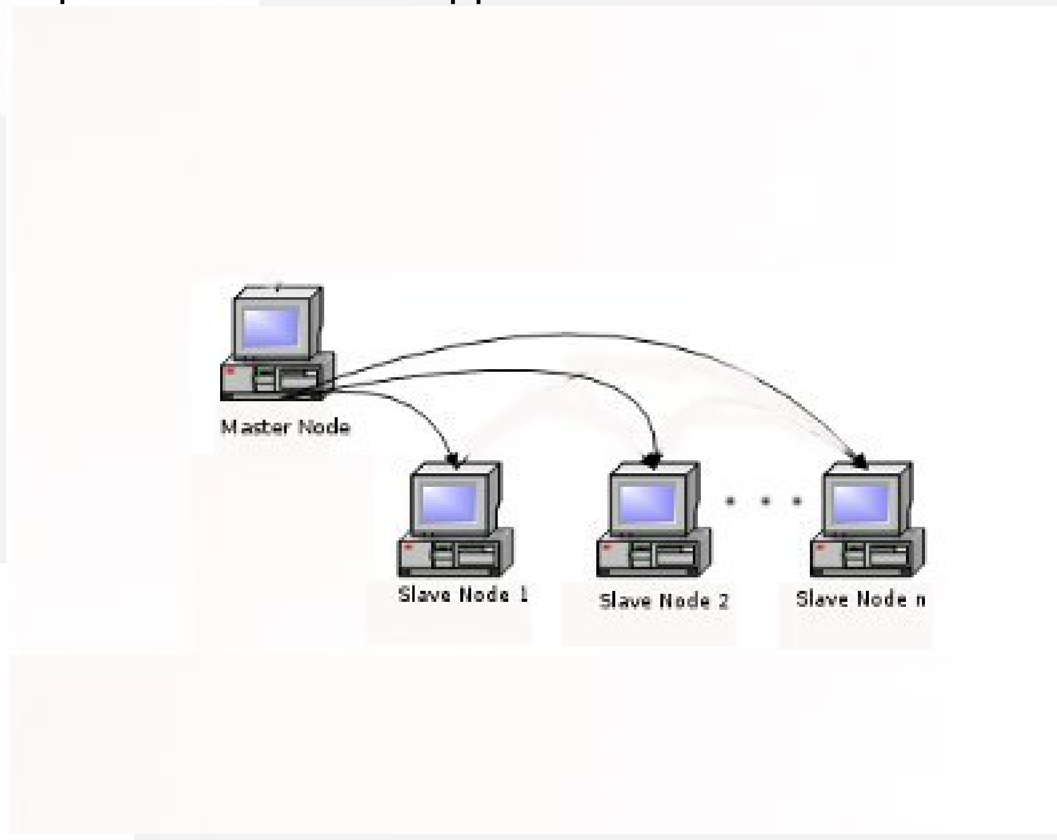
Grid computing is applying the resources of many computers in a network to a single problem at the same time, usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data.

### 5.1 Taxonomy in Grid Computing



### 5.2 Master/Slave grid computing

In such applications, a single master process controls the distribution of work to a set of identically operating slave processes. In evaluating Master-Slave applications, two performance measures of particular interest are speedup and efficiency. In general, the performance of master-slave applications will depend on the temporal characteristics of the tasks as well as on the dynamic allocation and scheduling of processors to the application.



Operating system algorithms are adapted to work in Master-Slave applications. In heterogeneous set, non-preemption matches well with master-slave applications because slaves are independent and consequently don't need to be running simultaneously.