

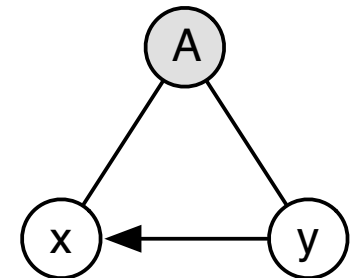
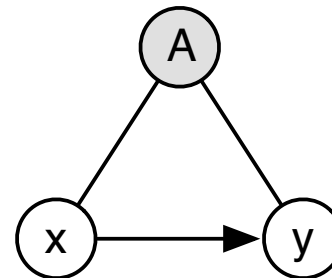
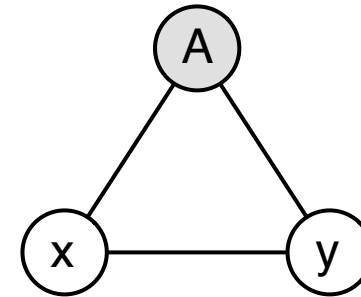
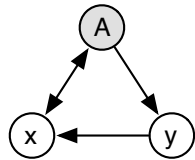
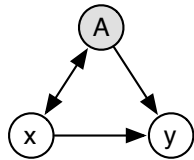
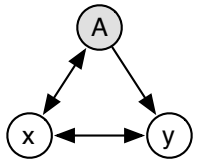
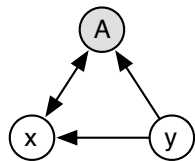
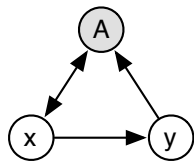
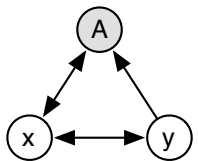
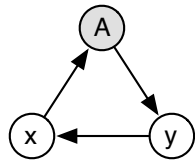
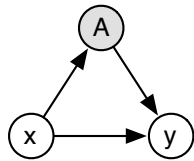
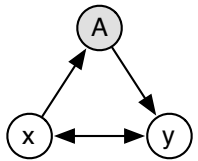
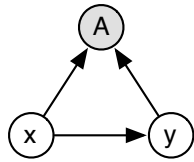
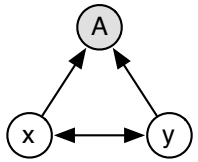
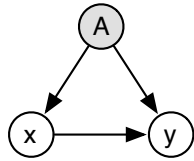
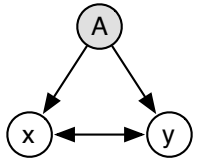
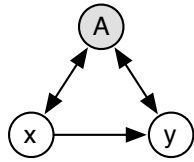
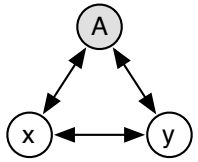


Efficient Computation of the Clustering Coefficient

*Benjamin Schiller, P2P Networks, TU Darmstadt
Research Meeting / 05.12.12*

BSc Johannes Decher

Triangles



Clustering Coefficient



- Neighborhood $N(v)$ = bidirectional connections
- $A \rightarrow B \Rightarrow 1$ triangle
- $A \leftarrow B \Rightarrow 1$ triangle
- $A \leftrightarrow B \Rightarrow 2$ triangles
- Transitivity
- Local Clustering Coefficient
- Average Clustering Coefficient

Algorithms (LCC @ V)

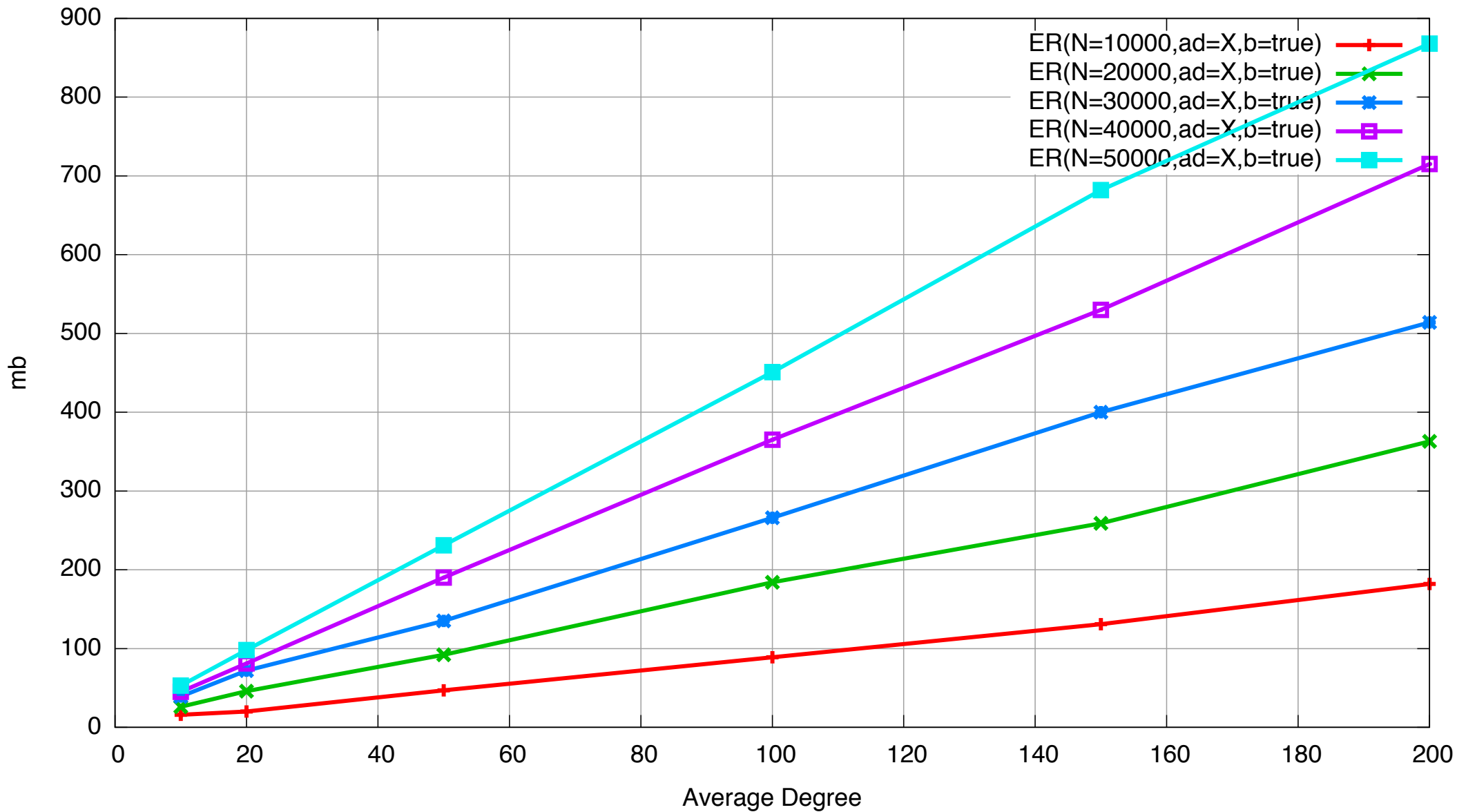


- Alg - V1 a/b
 - for a in $N(v)$
 - for b in $N(v) \setminus \{a\}$
 - is b in $\text{in/out}(a)$
 - OR is a in $\text{in/out}(b)$
- Alg - V2 a/b
 - for a in $N(v)$
 - for b in $N(a)$
 - is b in $\text{in/out}(v)$
 - OR is v in $\text{in/out}(b)$

Memory Requirements for Graphs (1/2)



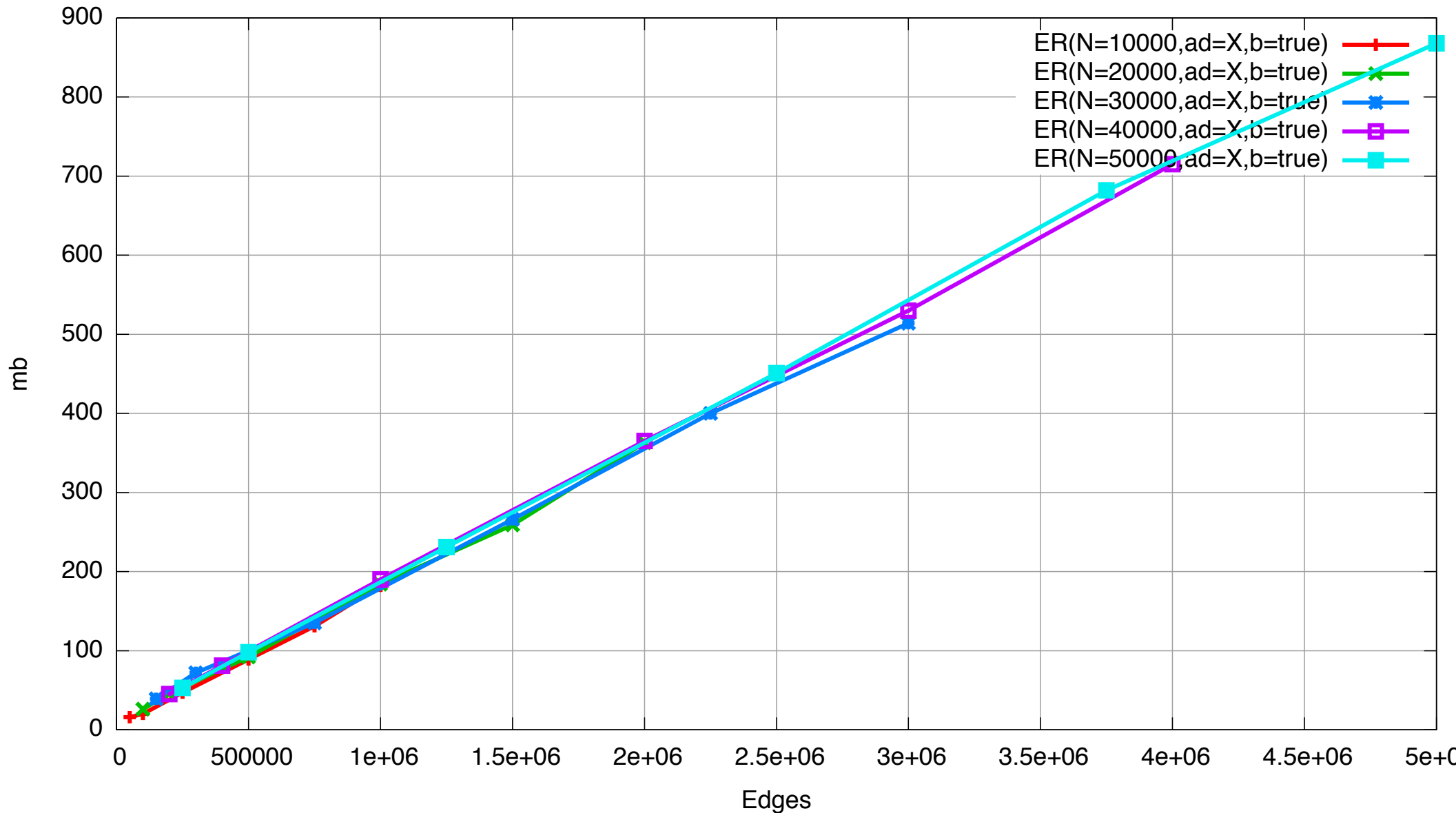
Used Memory



Memory Requirements for Graphs (2/2)



Used Memory



Idea



- Read node information from HDD on demand
- Too slow => in-memory cache
- Bad usage => processing order / graph traversal

0-out	0-in
1	1
3	4
16	16
23	35
25	66
126	167
...	...

Our Approache(s)



- Caching strategies (limited by size / # of nodes)
 - LRU
 - FIFO
 - LFU
 - Weighted

- Graph traversal strategies
 - BFS (d/r)
 - DFS (d/r)
 - BFS-Bubble (d/r)
 - Sorted (s)
 - Random (r)

Neighbor selection:

(s) = degree desc

(r) = random

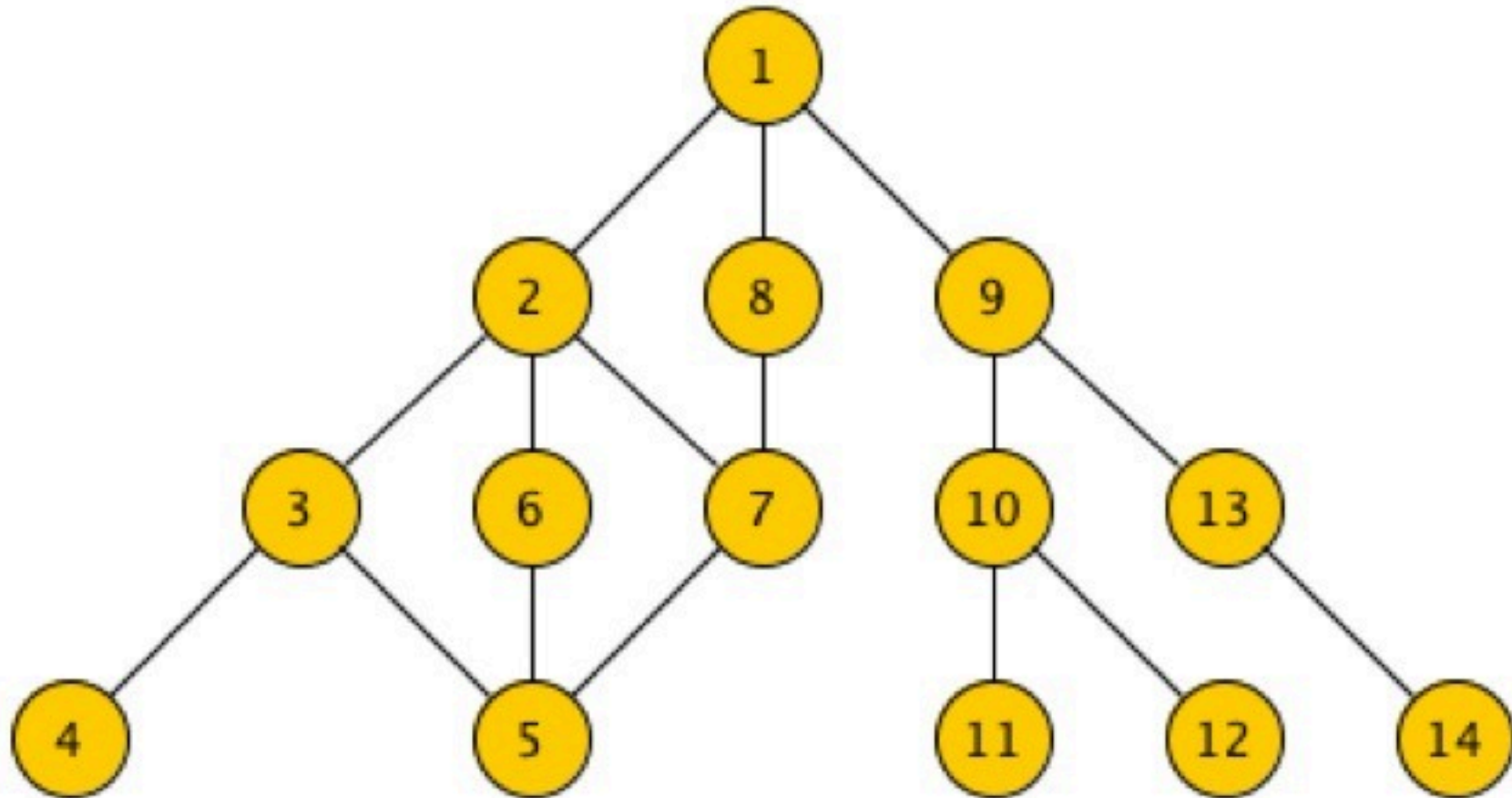


Abbildung 4: DFS-Traversierung

BFS-Bubble





- Cache requests
- Cache hits
- Cache misses
- Cache hit size (kB)
- Cache miss size (kB)

- => sum & every 1k requests

- also: current avgCC / current transitivity



10

Nodes: 165.801
Edges: 19.363.678

90

Nodes: 402.702
Edges: 33.259.850

136

Nodes: 403.474
Edges: 62.140.128

180

Nodes: 321.797
Edges: 78.598.862

201

Nodes: 285.941
Edges: 94.461.312

Cache hits (512 MB)



(d)

	BFS	DFS	BFS-Bubble	Sorted	Random
LRU-Cache	57,85 %	77,88 %	78,82 %	50,03 %	35,59 %
FIFO-Cache	54,00 %	76,31 %	77,23 %	46,75 %	33,25 %
LFU-Cache	36,43 %	40,33 %	37,93 %	38,90 %	38,33 %
Weighted-Cache	24,47 %	25,63 %	26,37 %	25,43 %	26,47 %

(r)

	BFS	DFS	BFS-Bubble
LRU-Cache	42,66 %	75,96 %	77,04 %
FIFO-Cache	39,87 %	74,99 %	76,04 %
LFU-Cache	35,57 %	43,05 %	39,94 %
Weighted-Cache	26,42 %	26,21 %	26,66 %

nodes in cache (Random / LRU) (512 MB)

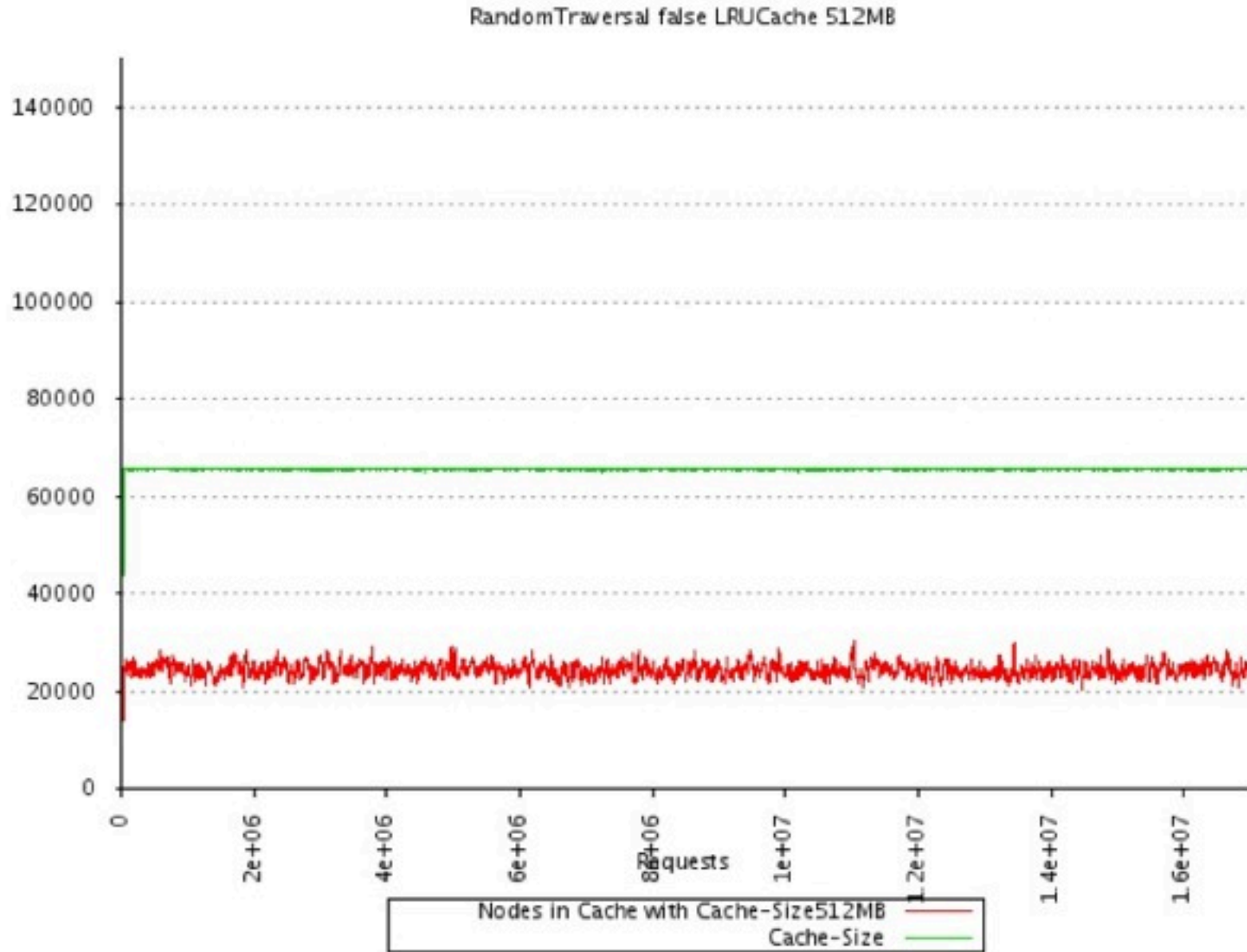


Figure 2: Cacheentwicklung während Random mit LRU

nodes in cache (BFS-Bubble / LRU) (512 MB)

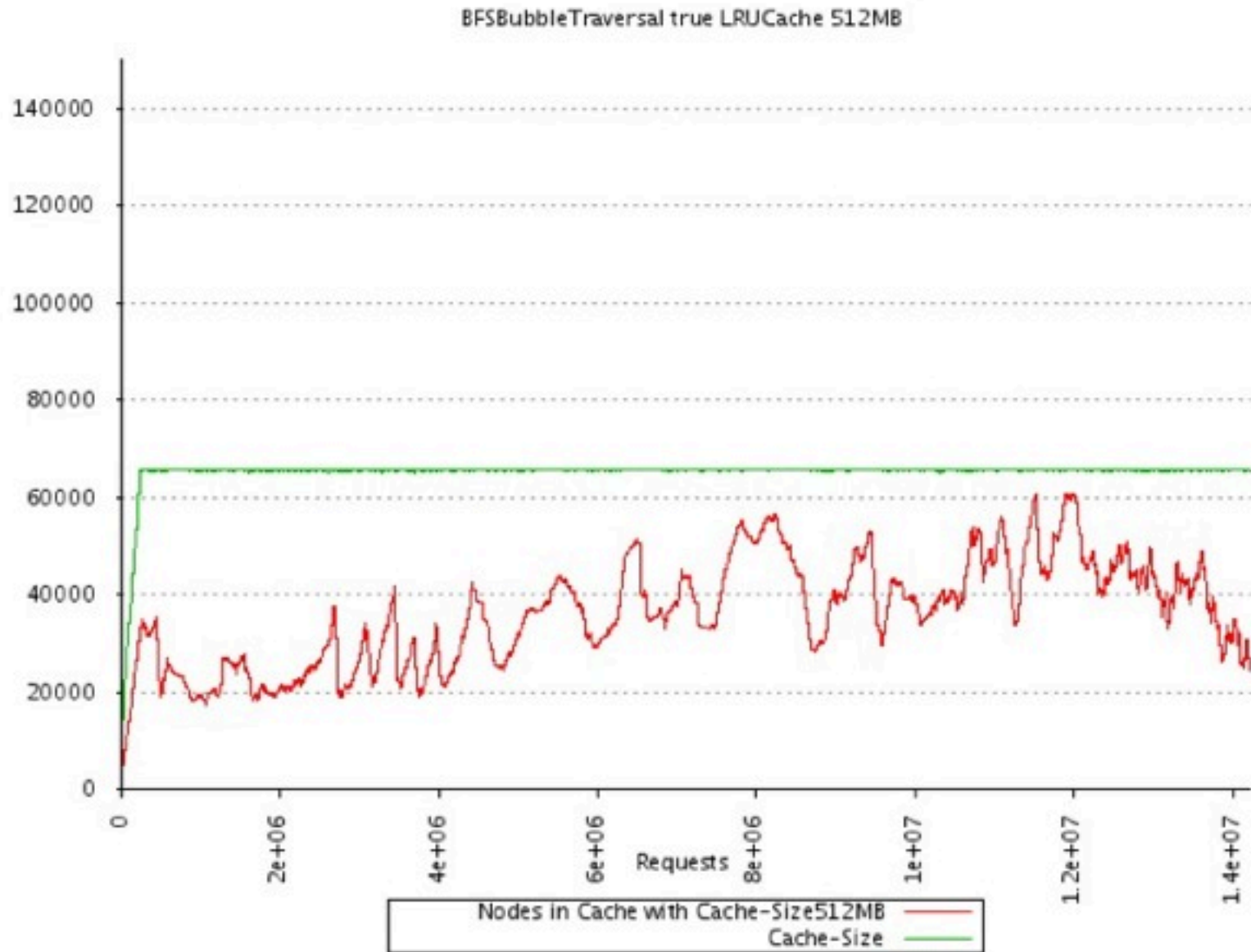


Figure 1: Cacheentwicklung während BFS-Bubble mit LRU

cache hits (4k nodes)



(d)

	BFS	DFS	BFS-Bubble	Sorted	Random
LRU-Cache	20,28 %	59,30 %	60,27 %	18,15 %	7,59 %
FIFO-Cache	18,72 %	57,30 %	58,23 %	17,41 %	7,21 %
LFU-Cache	7,84 %	8,45 %	7,83 %	5,56%	8,47 %
Weighted-Cache	14,01 %	14,10 %	12,85 %	14,14 %	15,78 %

	BFS	DFS	BFS-Bubble	Sorted	Random
LRU-Cache	57,85 %	77,88 %	78,82 %	50,03 %	35,59 %
FIFO-Cache	54,00 %	76,31 %	77,23 %	46,75 %	33,25 %
LFU-Cache	36,43 %	40,33 %	37,93 %	38,90 %	38,33 %
Weighted-Cache	24,47 %	25,63 %	26,37 %	25,43 %	26,47 %

cache size (BFS-Bubble / LRU) (4k nodes)

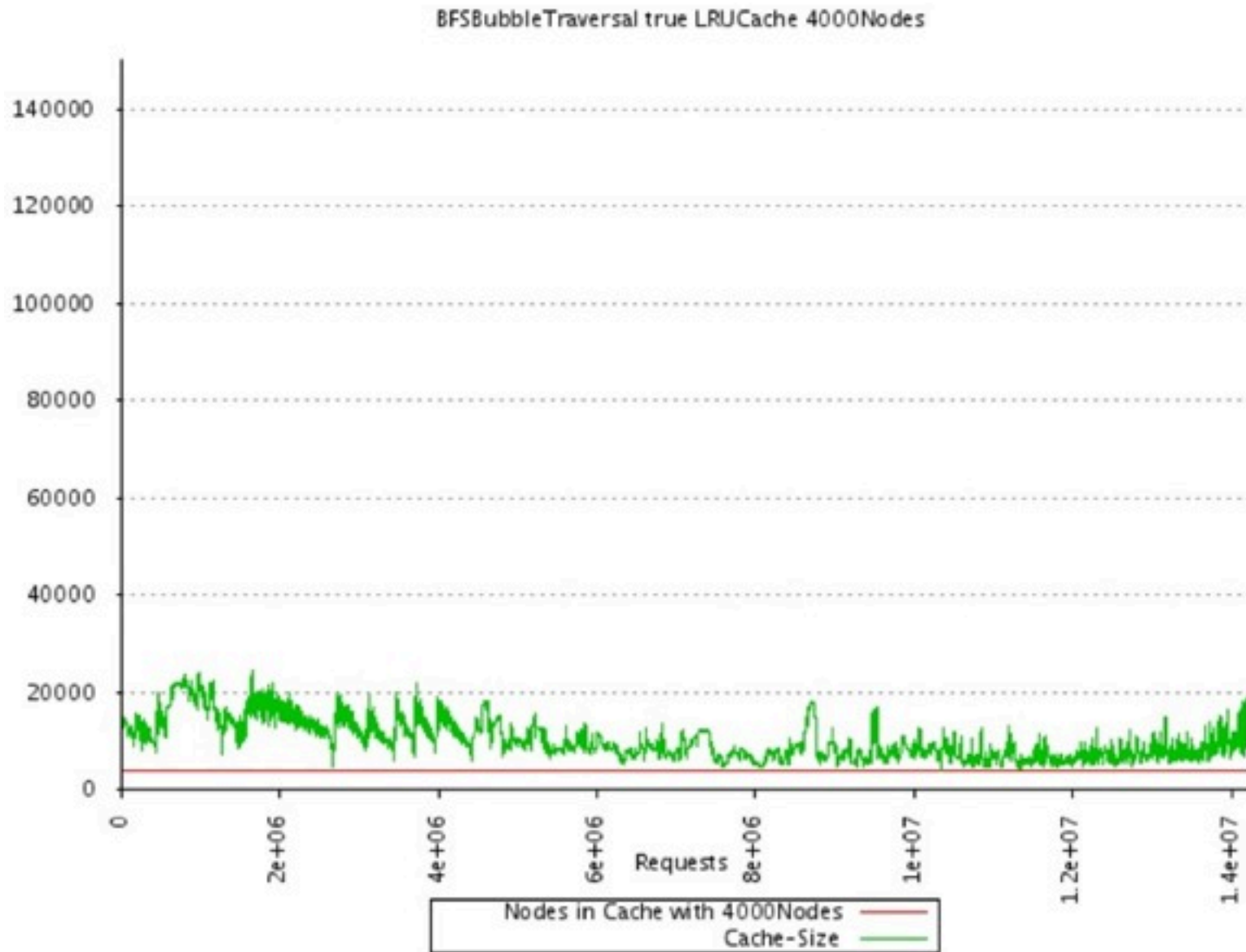


Figure 3: Cacheentwicklung während BFS-Bubble mit LRU 4000 Nodes

cache size (Random / LRU) (4k nodes)

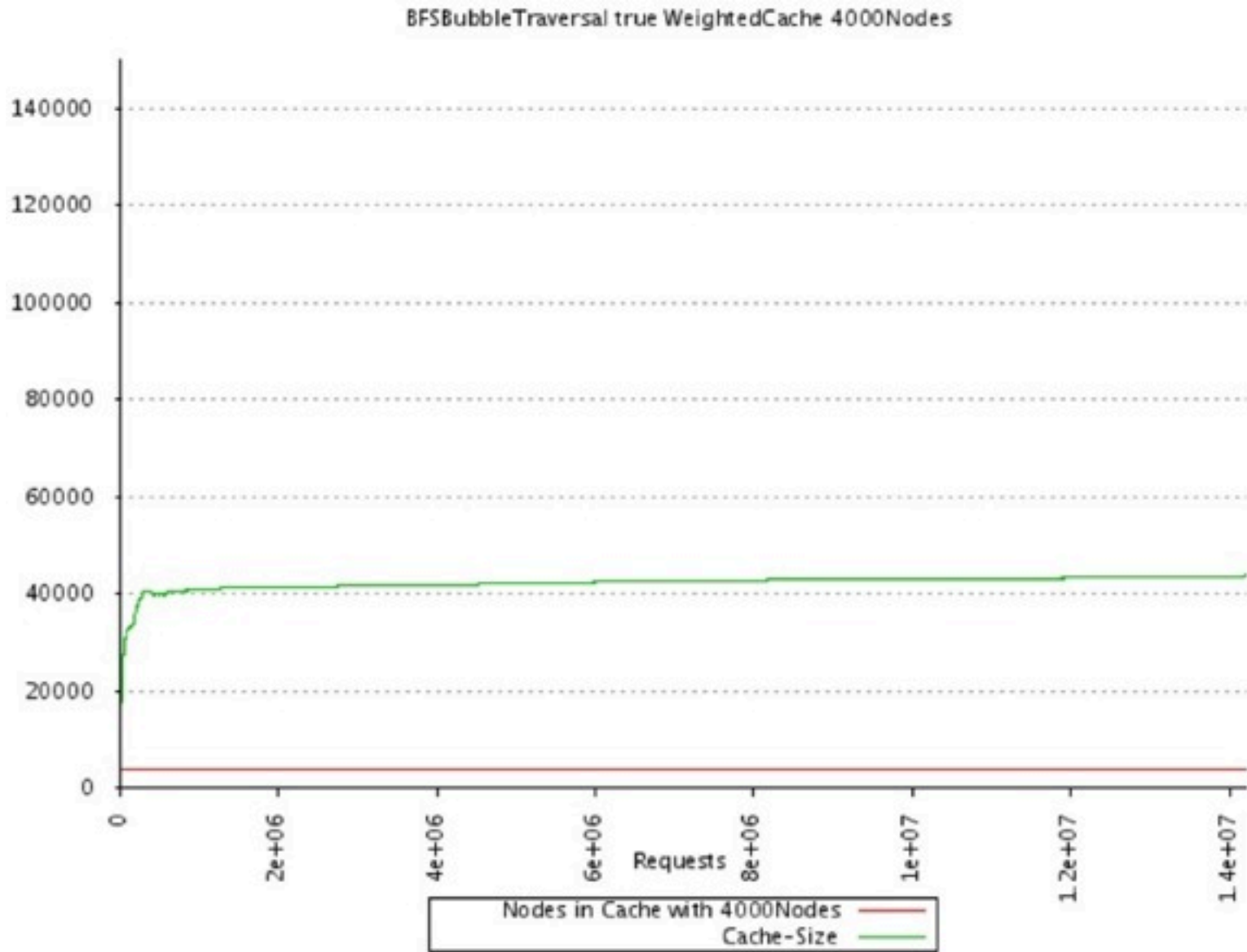


Figure 4: Cacheentwicklung während BFS-Bubble mit Weighted 4000 Nodes

Outlook



- Other graphs
- Implementation in C
- Investigate all measured metrics
- Other caching strategies?
- Other selection strategies?