Hasso Plattner Institut

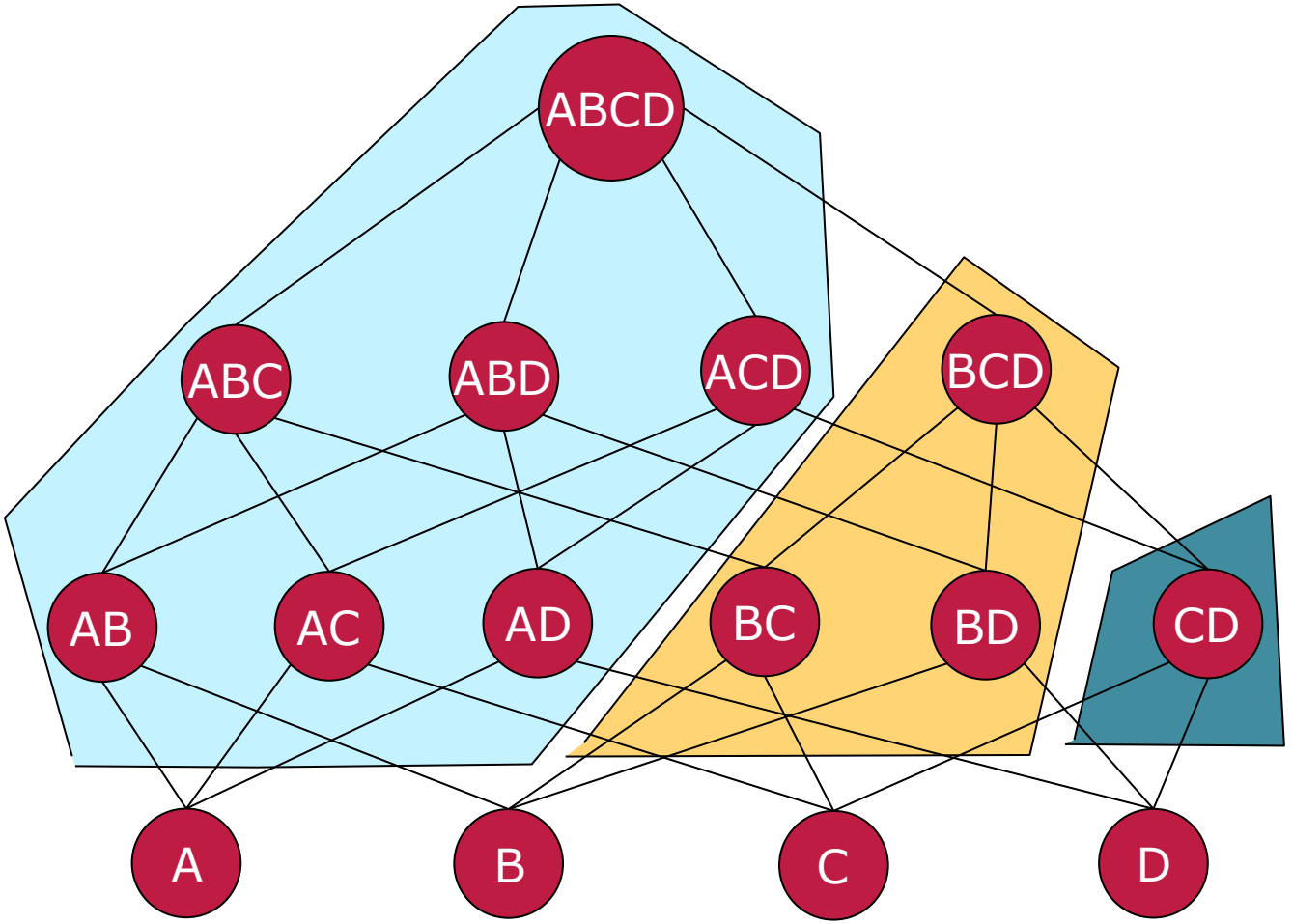
IT Systems Engineering | Universität Potsdam

Eclat Deep Dive

Uwe Hartmann, Peter Retzlaff

10.07.2012

Eclat - Reminder



Motivation - Hot Spots

3

Functions Doing Most Individual Work

Name	Exclusive Samples %
Microsoft.FSharp.Collections.FSharpSet`1[System.Int64].Intersection	76,51
Microsoft.FSharp.Collections.SetTreeModule.ofSeq	12,38
System.Linq.Enumerable.Count	7,18
System.Linq.Parallel.UnaryQueryOperator`2+UnaryQueryOperatorResults+ChildResultsRecipient[System.__C...	0,94
Microsoft.FSharp.Collections.SetTreeModule.countAux	0,85

Eclat vs. MaxClique

	Partition	Search
Eclat	Prefix-Based	Bottom-Up
MaxClique	Maximal-Clique-Based	Hybrid (mix between top-down and bottom-up search)

Example

TID	Items
1	bread, beer, diaper, cheese, pizza
2	bread, beer, diaper, cheese
3	bread, beer, diaper, pizza
4	bread, milk
5	bread, beer, tea
6	bread, milk, flour
7	cheese, tea
8	pizza, tea, toothpaste, cheese
9	bread, milk, tea, flour

Reminder: Eclat - Vertical Tid-List Format

Item	TIDs
Beer	1,2,3,5
Bread	1,2,3,4,5,6,9
Cheese	1,2,7,8
Diaper	1,2,3,6,9
Flour	4,6,9
Milk	4,6,9
Pizza	1,3,8
Tea	4,5,7,8,9
Toothpaste	8

Find Frequent One-Itemsets

Item	TIDs
Beer	1,2,3,5
Bread	1,2,3,4,5,6,9
Cheese	1,2,7,8
Diaper	1,2,3,6,9
Flour	4,6,9
Milk	4,6,9
Pizza	1,3,8
Tea	4,5,7,8,9
~~Toothpaste~~	~~8~~

Minimum Support Count = 2

Find Frequent Two-Itemsets

Itemset	TIDs	Itemset	TIDs
Bread, Beer	1,2,3	Beer, Pizza	1,3
Bread, Cheese	1,2,7	Cheese, Diaper	1,2
Bread, Diaper	1,2,3,6,9	Cheese, Pizza	1,8
Bread, Flour	4,6,9	Diaper, Flour	6,9
Bread, Milk	4,6,9	Diaper, Milk	6,9
Bread, Pizza	1,3	Diaper, Pizza	1,3
Bread, Tea	4,5,9	Flour, Milk	4,6,9
Beer, Diaper	1,2,3	Flour, Tea	4,9
Beer, Cheese	1,2	Milk, Tea	4,9

Extension 1: Partition into Cliques

- Idea: Interpret items as nodes of a graph
- two nodes are connected, if items appear in same frequent set
- maximal cliques in graph == maximal potential frequent itemsets

Generally: NP-Complete

- possible to list all maximal cliques in polynomial time
- necessary condition: sparse graph
- Sparse graph → number of edges linear to number of vertices

Extension 1: Partition into Cliques

Prerequisite: Eclat's Prefix-based equivalence classes

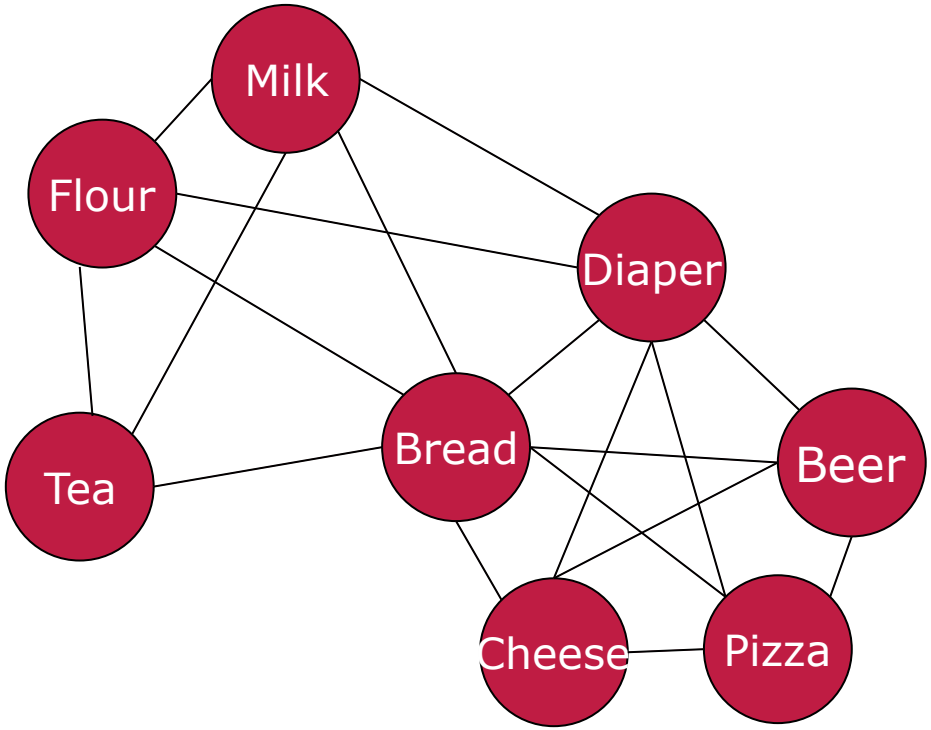
x	[x]
Bread	{Bread, Beer},{Bread, Cheese},{Bread, Diaper},{Bread, Flour}, {Bread, Milk},{Bread, Pizza},{Bread, Tea}
Beer	{Beer, Diaper},{Beer, Cheese},{Beer, Pizza}
Cheese	{Cheese, Diaper}, {Cheese, Pizza}
Diaper	{Diaper, Flour},{Diaper, Milk},{Diaper, Pizza}
Flour	{Flour, Milk}, {Flour, Tea}
Milk	{Milk, Tea}

Partition Prefix Equivalence Classes into Cliques

Prerequisite: Eclat's Prefix-based equivalence classes

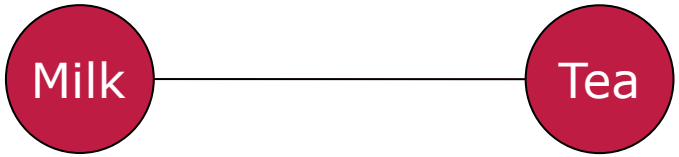
x	[x]
Bread	Beer, Cheese, Diaper, Flour, Milk, Pizza, Tea
Beer	Diaper, Cheese, Pizza
Cheese	Diaper, Pizza
Diaper	Flour, Milk, Pizza
Flour	Milk, Tea
Milk	Tea

Association Graph



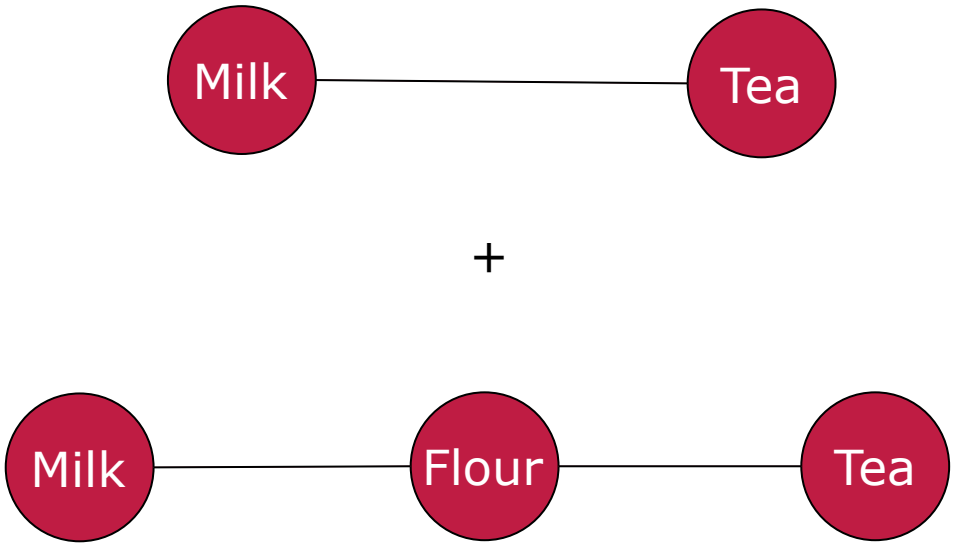
Partition Prefix Equivalence Classes into Cliques

x	[x]
Milk	Tea



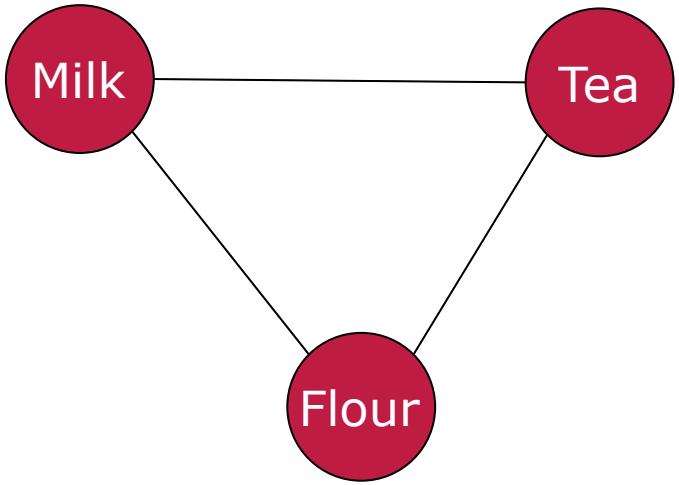
Partition Prefix Equivalence Classes into Cliques

x	[x]
Flour	Milk, Tea
Milk	Tea



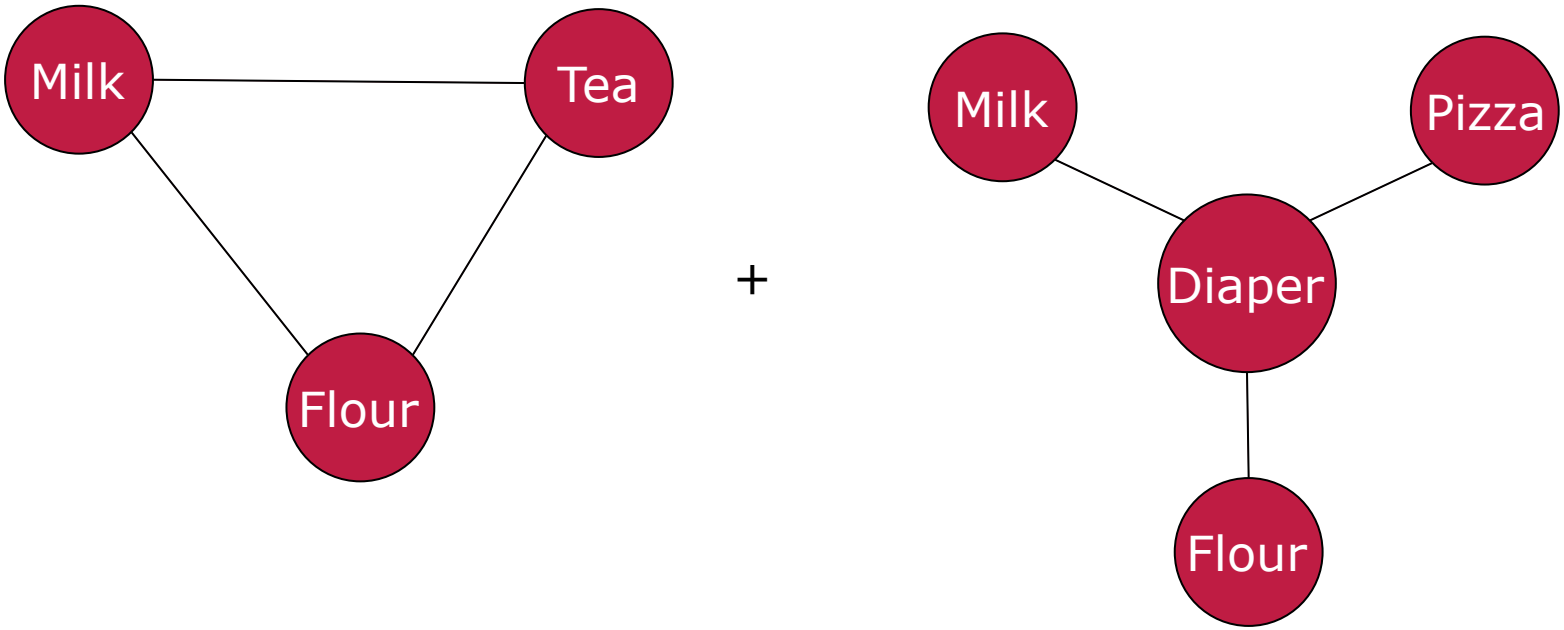
Partition Prefix Equivalence Classes into Cliques

x	[x]
Flour	Milk, Tea
Milk	Tea



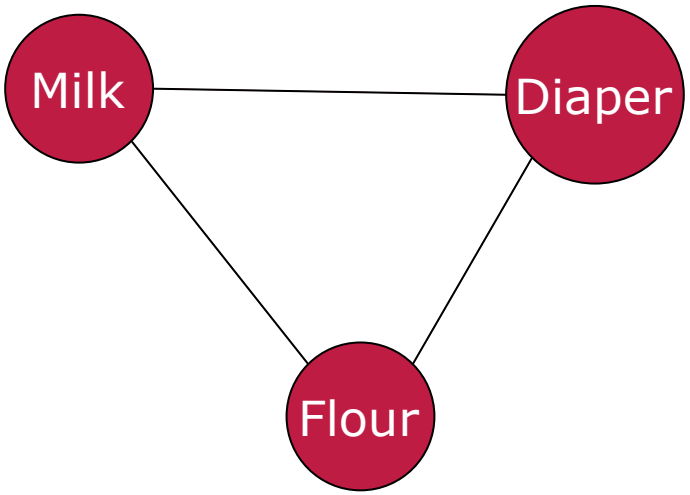
Partition Prefix Equivalence Classes into Cliques

x	[x]
Diaper	Flour, Milk, Pizza
Flour	Milk, Tea
Milk	Tea

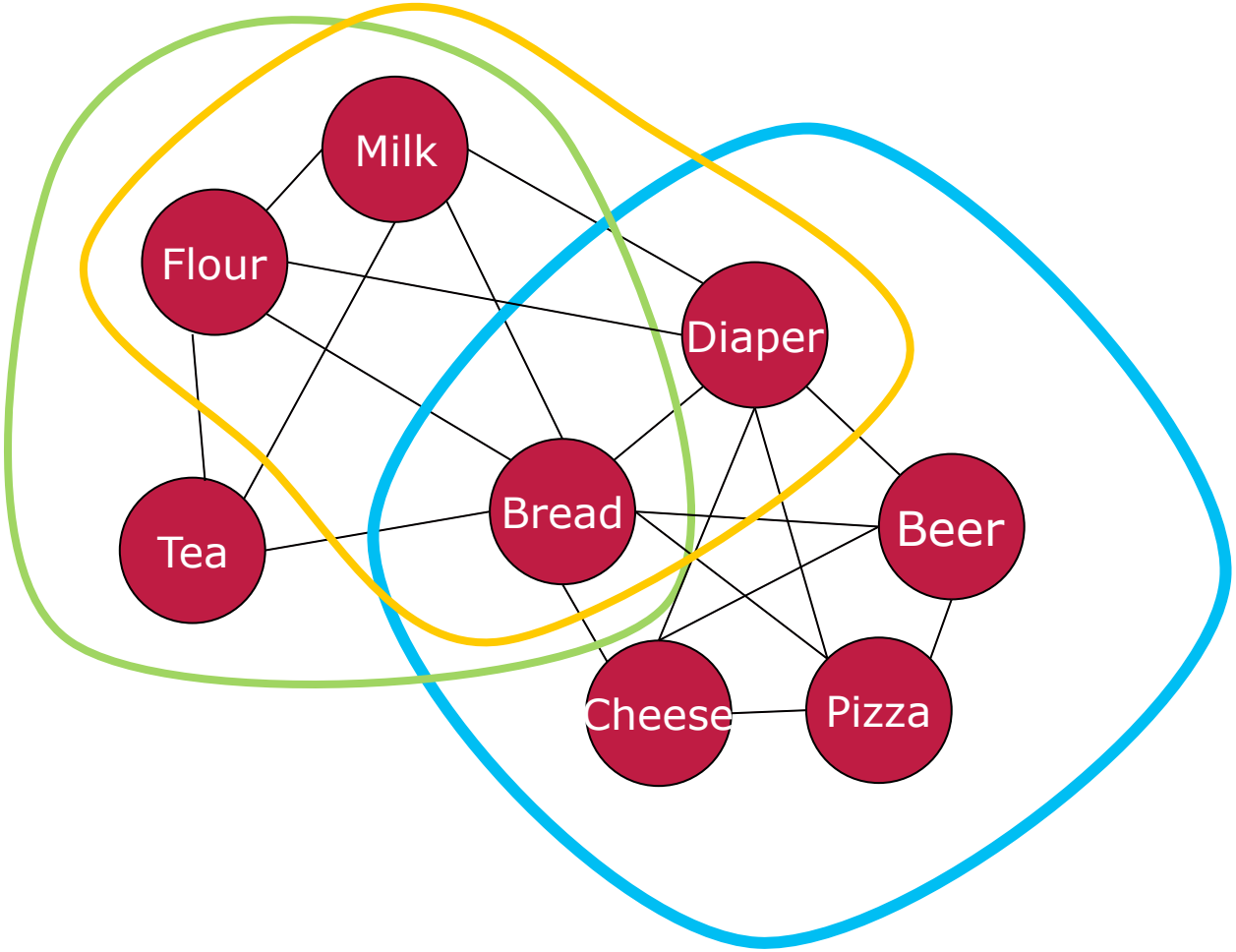


Partition Prefix Equivalence Classes into Cliques

x	[x]
Diaper	Flour, Milk, Pizza
Flour	Milk, Tea
Milk	Tea



Association Graph



Partition Prefix Equivalence Classes into Cliques

x	[x]
Bread	Beer, Cheese, Diaper, Flour, Milk, Pizza, Tea
Beer	Diaper, Cheese, Pizza
Cheese	Diaper, Pizza
Diaper	Flour, Milk, Pizza
Flour	Milk, Tea

x	[x] cliques
Bread	{Bread, Diaper, Beer, Cheese, Pizza}, {Bread, Flour, Milk, Tea}, {Bread, Diaper, Flour, Milk}

Prefix – Clique – Comparison

x	[x]
Bread	Beer, Cheese, Diaper, Flour, Milk, Pizza, Tea

- 7 atoms → $i \leq 2\uparrow n - n - 1 = 2\uparrow 7 - 8$ → ≤ 120 subsets to test
- i = 28 in our case

x	[x] cliques
Bread	{Bread, Diaper, Beer, Cheese, Pizza}, {Bread, Flour, Milk, Tea}, {Bread, Diaper, Flour, Milk}

- 2^4 - 5 + 2^3 – 4 + 2^3 - 4 = 11+ 4 + 4 → ≤19 to test
- i = 17 in our case

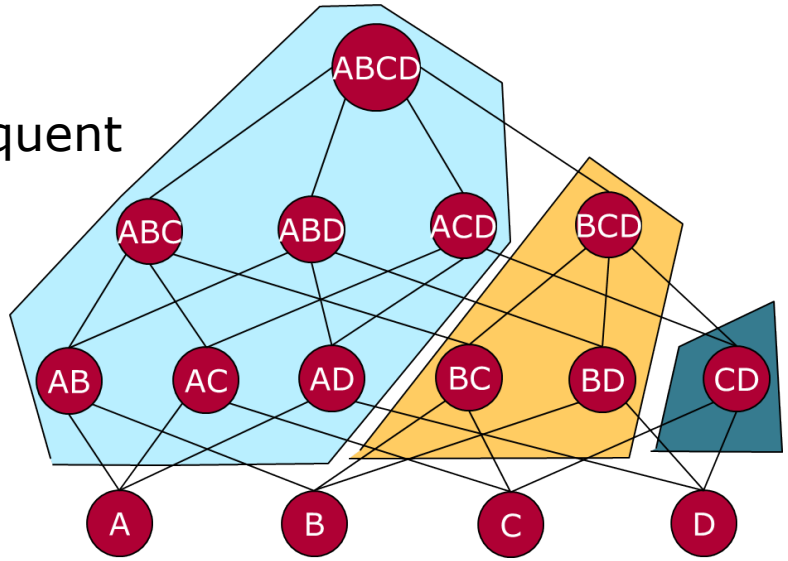
Top-Down vs. Bottom-Up

Bottom up

- Item infrequent -> supersets infrequent
- Faster for many infrequent items

Top down

- Item frequent -> subsets frequent
- Faster for many frequent items



Hybrid Search

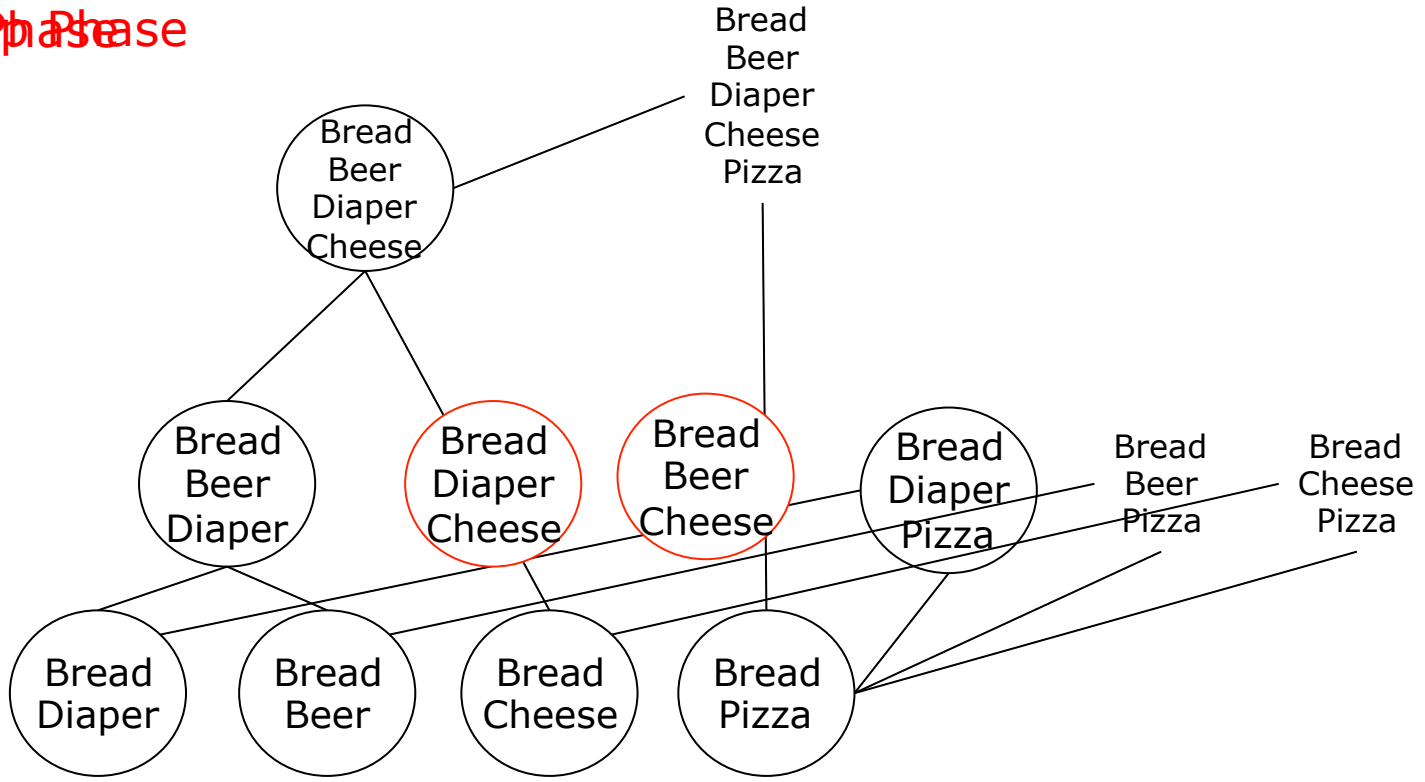
- Based on intuition that very frequent itemsets are subsets of very long maximal itemsets

Extension 2: Hybrid Search

Given: Clique-based pseudo equivalence class, sorted by support

({Bread, Diaper}, {Bread, Beer}, {Bread, Cheese}, {Bread,Pizza})

Bottom-Up Phase Hybrid Phase



Hybrid Search

Eclat Bottom-Up Search

- Guarantees to find all frequent itemsets

MaxClique Hybrid Search

- Guarantees to find all maximal frequent itemsets, plus may find some non-maximal frequent itemsets
- The longer the found maximal frequent itemset, the greater the advantage
- Can be applied recursively or only on first level

Metrics: Support

Cooking

Jack Johnson

Kings of Leon

MTV

Two and a Half Men

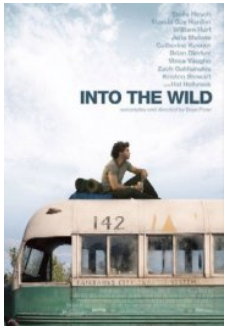
The Hangover

Harry Potter

Better Metrics: Confidence

association rule r:


Kristen Stewart

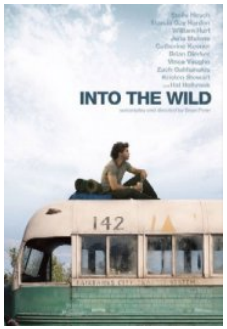


Into The Wild


Twilight


Twilight


Kristen Stewart



Into The Wild

Better Metrics: Lift

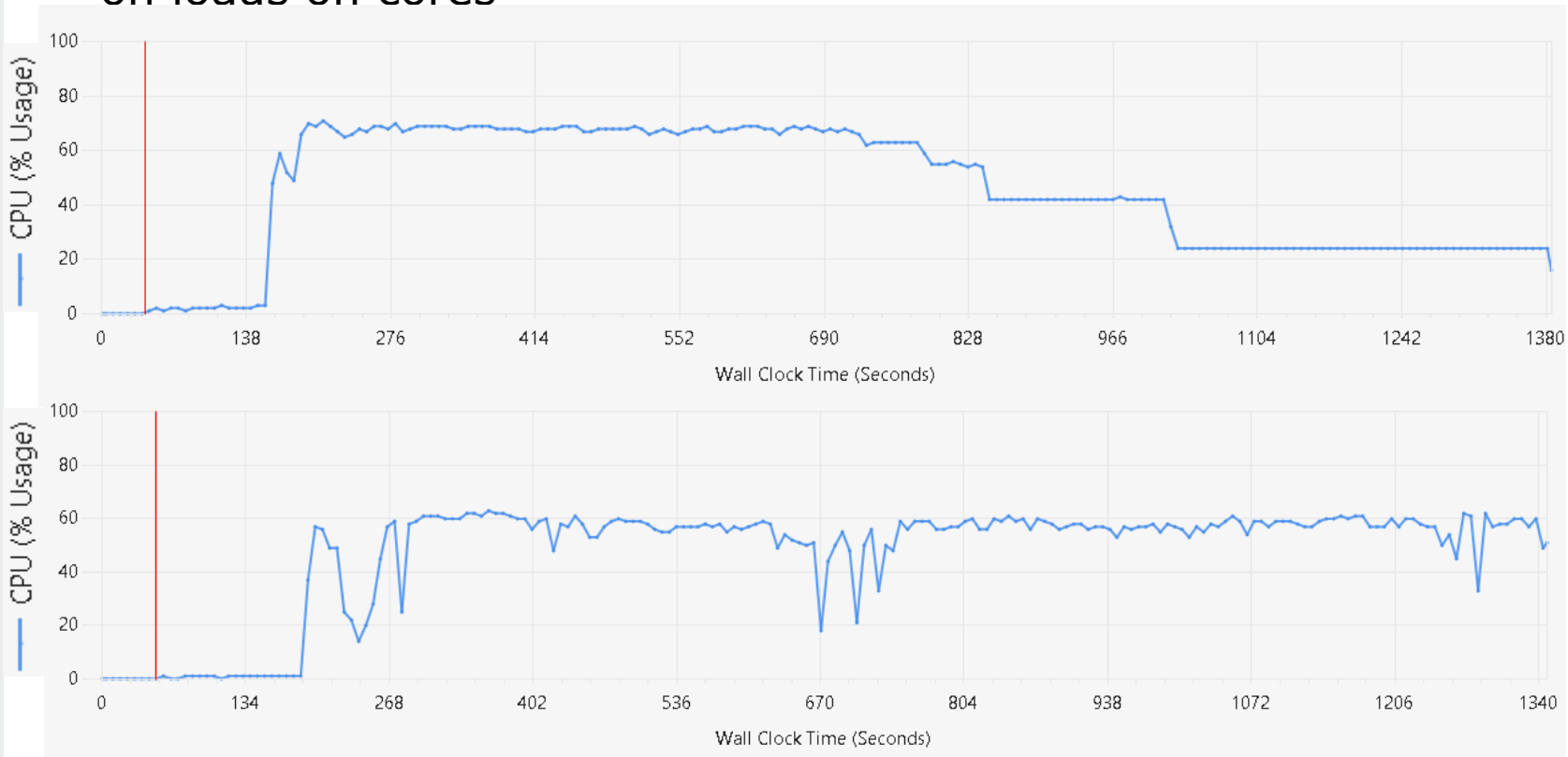
= **50%**

= **45%**

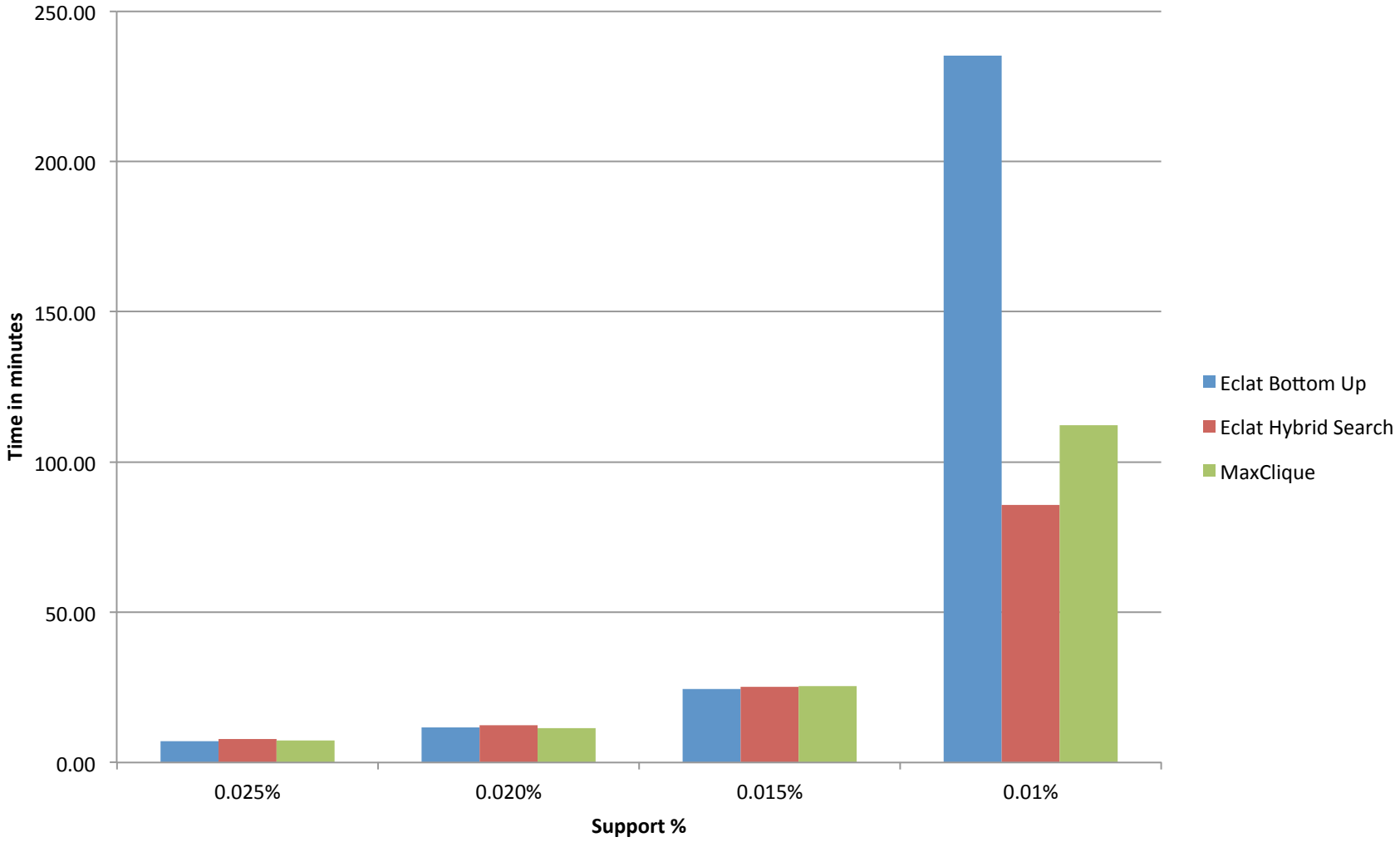
$$lift(A \Rightarrow B) = confidence(A \Rightarrow B) / support(B)$$

Additions to Parallelization

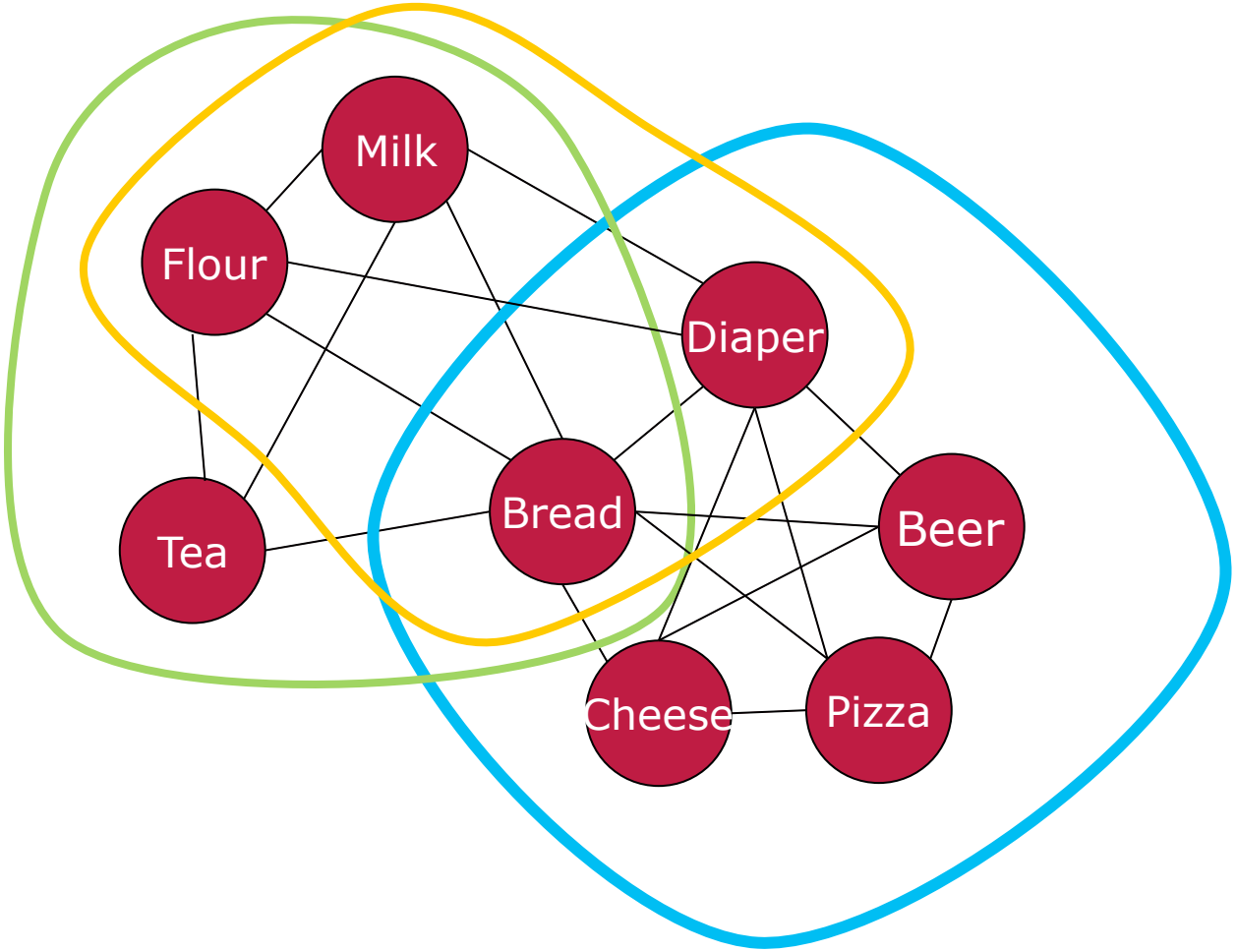
- Calculation of frequent-two-itemsets parallel
- Dynamic use of additional threads after each recursion depending on loads on cores



Benchmarks


250.00
200.00
150.00
100.00
50.00
0.00
Time in minutes
0.025%
0.020%
0.015%
0.01%
Support %
Eclat Bottom Up
Eclat Hybrid Search
MaxClique

Association Graph



Weak Maximal Cliques.

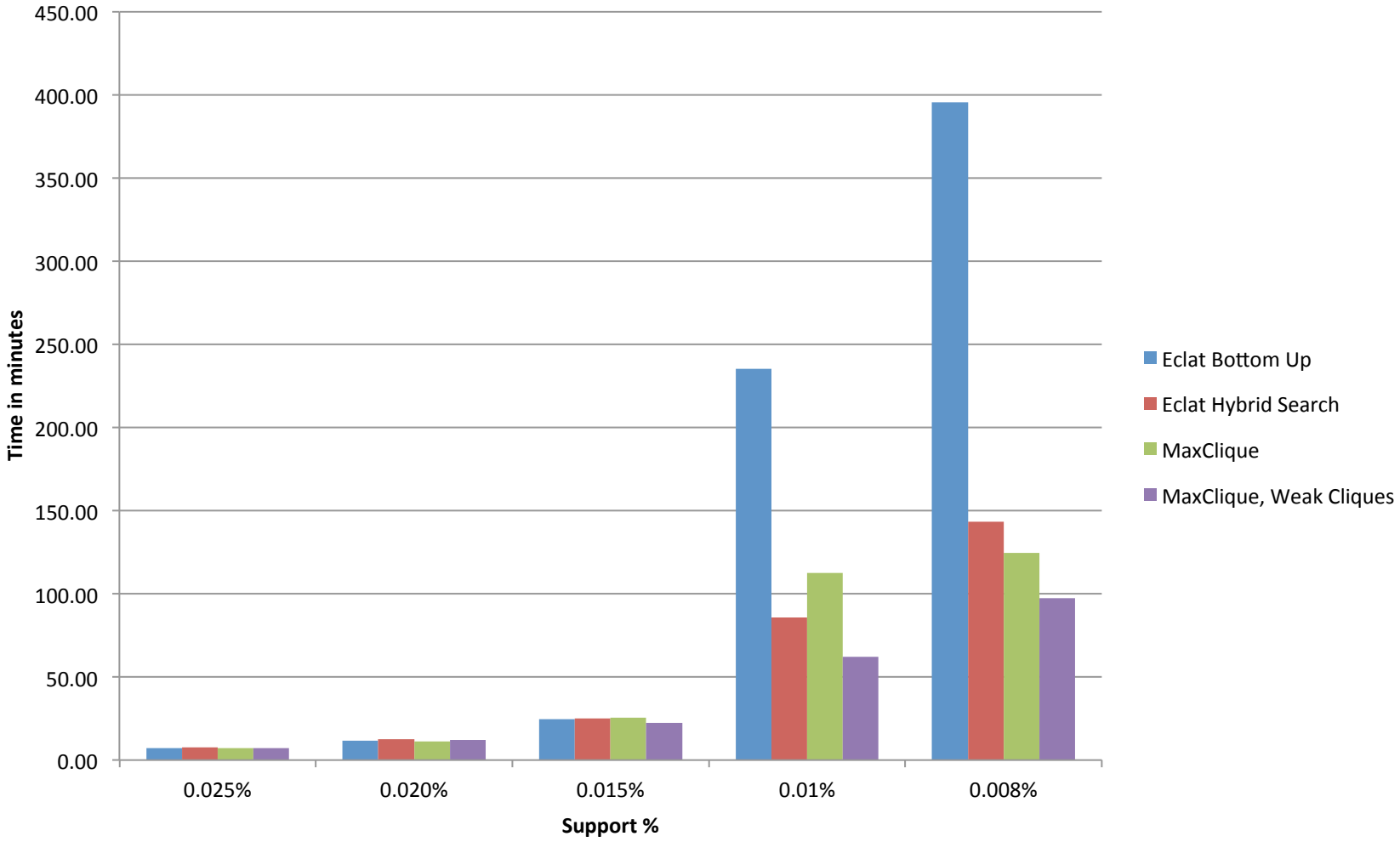
Problem:

Prefix-based class : {12345678}

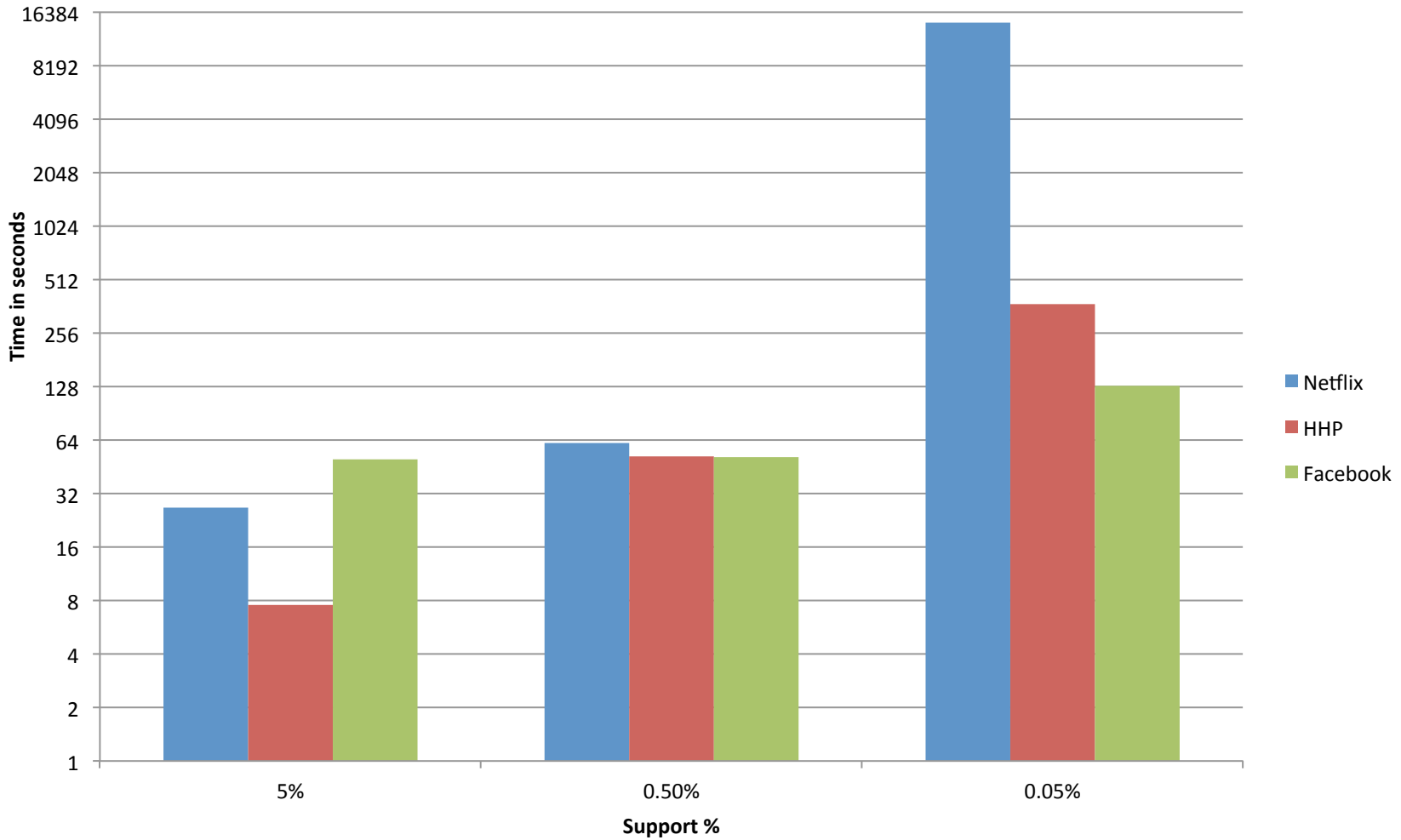
Clique-based classes : {**12345**6, **12345**7, **12345**8}

Solution: cliques X and Y are α-related, if $|X \cap Y| / |X \cup Y| \geq \alpha$

Benchmarks – merge α-related cliques



Benchmarks: Other Data Sets



Discussion

- Pro
  - Partition allows independent and parallel and scalable calculation
  - Vertical Database Format
  - Tid-List Intersection (previous work)
- Con
  - Prerequisite: frequent two-itemsets
    - Hard problem
    - Obtaining two-itemsets might eliminate advantage of partitioning
  - MaxClique improvements
    - advantages highly data-dependent
    - Both involve significant additional computations (Solving the Clique Problem(NP-hard), sorting) which might not pay off