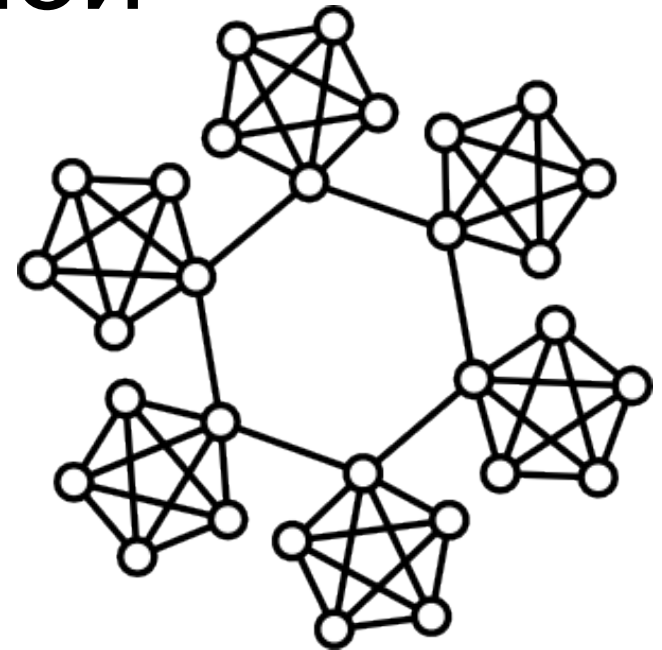


65-я научная конференция ЮУрГУ
25 апреля 2013
г. Челябинск

Разбиение сверхбольших графов с помощью реляционной параллельной СУБД

Пан К.С.,
Цымблер М.Л.,
кафедра системного
программирования



Разбиение графа

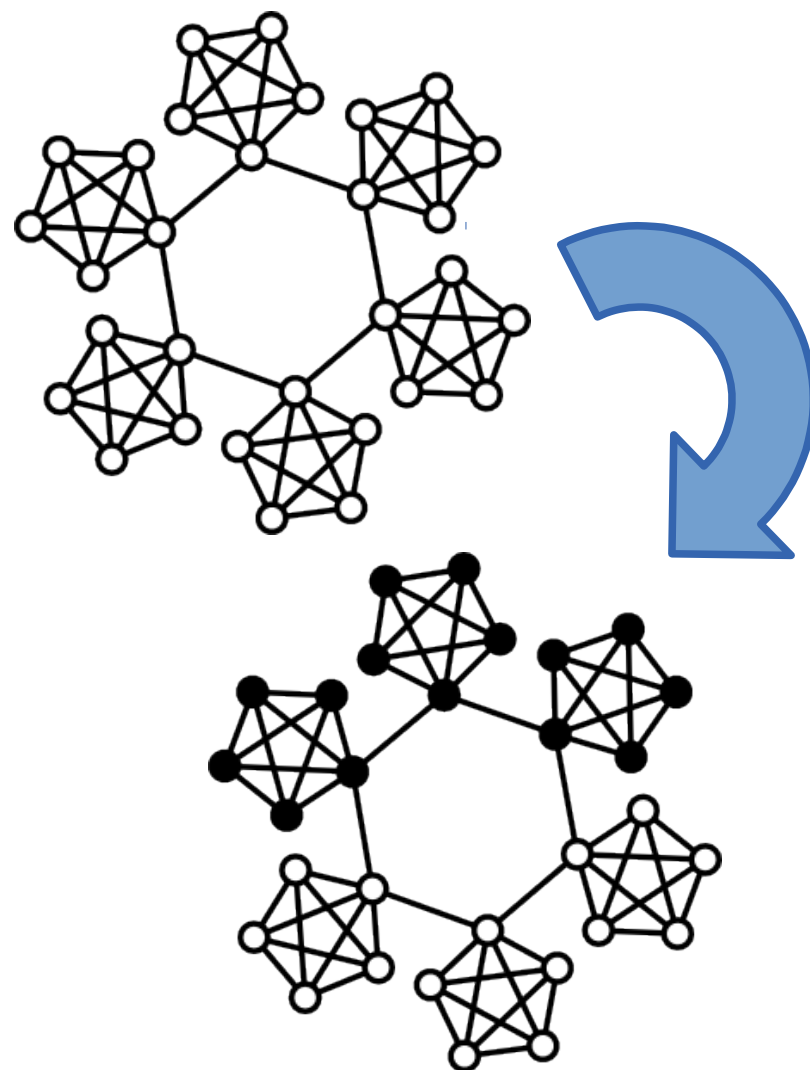
$$G = (N, E)$$

$$\bigcup_{i=1}^p N_i = N$$

$$N_i \cap N_j = \emptyset \quad \forall i \neq j$$

$$|N_i| \approx \frac{|N|}{p}, \quad \forall i$$

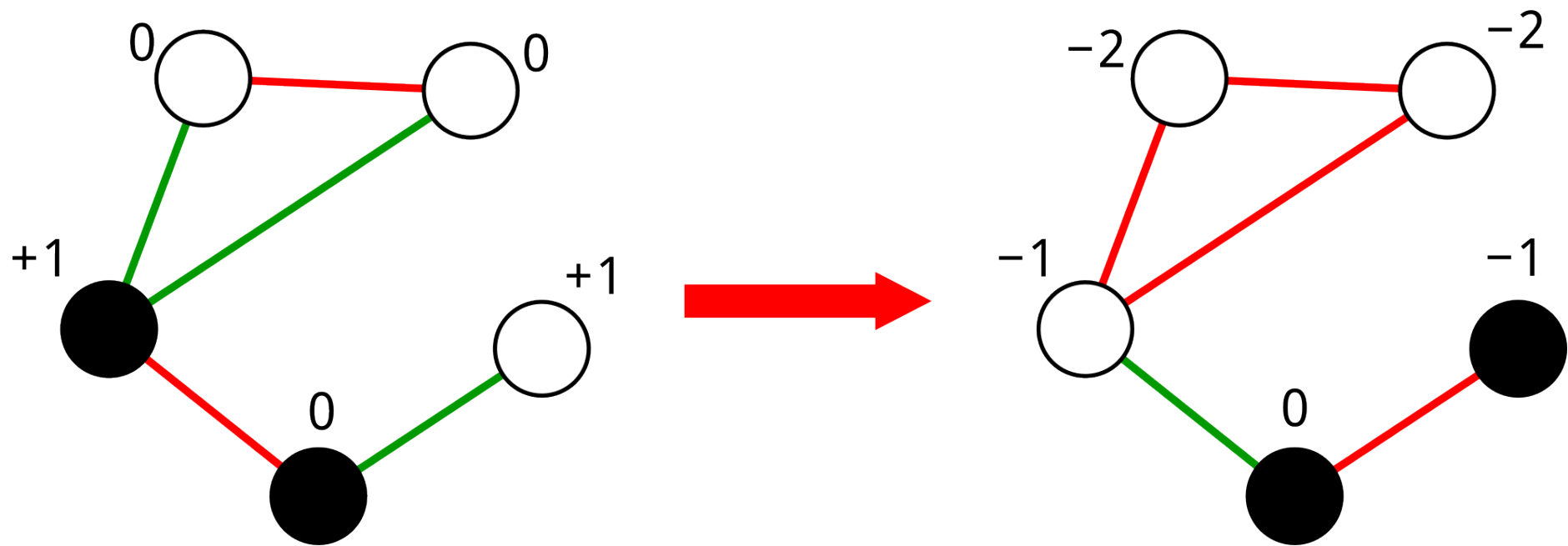
$$\sum_{u,v:P(u) \neq P(v)} W(u,v) \rightarrow \min$$



Приложения

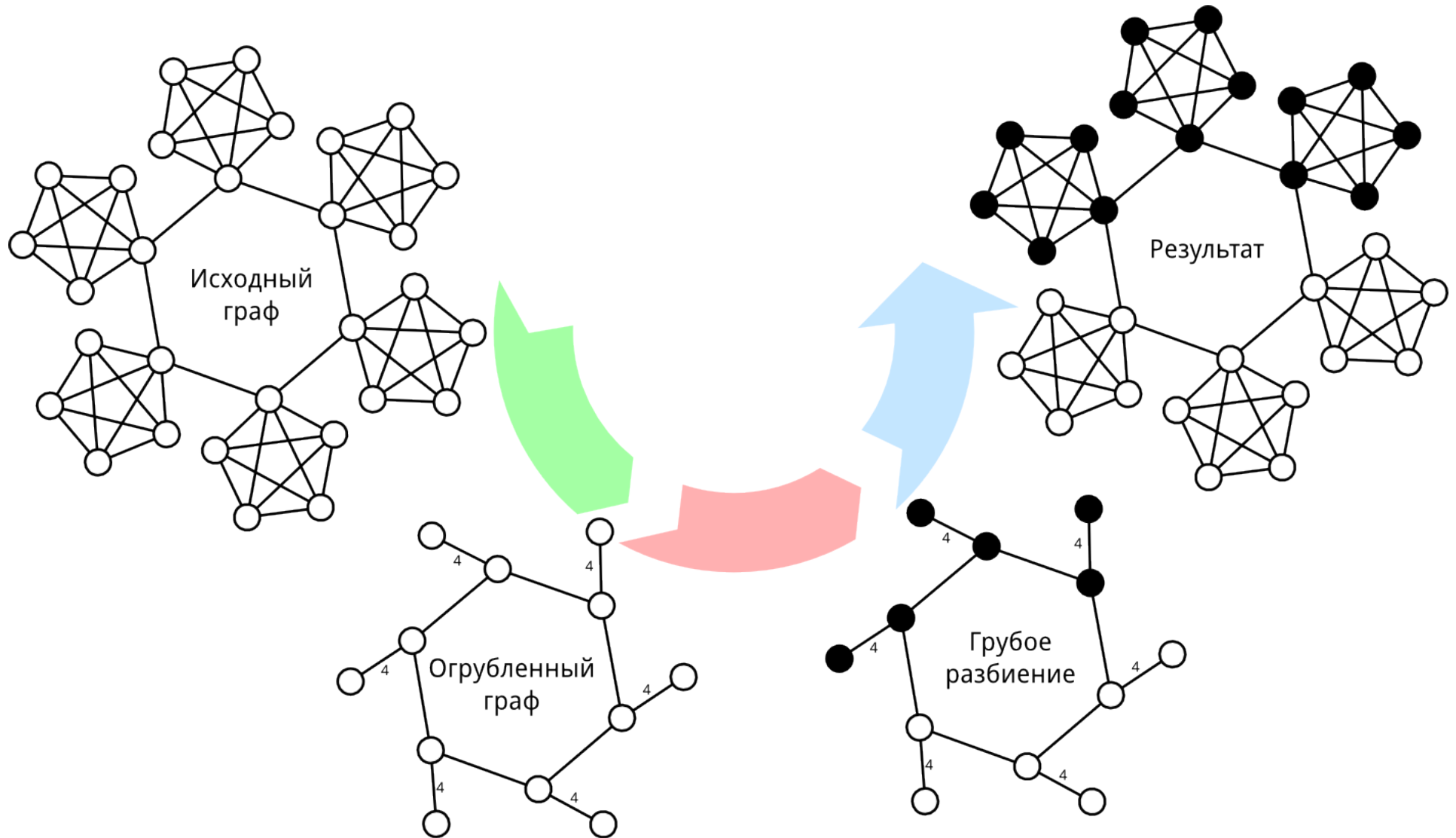
- Проектирование сложных электронных схем
- Проектирование топологии локальной сети
- Конечно-элементное моделирование
- Распределение графа алгоритма на многопроцессорную систему
- Интеллектуальный анализ
 - социальных и биологических сетей,
 - химических соединений и белковых структур
 - web, xml-документов и др.

Основные подходы: эвристика Кернигана—Лина

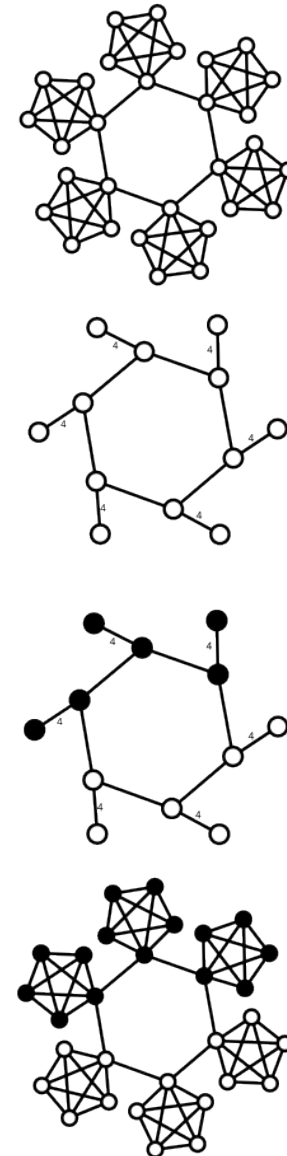
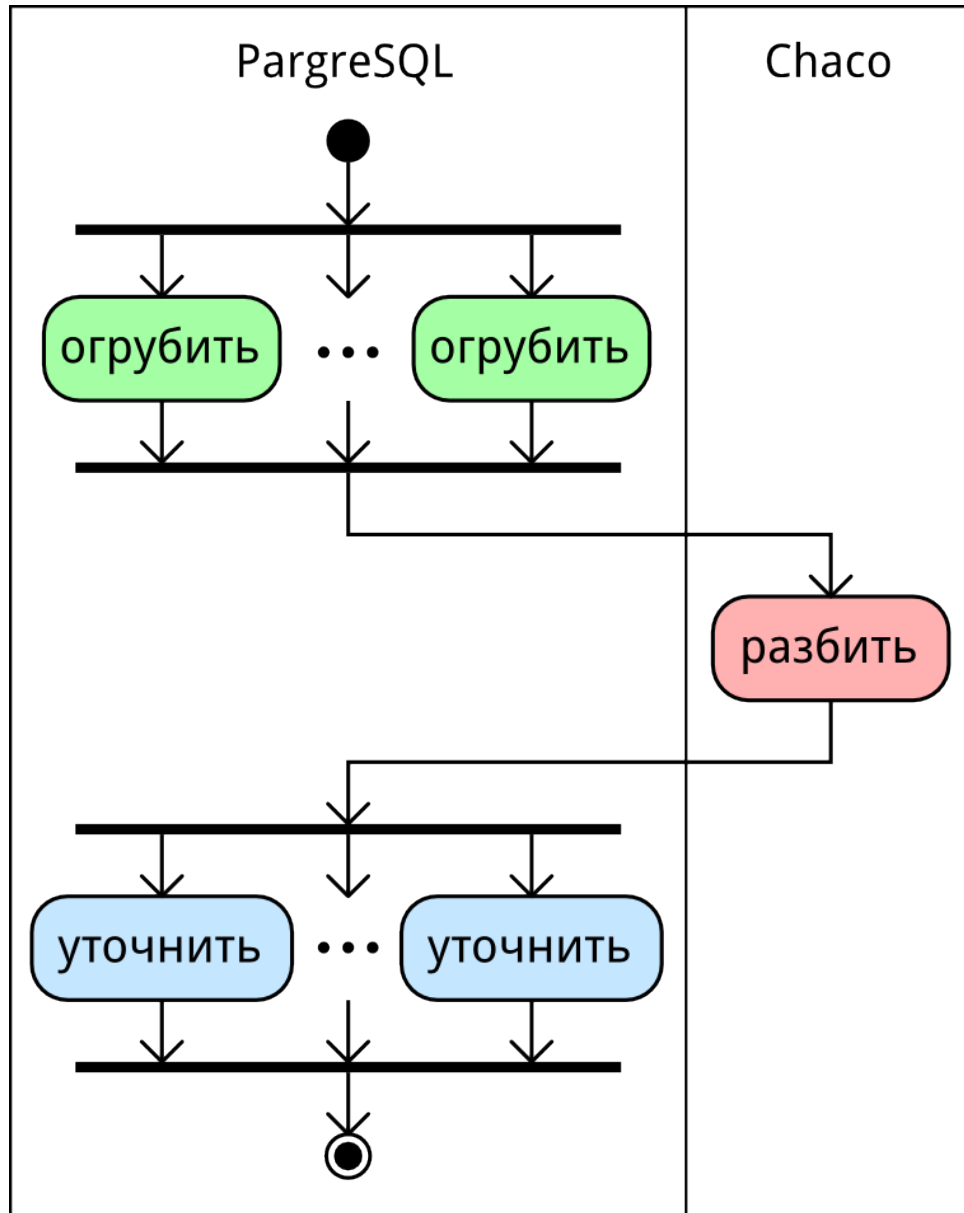


$$\text{gain}(v) = \sum_{(v,u) \in E, P(v) \neq P(u)} w(v,u) - \sum_{(v,u) \in E, P(v) = P(u)} w(v,u)$$

Основные подходы: многоуровневое разбиение



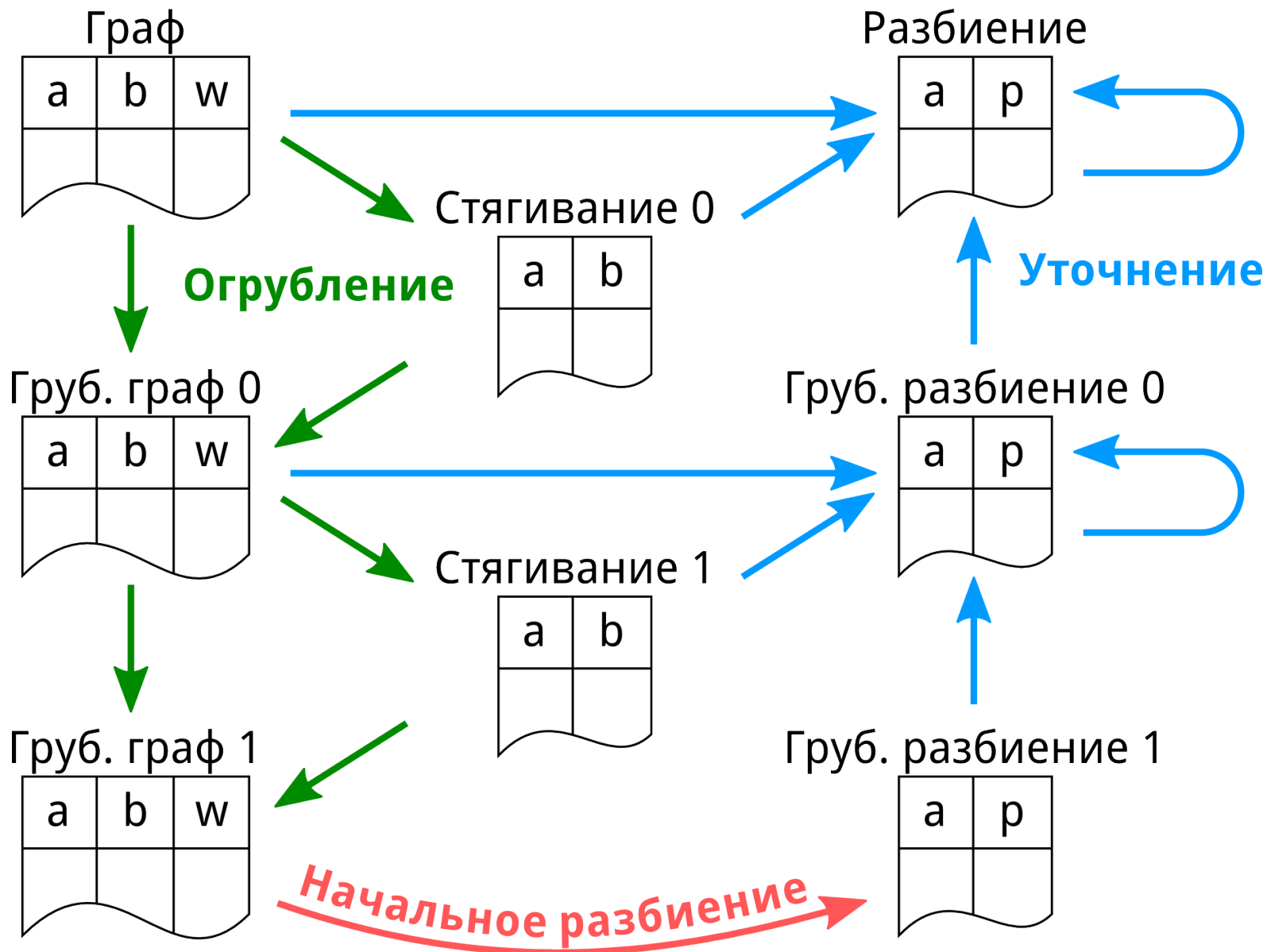
Разбиение с помощью ПСУБД



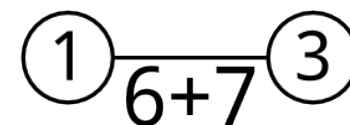
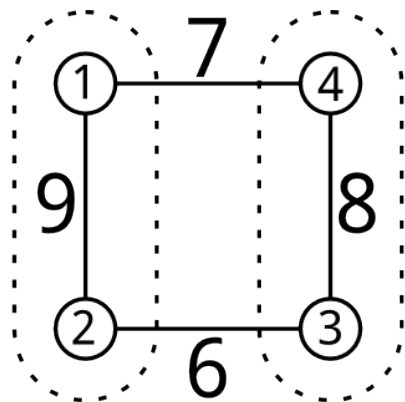
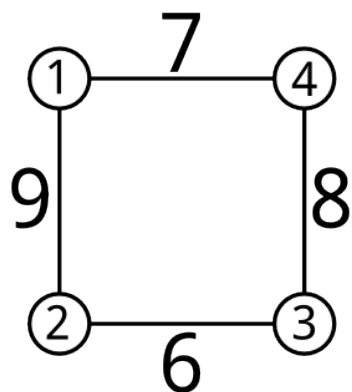
Реляционная структура

Таблица (имя и поля)	Семантика
GRAPH(A, B, W)	Исходный граф в виде списка ребер A, B – концы ребра, W – вес ребра
MATCH(A, B)	Максимальное паросочетание исходного графа A, B – концы ребра
COARSE_GRAPH(A, B, W)	Огрубленный граф A, B – концы ребра, W – вес ребра
COARSE_PARTITIONS(A, P)	Начальное разбиение огрубленного графа A – номер вершины, P – цвет вершины
PARTITIONS(A, P, G)	Разбиение исходного графа A – номер вершины, P – цвет вершины, G – значение функции выгоды

Схема разбиения



Реализация огрубления



поиск

стягивание

GRAPH	A	B	W
1	2	9	
2	3	6	
3	4	8	
4	1	7	

MATCH	A	B
1	2	
3	4	

COARSE GRAPH	A	B	W
1	3	13	

Реализация огрубления

-- поиск

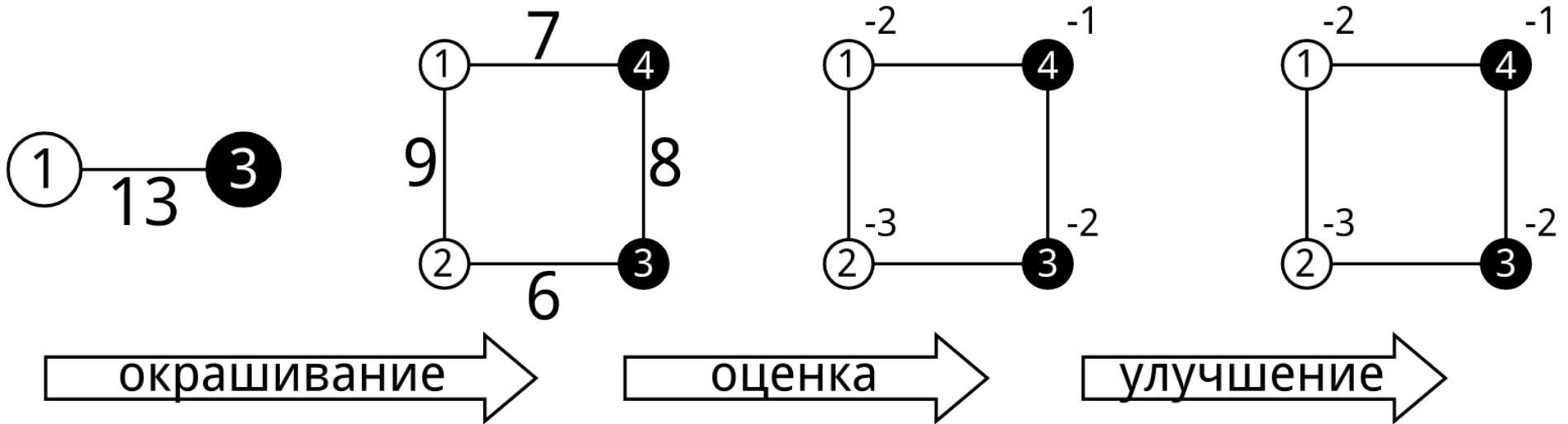
```
for edge in (select A,B from GRAPH order by W desc)
loop
  if not exists(select * from visited where A =
edge.A or A = edge.B) then
    insert into visited values (edge.A);
    insert into visited values (edge.B);
    insert into MATCH values (edge.A, edge.B);
  end if;
end loop;
```

Реализация огрубления

-- стягивание

```
select least(newA, newB) as A, greatest(newA,
newB) as B, sum(W) as W
from (
  select
    coalesce(match2.A, GRAPH.A) as newA,
    coalesce(MATCH.A, GRAPH.B) as newB,
    GRAPH.W
  from
    GRAPH, left join MATCH on GRAPH.B=MATCH.B
    left join MATCH as match2 on GRAPH.A=match2.B)
where newA<>newB group by A, B
```

Реализация уточнения



COARSE PARTITIONS	A	P
	1	1
3	0	0

PARTITIONS	A	P
	1	1
2	1	1
3	0	0
4	0	0

PARTITIONS	A	P	G
	1	1	1
2	1	1	-3
3	0	0	-2
4	0	0	-1

PARTITIONS	A	P	G
	1	1	1
2	1	1	-3
3	0	0	-2
4	0	0	-1

Реализация уточнения

-- окрашивание

```
select a, p from COARSE_PARTS
union
select match.b, part.p
from MATCH as match, COARSE_PARTS as
part
where match.a = part.a
```

Реализация уточнения

-- оценка

```
select PARTITIONS.A, PARTITIONS.P, sum(subgains.Gain) as Gain
from
  PARTITIONS left join (
    select GRAPH.A, GRAPH.B,
    case when ap.P = bp.P then -GRAPH.W
    else GRAPH.W end as Gain
    from
      GRAPH left join PARTITIONS as ap on GRAPH.a = ap.A
      left join PARTITIONS as bp on GRAPH.b = bp.A
    ) as subgains
  on PARTITIONS.A = subgains.A or PARTITIONS.A = subgains.B
group by PARTITIONS.A, PARTITIONS.P
```

Реализация уточнения

-- улучшение

```
select * from PARTITIONS
where
  P = current and G = (select max(G) from PARTITIONS
    where P = current)
limit 1
into V

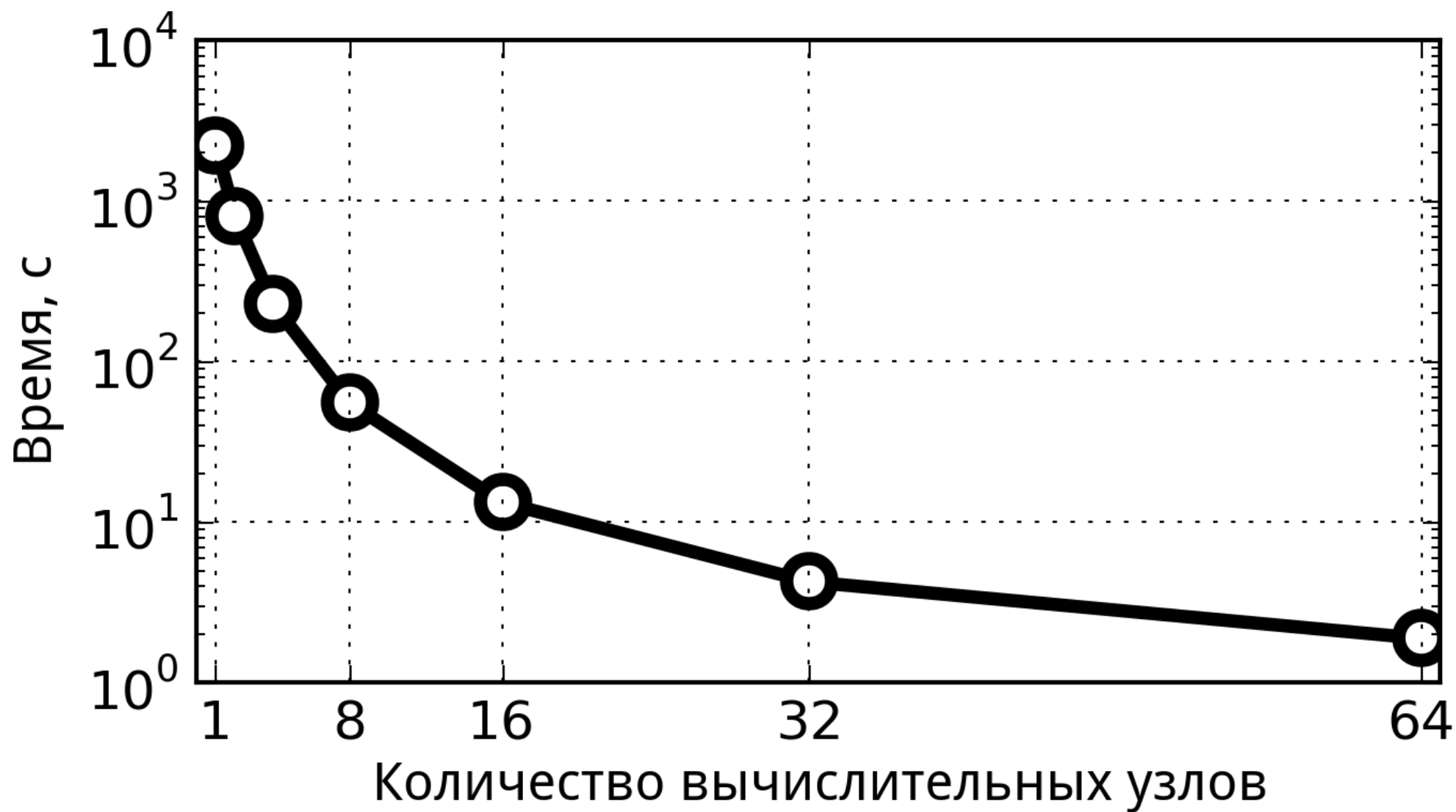
update PARTITIONS
  set G = G + W * (case when P = V.P then 2 else -2 end)
from (
  select case when A = V.A then B else A end, W from GRAPH
  where B = V.A or A = V.A) as neighbors
where neighbors.A = PARTITIONS.A;

update PARTITIONS
  set G = -G, P = 1 - P
where A = V.A;
```

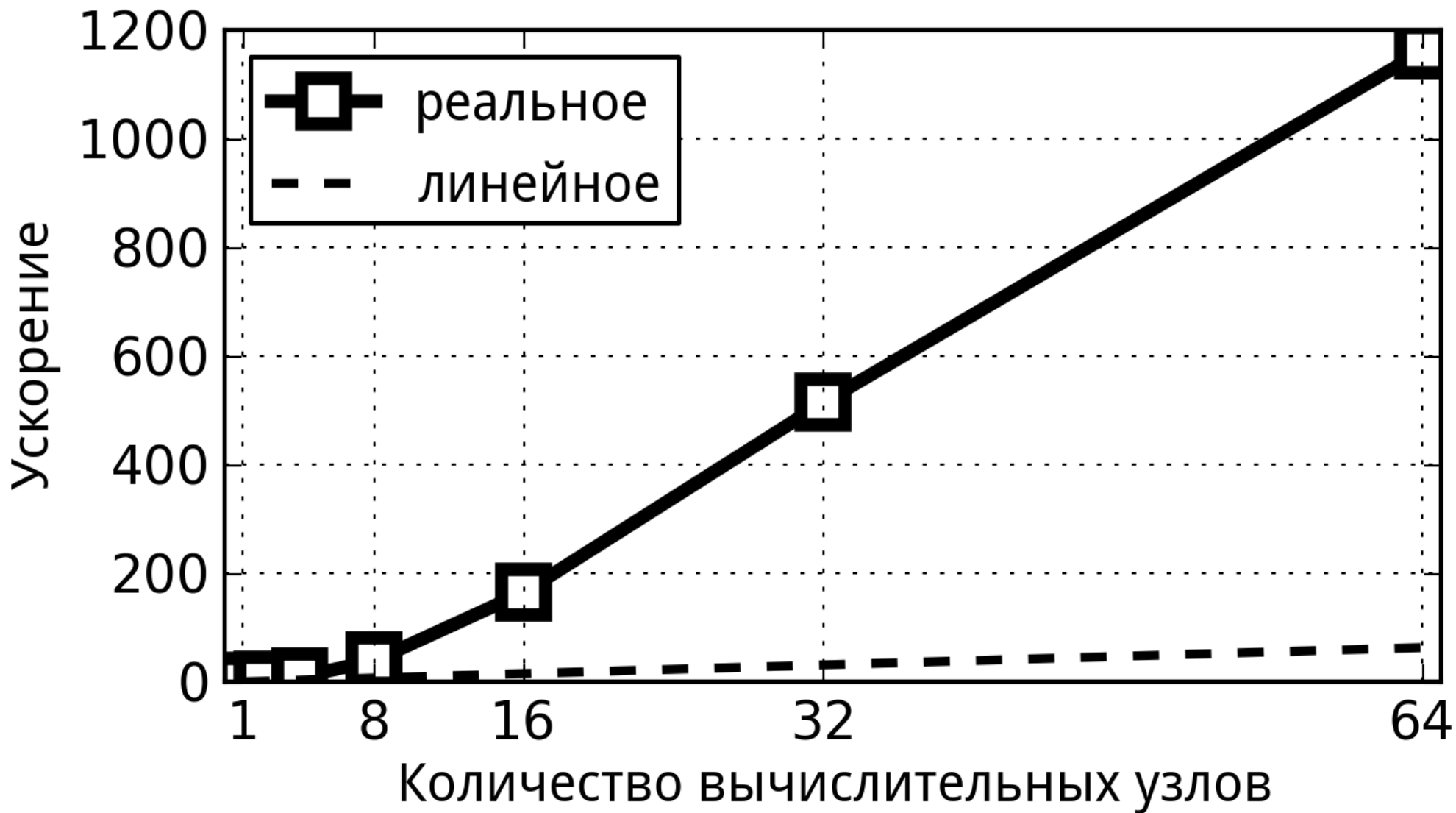
Эксперименты

- Граф
 - карта дорог Люксембурга с OpenStreetMap
 - 114599 вершин
 - 119666 ребер
 - распределение по узлам на основе функции распределения $\varphi(t) = t.A * N_{\text{узлов}} / N_{\text{ребер}}$
- Вычислительная система
 - 64 узла суперкомпьютера Торнадо в ЮУрГУ

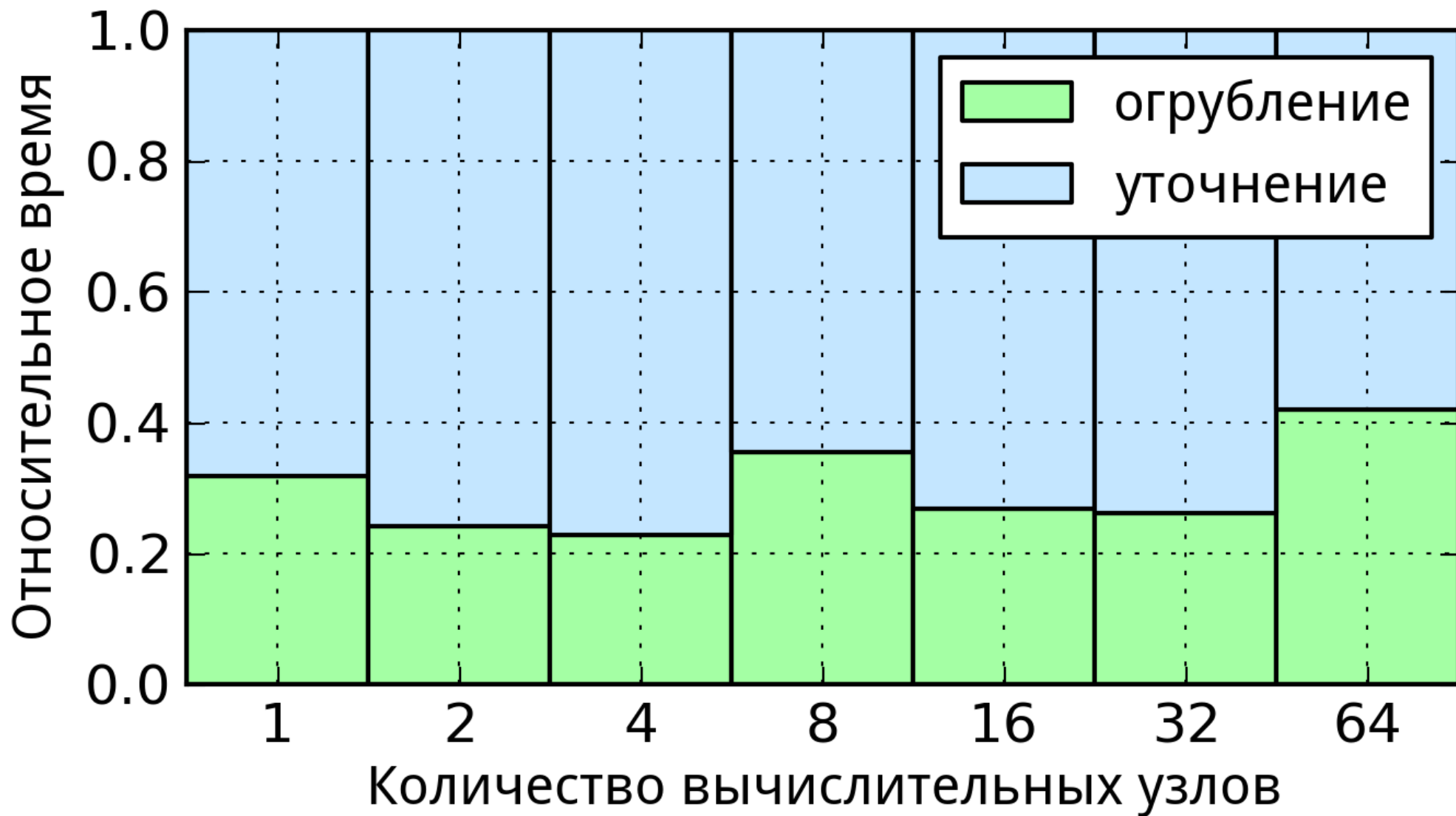
Эксперименты



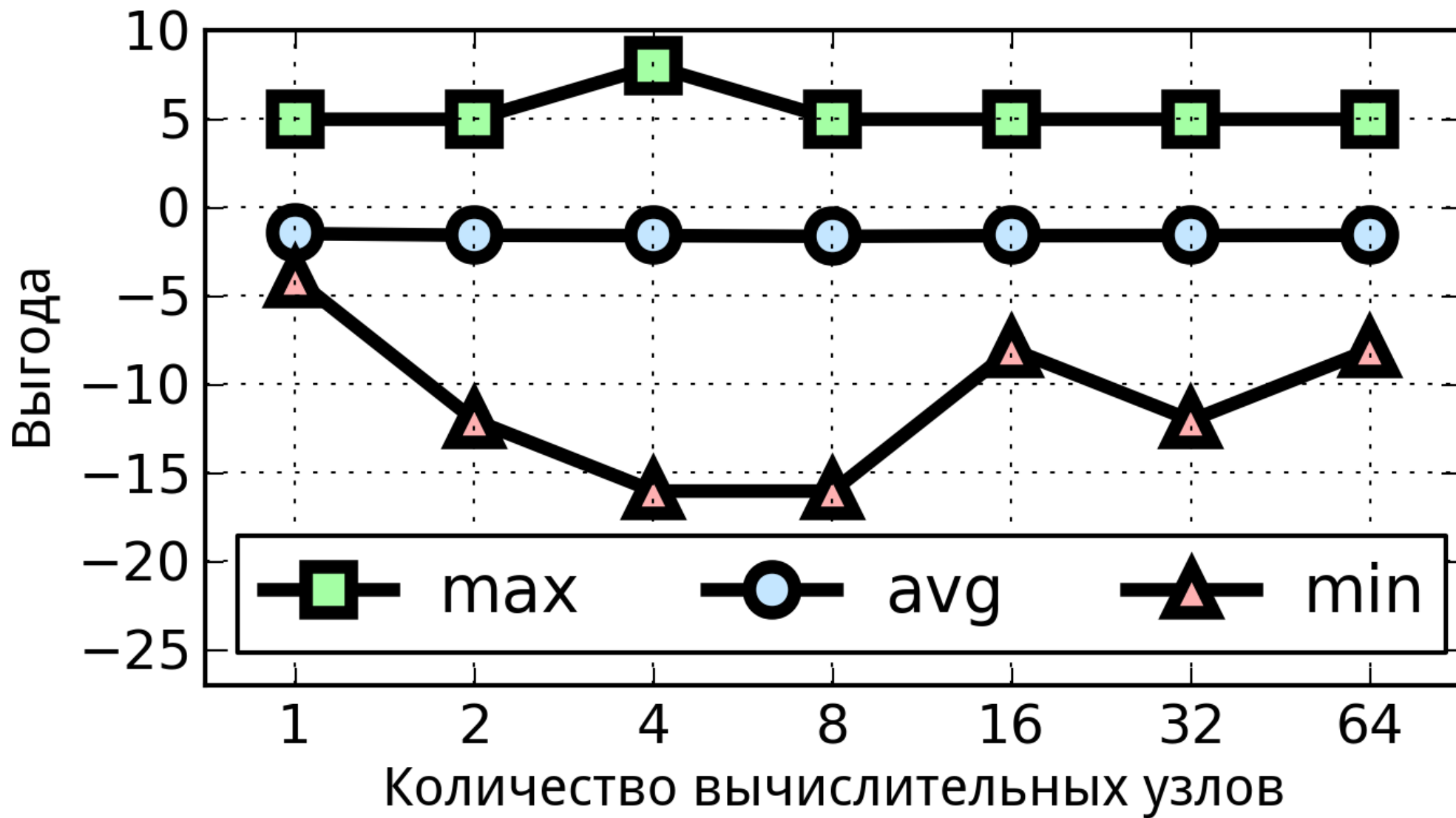
Эксперименты



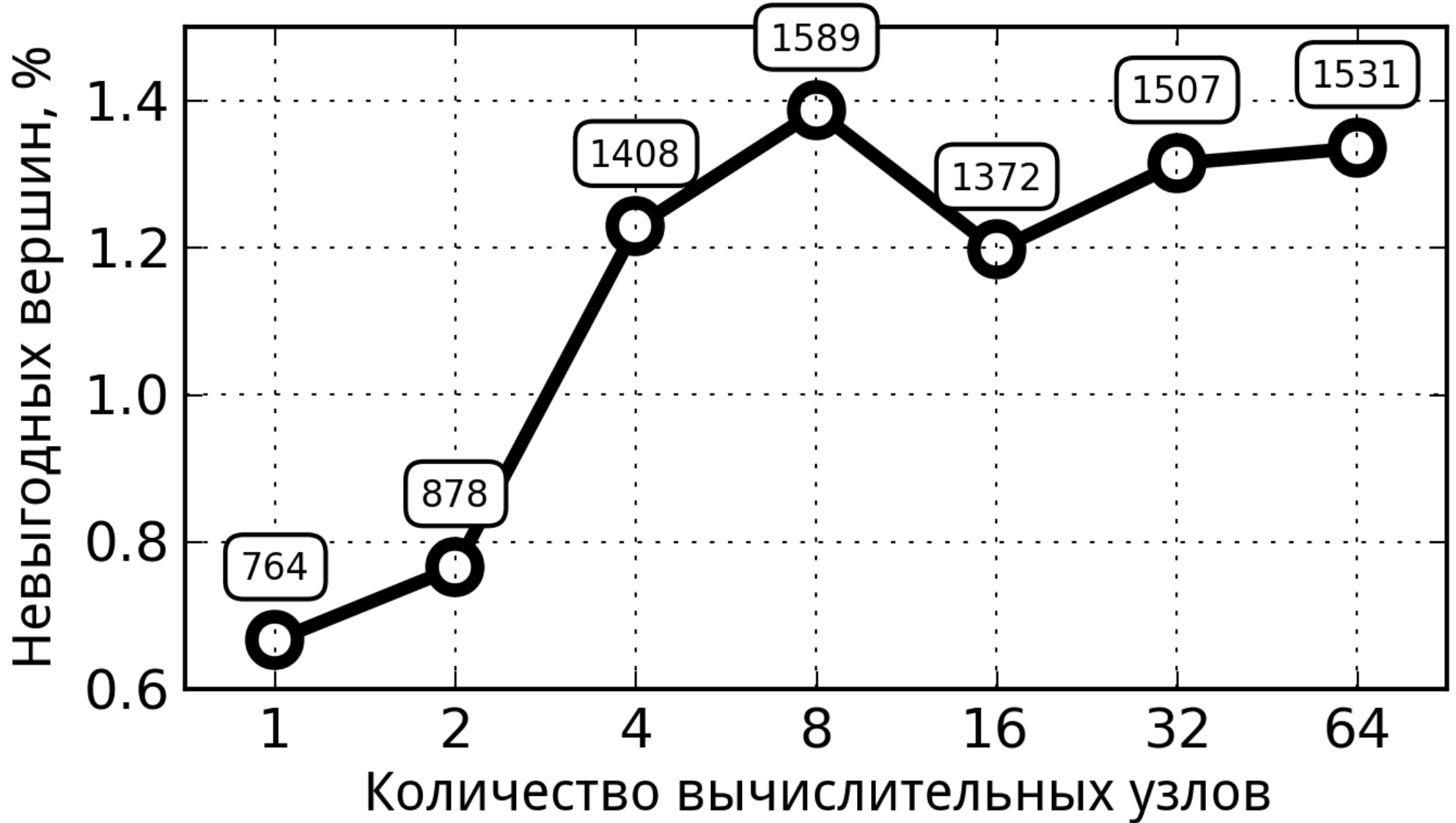
Эксперименты



Эксперименты



Эксперименты



Заключение

- Предложен новый подход к разбиению сверхбольших графов, основанный на использовании реляционной параллельной СУБД PargreSQL
- Проведены вычислительные эксперименты на реальном графе, показавшие высокую масштабируемость подхода при приемлемом качестве разбиения