
Contents

1. Introduction	1
1.1. Notation and tools	3
1.2. Chord and Distance Halving	4
1.3. Half-life	6
1.4. Our contribution	7
1.5. Bibliographic notes	8
I. Load Balancing in Distributed Hash Tables	9
2. Introduction to Load Balancing	11
2.1. Our results and organization of the first part	12
2.2. Other work on load balancing	14
2.3. Uniform distribution of nodes on the ring	15
3. Approximating the Current Number of Nodes	19
3.1. The 1-marker scheme	19
3.2. The $\log n$ -markers scheme	23
4. Self Stabilizing Load Balancing	25
4.1. The load balancing algorithm <code>ALGSTAT</code>	25
4.2. The analysis of <code>ALGSTAT</code>	28
4.3. Experimental evaluation	31
5. Bounding The Communication Cost	35
5.1. The load balancing algorithm <code>ALGDYN</code>	35
5.2. The performance and dynamics model	38
5.3. Performance and communication cost of <code>ALGDYN</code>	39

II. Transparent Data Structures	47
6. Introduction to Transparent Data Structures	49
6.1. Our contribution and organization of the second part	50
6.2. Linear addressable memory model in a distributed environment	51
6.3. Pointer model in a distributed environment	52
6.4. Transparent memory models	53
6.5. Other work on search structures	54
7. Hypertree Memory Model	55
7.1. A family of graphs for the hypertree model	56
7.2. Dynamic graphs for the hypertree model	57
7.3. Joining and leaving the dynamic de Bruijn cube	58
7.4. Robustness	59
8. Search Tree	61
8.1. Amortized analysis	64
8.2. Recovery	66
8.3. Local balancing procedure	71
9. Summary and Outlook	77
9.1. Load balancing	77
9.2. Data structures	78
Bibliography	85