Chapter 3

Software Parallel Processing in Pervasive Computing

Jitesh Dundas
Edencore Technologies, USA

ABTRACT

This chapter proposes the application of periodic wave concepts in management of software Parallel processing projects or processes. This chapter lays special emphasis on Runaway project, which create a lot of problems in Project Management for the stakeholders. This chapter proposes a new and dynamic way to control the software project estimation activity of Runaway project/processes, thereby reducing their future occurrences. This chapter also explains how the equations of unidirectional periodic Waves can be applied in software Parallel processing to measure quality and project execution in a dynamic way at any point in time. The concepts proposed here are dynamic unlike PERT/CPM and other metrics, which fail in worst-case scenarios. The Propagation Speed 'C' at any point in time of a stage or part of a software system executing in Parallel can be given by: C = H/ k (1). Where, H = length of the Wave (i.e. highest point), k = time taken in completing the stage.

INTRODUCTION

As the environment in which we use technology becomes more and more pervasive (Mark .W., 1999), the efficient execution of software has become very important for us. Thus, Parallel processing of software is one step forward in this direction. It is very well known that Parallel processing can help us reduce time of execution.

DOI: 10.4018/978-1-61520-741-1.ch003

Any software function running in Parallel can be divided into stages, with each stage running with varying degrees of freedom, without controlling or affecting the other stages or pieces during execution. We can use equations of periodic Waves to control, monitor and predict the flow of the Parallel execution of software /project. If we arrange these pieces in their order of execution, we will find that they all work in Parallel. Any software that is to be run in Parallel needs to be connected by some link, if being run for the first time. For the first time scenario, the pieces will run one after another, provided the second piece's input is dependent on the first's output. Thus, it will create a staircase like figure.

As the process is repeated again and again, this software will be able to reduce the time-lag in running the pieces in Parallel. This is because the same process is repeated frequently and thus the processing becomes predictable with each execution. Thus, all those pieces or stages of software that are not dependent on the other can be started. As the number of executions increases, the logic of the software for execution becomes more and more familiar and thus it can be executed faster and faster. Thus, the effort for the perfect Parallel execution would be that all the stages are running in Parallel, without any dependency and thus achieving faster execution. However, care must be taken to prevent any loss of quality or Efficiency. In short, the processing cycle expands horizontally and contracts vertically. This is called "The Sandwich Effect".

For e.g.) Assume that there is a software 'S' made up of pieces A,B,C,D,E,F. Thus, the sum of efficiencies of all runs of software pieces or stages should be greater than the Efficiency of the entire software run as a whole.

Thus, we get,

$$\sum St_i > _{\sum} S, \text{ where} \tag{1}$$

i=0 to n

St = Stages of project's SDLC cycle or pieces of software

n = Number of stages or pieces of the software

Parallel software development/processing will be useful and efficient only if the above inequality holds true. This research paper employs the dynamics of Wave functions and their attributes in Parallel software development.

Any two stages are separated by a separation line. This line is important as it shows the end of one stage and the start of another. It also explains the point that any two stages may be related or dependent but they need to be executed independently. The vertical line shows the execution time for the Wave and thus decides its Amplitude. Also, any stage execution starts from the starting point of the separation line for that stage and ends with the end-point for that stage. As time is always positive and moving forward, the Wave will be unidirectional and periodic. The starting point "i" and ending point "j" will give the Wave representation of any stage S by:

$$_{i=0}{}^{j}\int S_{ij},$$

where i moves from 0 to j

The imaginary line that separates two consecutive stages is always constant. Thus, the vertical line (that depicts the execution time) decides the height or Amplitude of the periodic Wave. In order to include the imaginary separation line that separates two stages, we need to create a Wave like representation.

BACKGROUND

Here, we can use the Wave sine equation (Raymond D, 2007) here to find the displacement of the Wave at any point in time as:-

$$y(t) = A*\sin(\omega t + \Theta) \tag{2}$$

Which describes a Wavelike function of time (*t*) with:

1) peak deviation from center = *A* (aka *Amplitude*)
2) angular frequency ω, (radians per second)
3) phase = *θ*

When the phase is non-zero, the entire Waveform appears to be shifted in time by the amount

Related Content