

Master Thesis Defense
Department of Statistics and Informatics
Applied Informatics Post-Graduation Program



Do you care about Big Data ?
Do you think Exabyte (10^{18}) is Big ?

If you answered YES !!!

Then come to see us kidding with crying Exabyte babies !
Come to see us eating 10^{39457} universes !
And see how to play with even more !

Come to see our ultimate

Distributed van Emde Boas tree

Master Student:	Edgard Lima	Board:	
Advisor:	Ph.D. Guilherme Vilar		Ph.D. Fernando Aires
Co-Advisor:	Ph.D. Tiago Ferreira		Ph.D. Jones Albuquerque
			Ph.D. Kátia Guimarães (CIn / UFPE)

Wednesday, 02 Aug 2017

13:00h GMT-3

UFRPE - PPGIA Auditorium

Abstract

A very important computational problem is how to organize information. In particular, the contemporaneous world has been presented with a new class of problem, to handle a very larger amount of data, called Big Data Problem. Typical data structures has $O(\lg n)$ time cost, where n is the size of the database and \lg is the binary logarithm (\log_2). However, if n is a very large number, like a googol (10^{100}) or a googolplex ($10^{10^{100}}$), data structures of $O(\lg n)$ still have a hard cost to solve a problem. To address this problem, a data structure named van Emde Boas Tree (vEBt) could be used. A vEBt has $O(\lg \lg U)$ worst case time cost (where U is the data universe size), but this low cost demands a lot of memory. The size of memory to implements a typical vEBt is so big that there is no any today's machine that could just instantiate an empty vEBt of 2^{128} universe size. This research proposes a strategy to implement a class of distributed van Emde Boas tree ables to work with huge data mass (big data). The time cost still is $O(\lg \lg U)$ and a computer cluster can be used to run this distributed vEBt, where each cluster's node needs to have very little memory. As we show on experiments, with our solution, now even cheap 4 GB PC machines can handle up to vEB($2^{2^{17}} = 2^{131,072} \approx 10^{39,457}$) trees, which is much bigger than a googol (10^{100}).