BOINC-based Data Mining

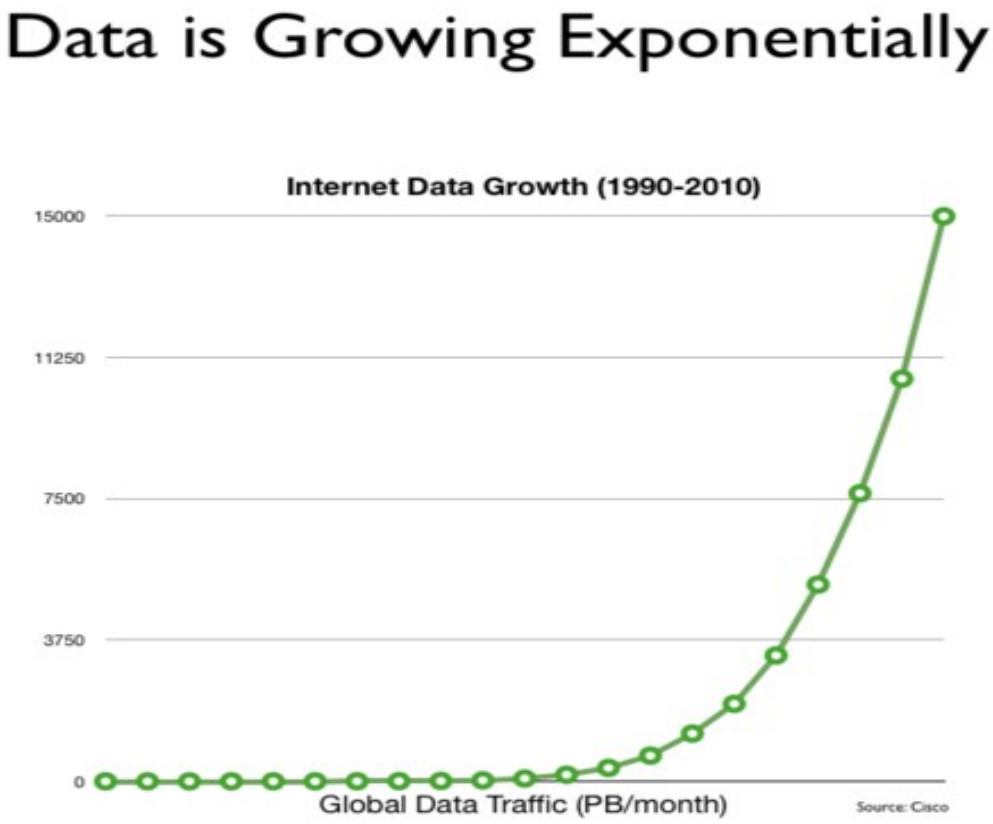
Evgeny Ivashko

Big Data

"Big data" is a term applied to data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time.

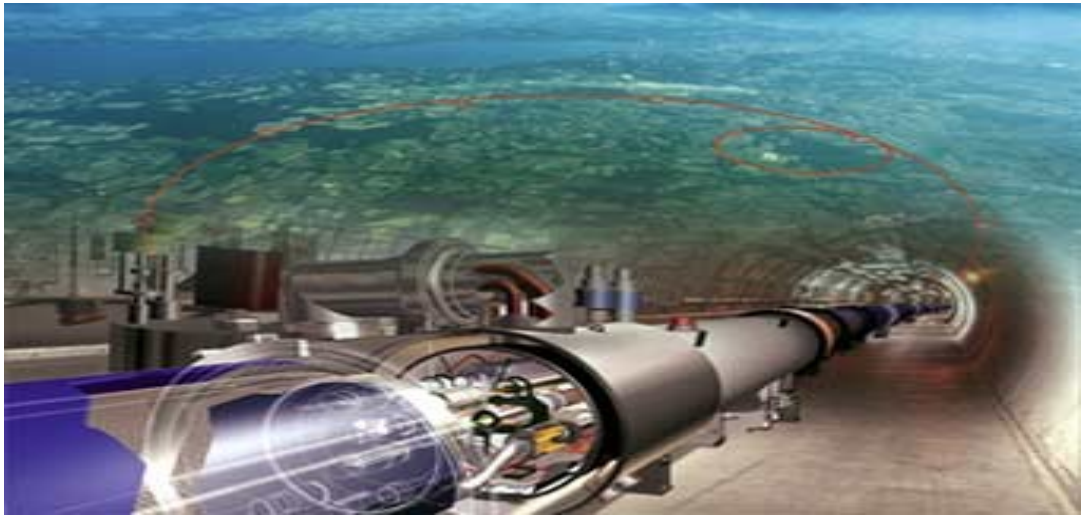
For the enterprise CTO, speaking of "big data" implies the need for a strategy for dealing with large quantities of data.

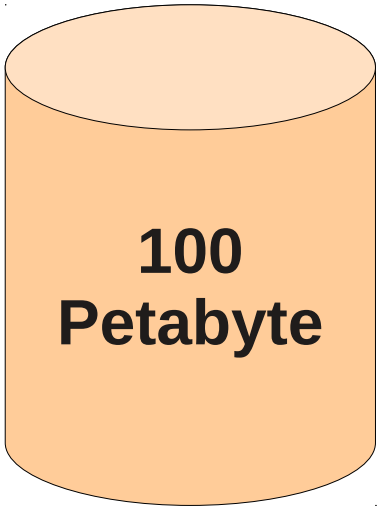
IBM reports that: “There are expected to be 1 trillion new devices connected to the Internet in the near future, which will help drive 44X digital data growth by the year 2020, 80 percent of which will be unstructured content and will require great effort to analyze.”

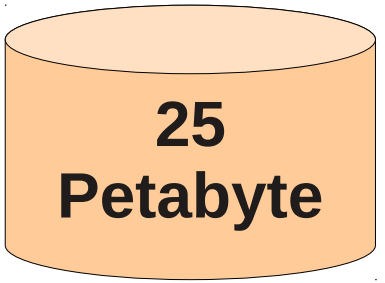


Big Data

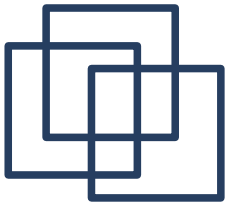
Google







High-Performance Analytics solves complex business problems and delivers highly accurate insights to take impactful decisions and gain competitive advantage



Data Mining

Data mining (the analysis step of the "Knowledge Discovery in Databases" process) is the process that results in the discovery of new patterns in large data sets. It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract knowledge from an existing data set and transform it into a human-understandable structure for further use. Besides the raw analysis step, it involves database and data management aspects, data preprocessing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of found structures, visualization, and online updating.

Association rule learning is a popular and well researched method for discovering relations between variables in large databases.

Association rule is an implication $A, B, C \rightarrow D, E$

Based on the concept of strong rules, association rules were introduced for discovering regularities between products in large scale transaction data. For example, the rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, (s)he is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics.

Basic terms

$I = \{i_1, i_2, i_3, \ldots i_n\}$ – set of elements.

D – set of transactions where each transaction T is a subset of elements from I, $T \subseteq I$

Support of the set of elements is a number of transactions that contain this set of elements.

Set of elements is frequent if its support is greater than certain threshold.

I = {A, B, C, D, E}
T_3 = {D, E, A, C}
support(DE) = 5
or support(DE) = 71,43%

Number	Transaction
1	A B C D E
2	B A C
3	D E A C
4	C D E B
5	B C
6	B D E
7	E D C B

Association rule is an implication $X \Rightarrow Y$, where $X \subset T, Y \subset T$ and $X \cap Y = \emptyset$

The rule $X \Rightarrow Y$ has **support** s if s% transactions of D contain $X \cup Y$

The rule $X \Rightarrow Y$ has **confidence** c if c% transactions of D that contain X also contain Y

Number	Transaction
1	A B C D E
2	B A C
3	D E A C
4	C D E B
5	B C
6	B D E
7	E D C B

Rule $CE \Rightarrow B$

3 of 7 transactions contain (B, C, E) → $supp(CE \Rightarrow B) = 42{,}86\%$

Subset {C,E} appears in 4 transactions, {B, C, E} – in 3 transactions → $conf(CE \Rightarrow B) = 75\%$

Applications

- Segmentation of consumers by behaviour;
- Analysis of consumers preferences;
- Arrangement of goods;
- Analysis of text;
- Analysis and forecasting of equipments failures;
- etc.

Difficulties

- Exponential growth of the computational complexity with the growth of number of elements.
- Growth of the computational complexity with the growth of length and number of transactions.
- Analysis of the file require repeated run with different values of the parameters (minimal support and confidence) to find out non-trivial and valid rules.

Grid Computing

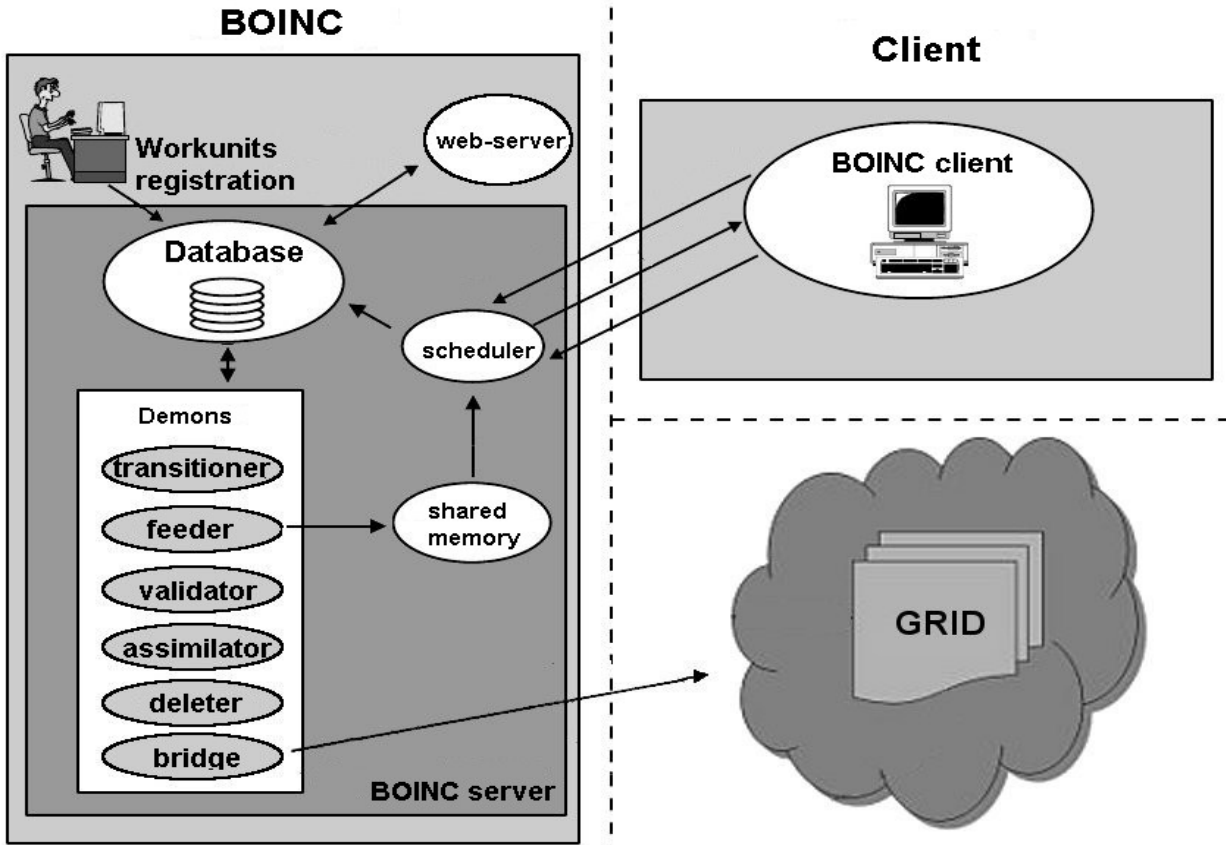
Grid computing *is a term referring to the combination of loosely coupled, heterogeneous, and geographically distributed computer resources from multiple administrative domains to reach a common goal. Grids are constructed with the aid of general-purpose grid software libraries known as middleware.*

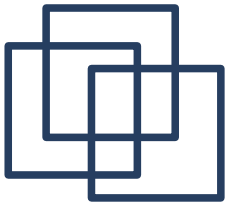
Desktop Grids allow to employ otherwise idle computing time of Desktop computers for large computational programmes. Desktop Grids can be used inside an organisation, or they can collect computing time from volunteers all over a city, a country, or even all over the world.

BOINC

The Berkeley Open Infrastructure for Network Computing (BOINC) is an open source middleware system for volunteer and grid computing. It was originally developed to support the SETI@home project before it became useful as a platform for other distributed applications in areas as diverse as mathematics, medicine, molecular biology, climatology, and astrophysics. The intent of BOINC is to make it possible for researchers to tap into the enormous processing power of personal computers around the world.

BOINC has server-client architecture.



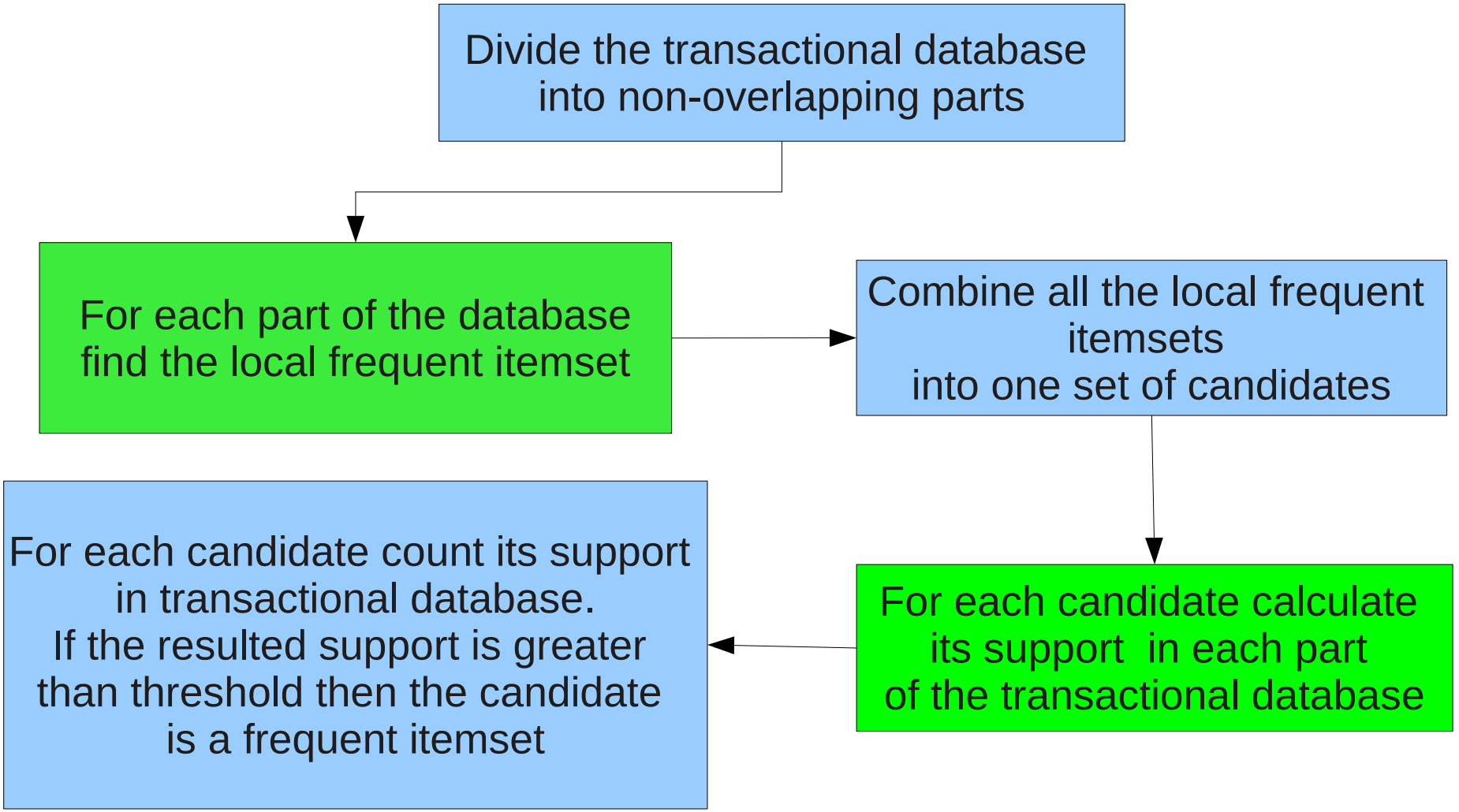


Aim

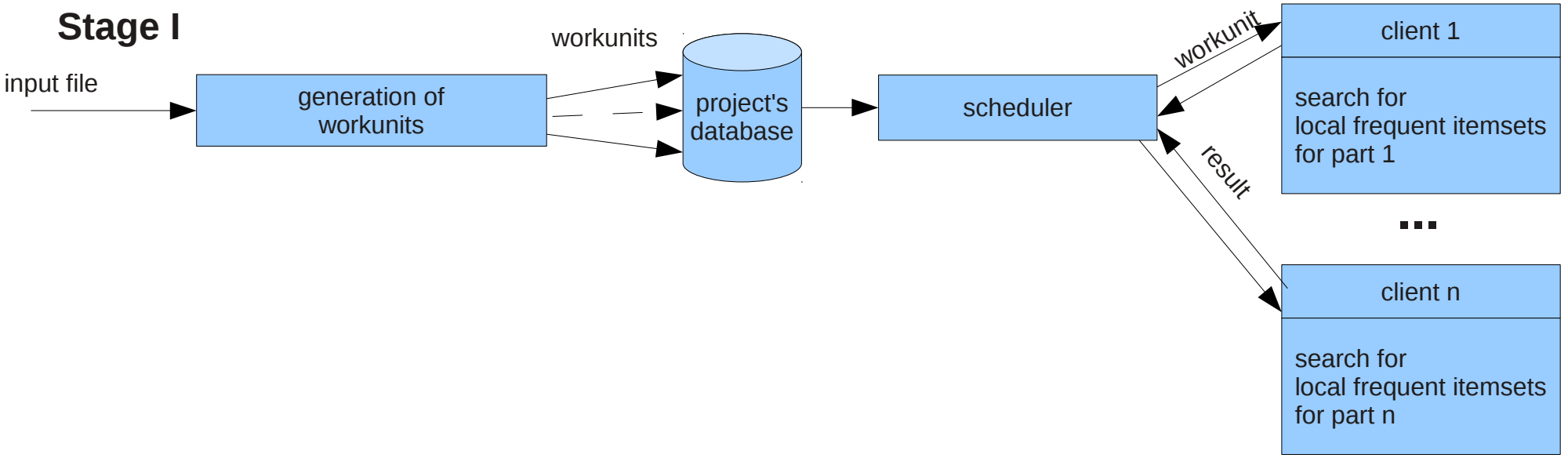
The aim of the project is to develop a BOINC-based system that

- Can process large (and very large) files
- Has good scalability
- Has small enough overhead
- Automatically adopted to the grid structure

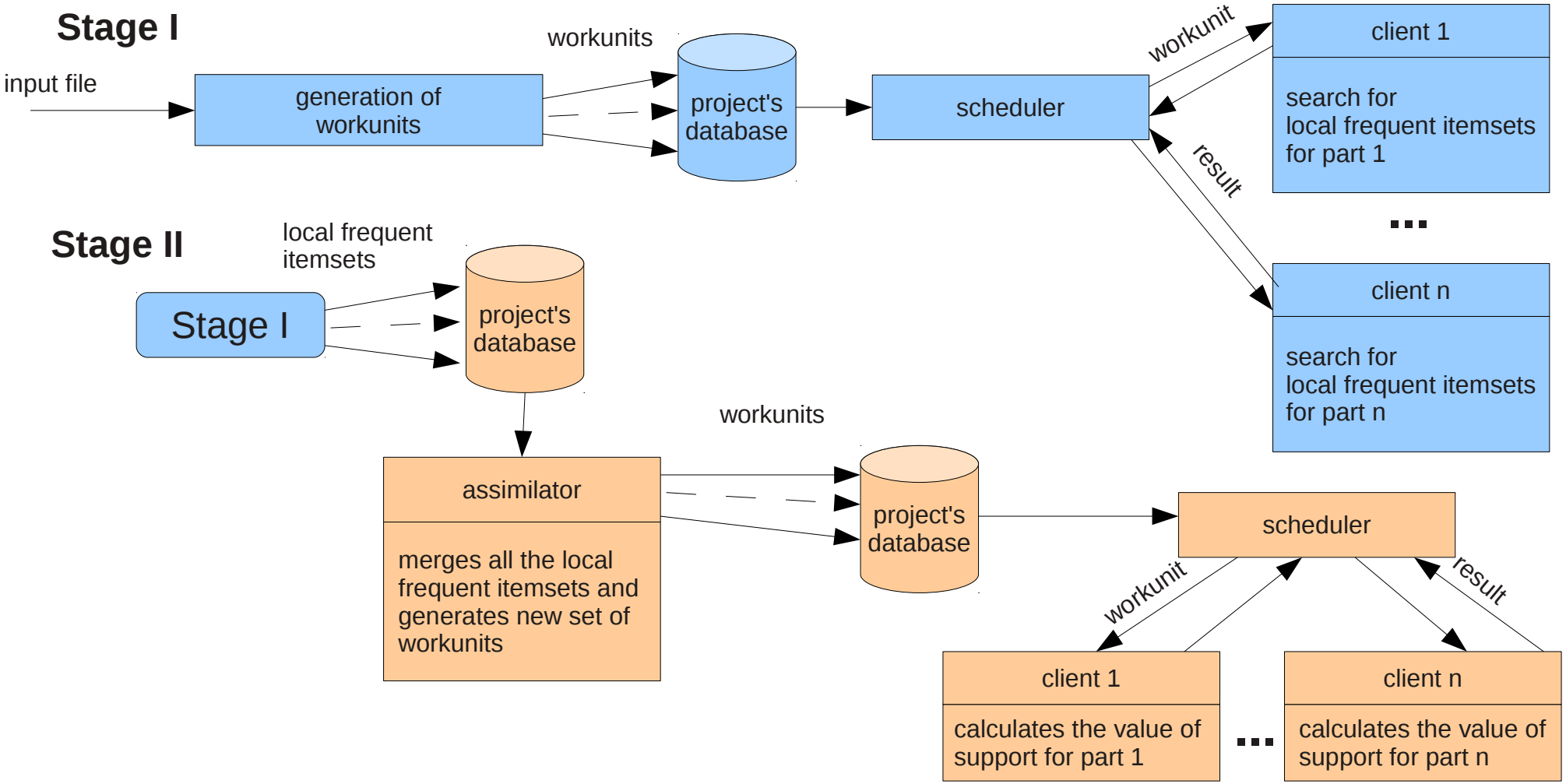
Algorithm Partition



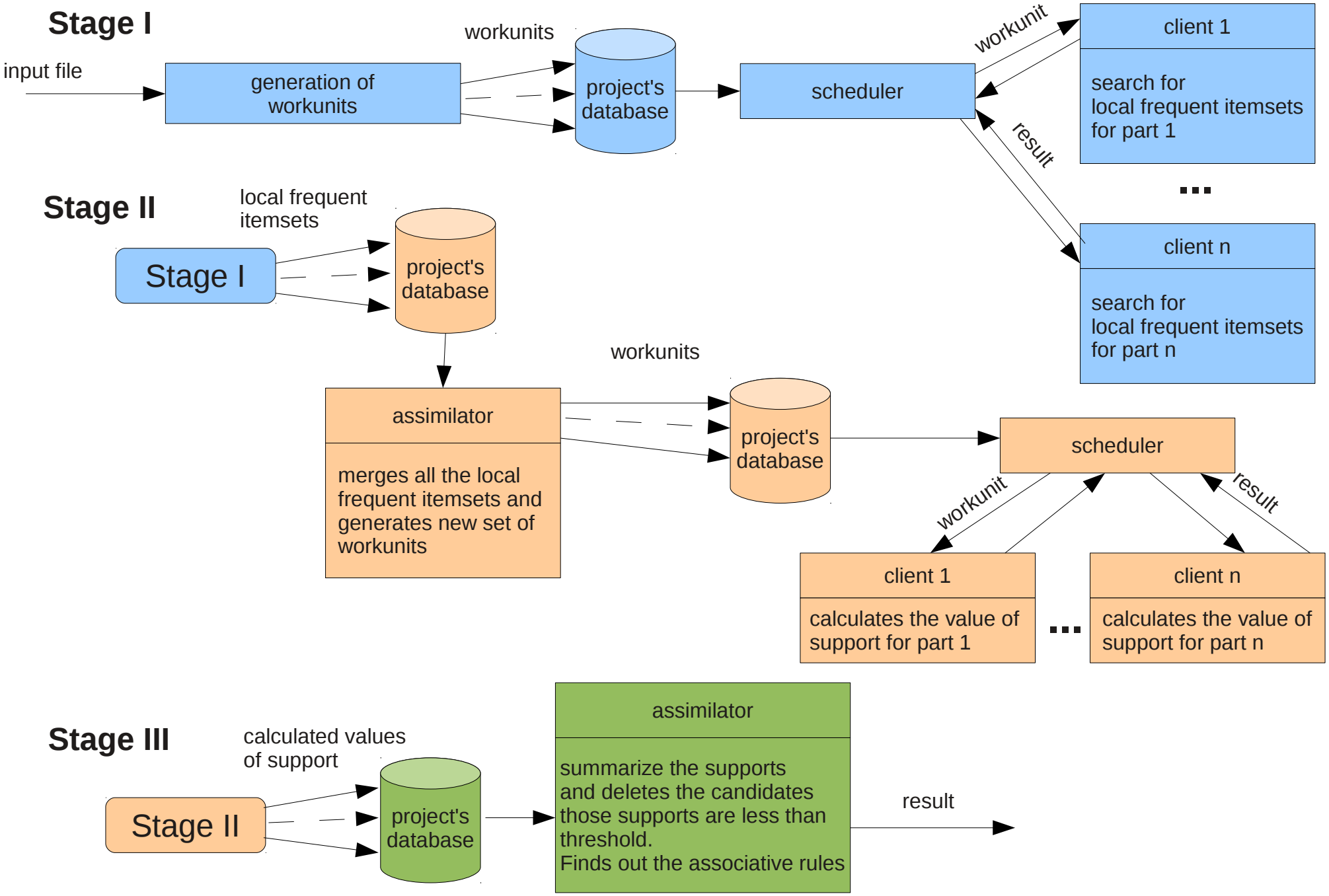
Stage I


input file
generation of workunits
workunits
project's database
scheduler
workunit
result
client 1
search for local frequent itemsets for part 1
...
client n
search for local frequent itemsets for part n

Implementation of Partition in BOINC



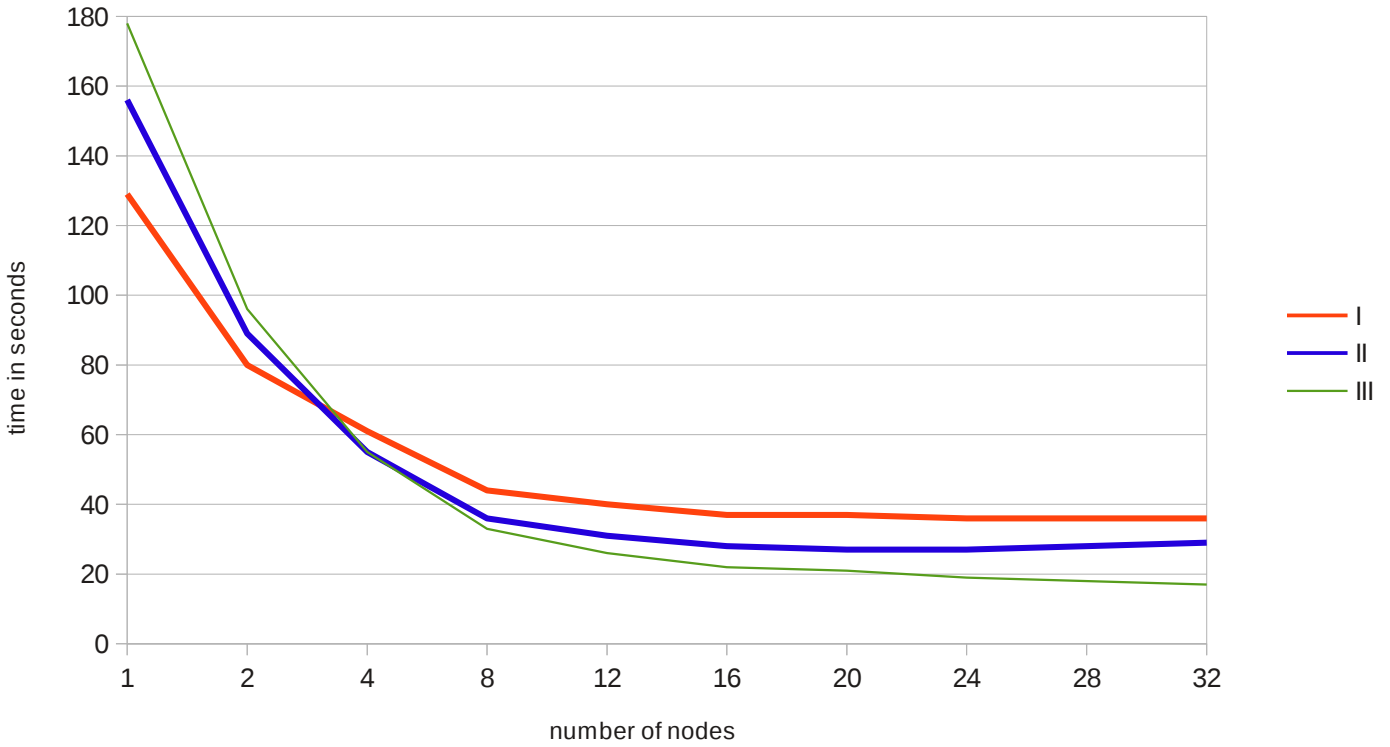
Implementation of Partition in BOINC



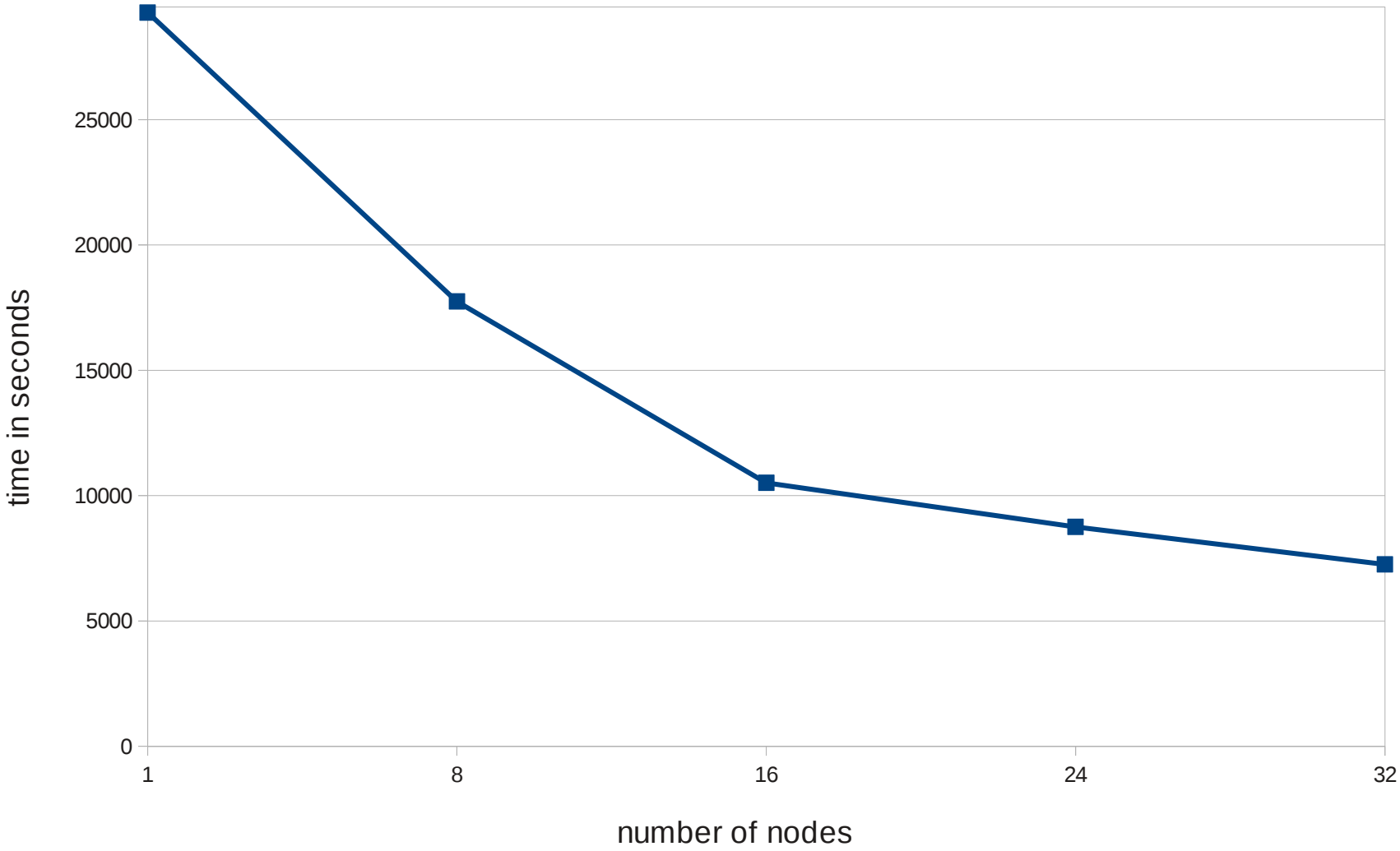
Experiments

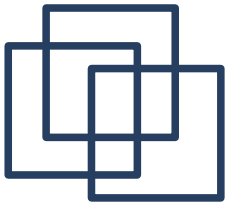
Experiments are based on itemsets from Frequent Itemset Mining Dataset Repository:

	File	Number of transactions	Average length of transaction	Minimal support
I	T10I4D100K.dat	100000	10	1%
II	T25I20D100K.dat	100000	25	1,5%
III	T40I10D100K.dat	100000	40	5%



Analysis of the 100Gb file





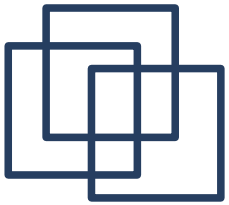
Achievements

Application

Experiments (up to 100Gb files)

Analysis of the weak points

Analysis of the overhead



Thank you for your attention!

Алгоритм Apriori. Свойство антимонотонности

Алгоритм Apriori — классический алгоритм поиска ассоциативных правил. Главное свойство — **свойство антимонотонности**, уменьшающее размерность пространства поиска:

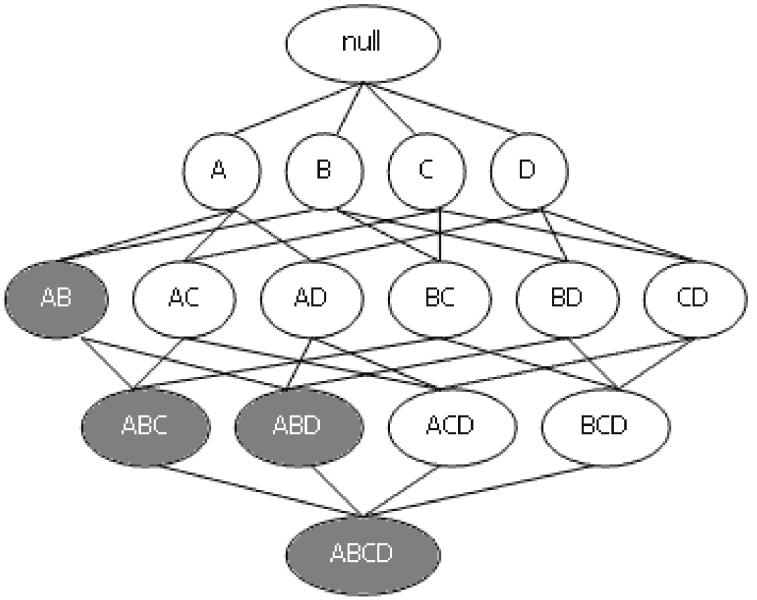
поддержка любого набора элементов не может превышать минимальной поддержки любого из его подмножеств.

Любая транзакция, содержащая {A, B, C} должна также содержать наборы {A, B}, {A, C} и {B, C}, причем обратное не верно. Свойству антимонотонности можно дать и другую формулировку:

с ростом размера набора элементов

поддержка уменьшается

или остается прежней.



Алгоритм Apriori

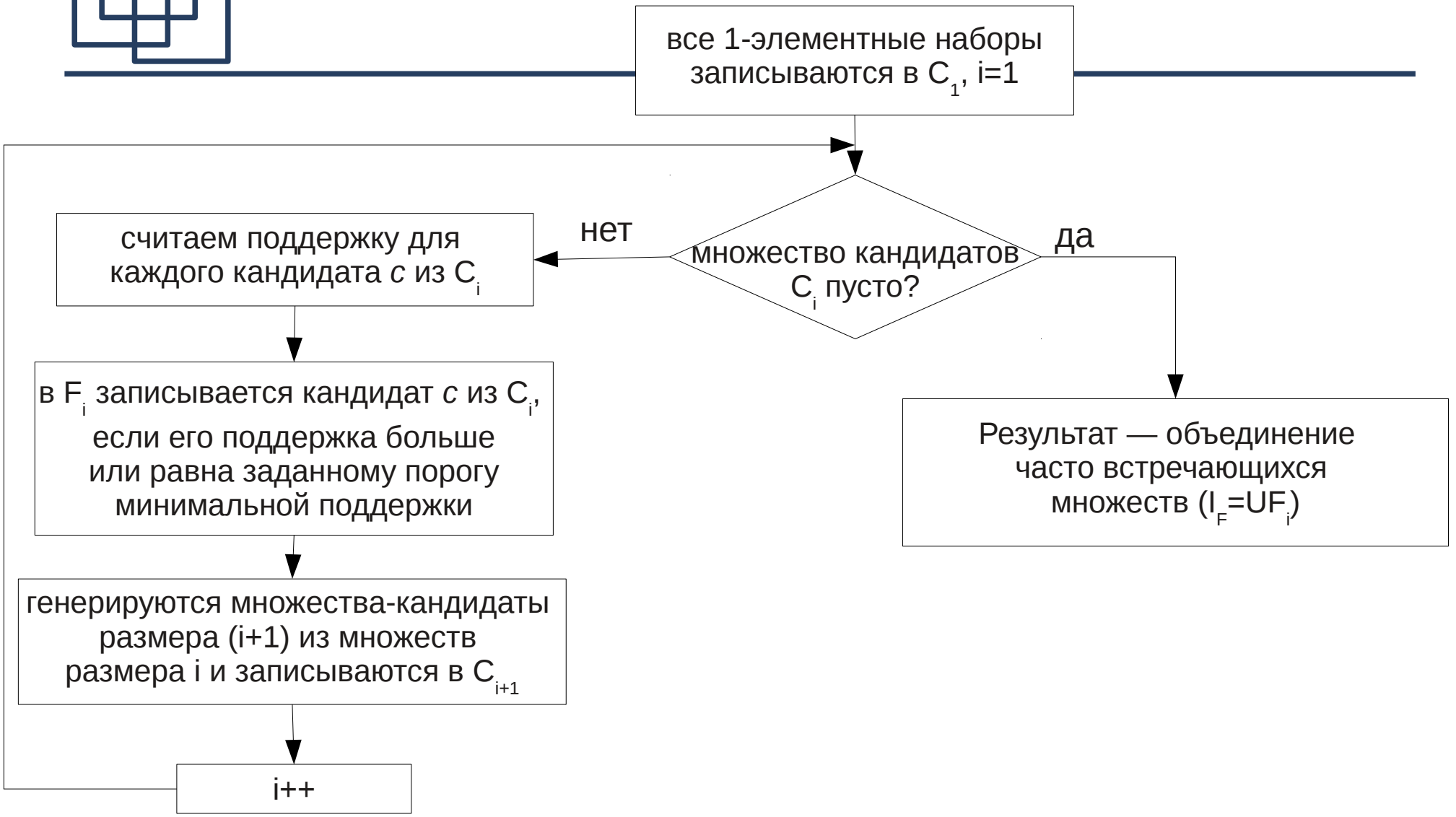


Схема алгоритма Partition

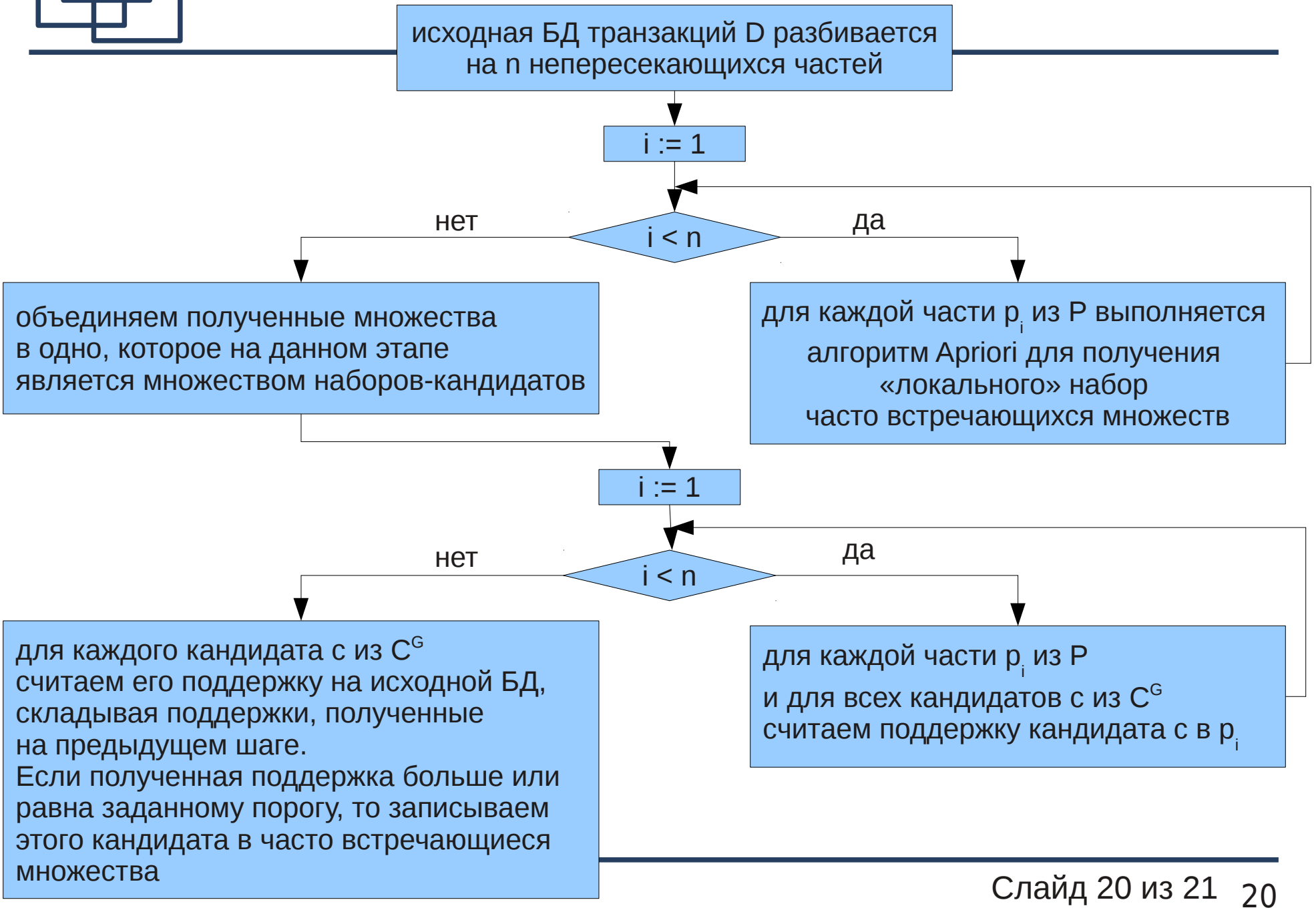


Схема алгоритма Partition