

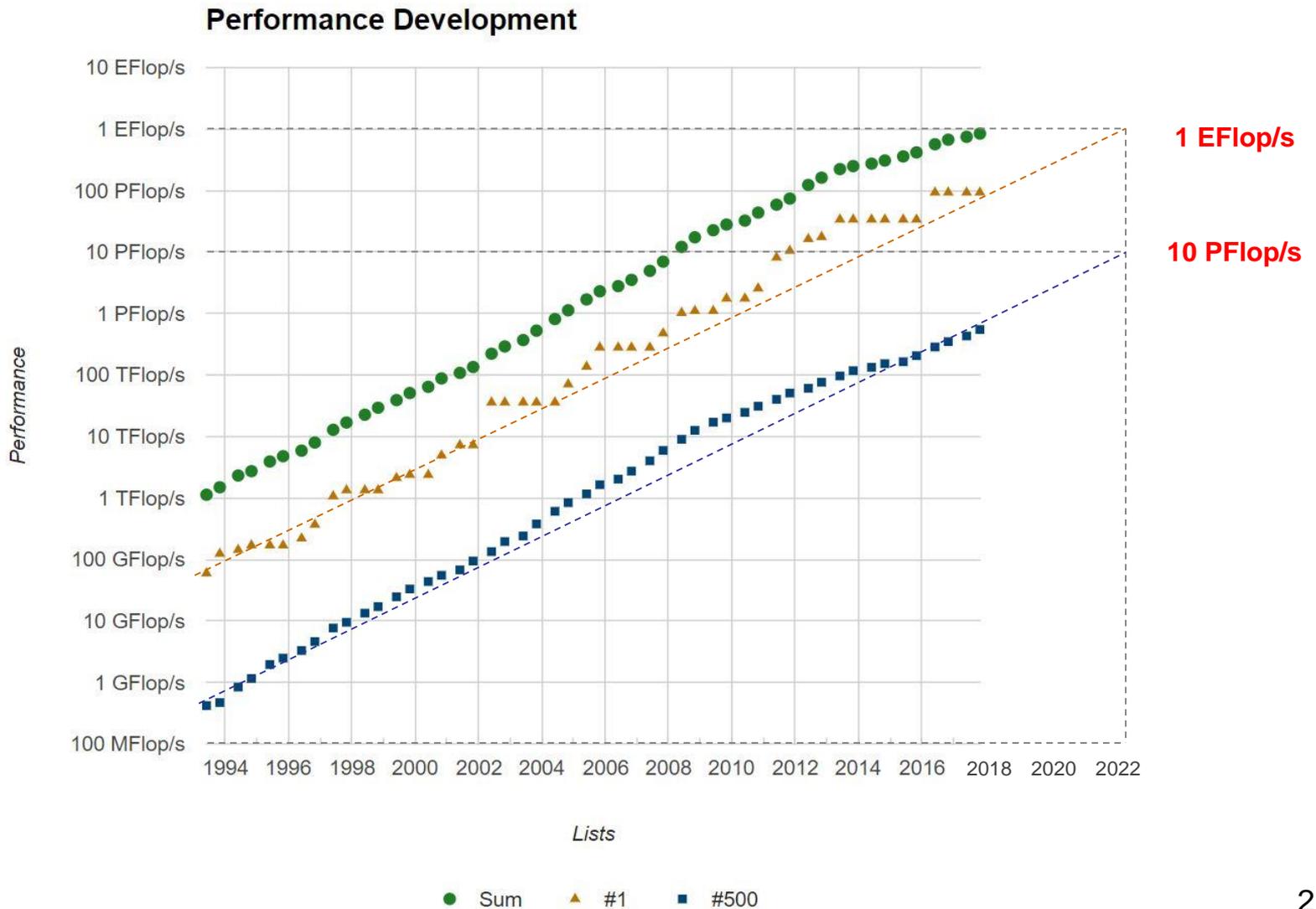


BSF: модель параллельных вычислений для многопроцессорных систем с распределенной памятью

д.ф.-м.н., профессор Л.Б. Соколинский,
аспирант Н.А. Ежова

Южно-Уральский государственный университет
(национальный исследовательский университет)

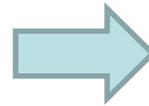
TOP500 (ноябрь 2017)



Изменение основной парадигмы дизайна численных алгоритмов

Разработать алгоритм,
эффективно работающий
на малых вычислительных
ресурсах

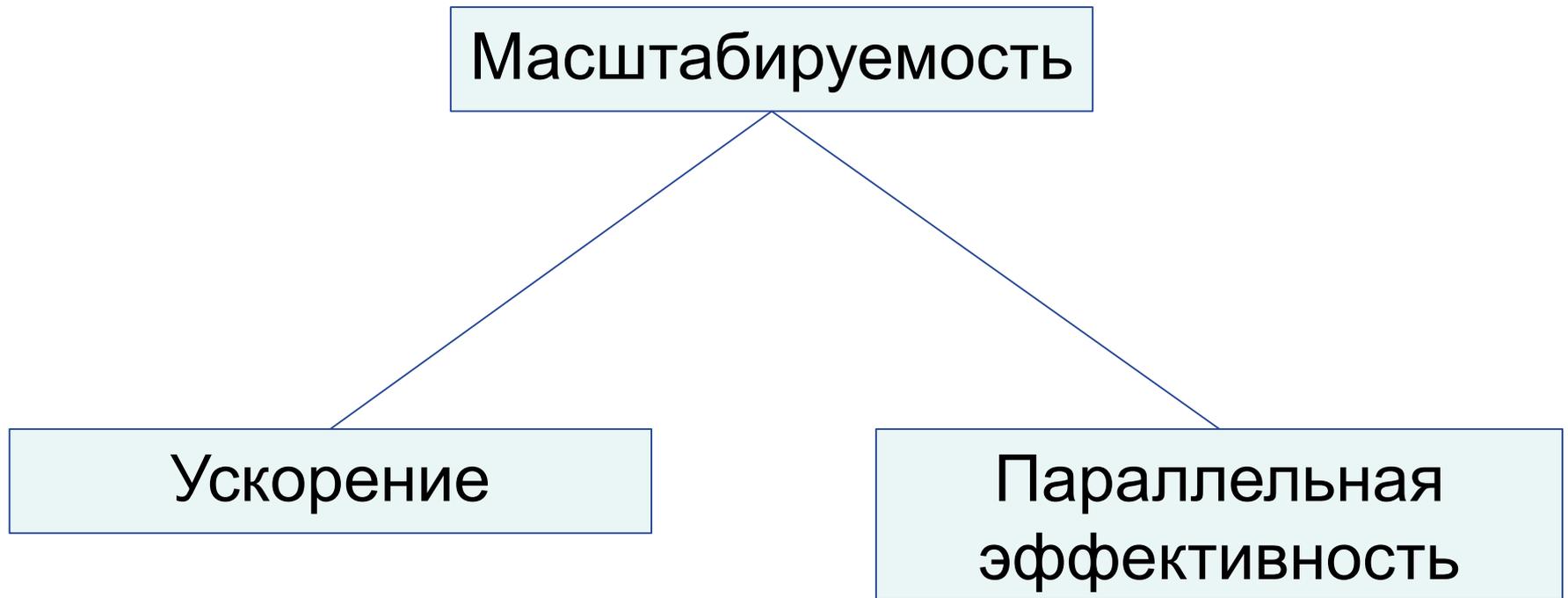
1988



Разработать алгоритм,
способный эффективно
использовать большие
вычислительные ресурсы

2018

Главная характеристика современного параллельного численного алгоритма



Ускорение

$$a(K) = \frac{t_1}{t_K}$$

Количество процессорных узлов

Время решения задачи на 1 узле

Время решения задачи на K узлах

The diagram illustrates the formula for acceleration $a(K)$. It consists of the equation $a(K) = \frac{t_1}{t_K}$ in the center. Three callout boxes with arrows point to the variables: K is labeled 'Количество процессорных узлов' (Number of processor nodes), t_1 is labeled 'Время решения задачи на 1 узле' (Time to solve the problem on 1 node), and t_K is labeled 'Время решения задачи на K узлах' (Time to solve the problem on K nodes).

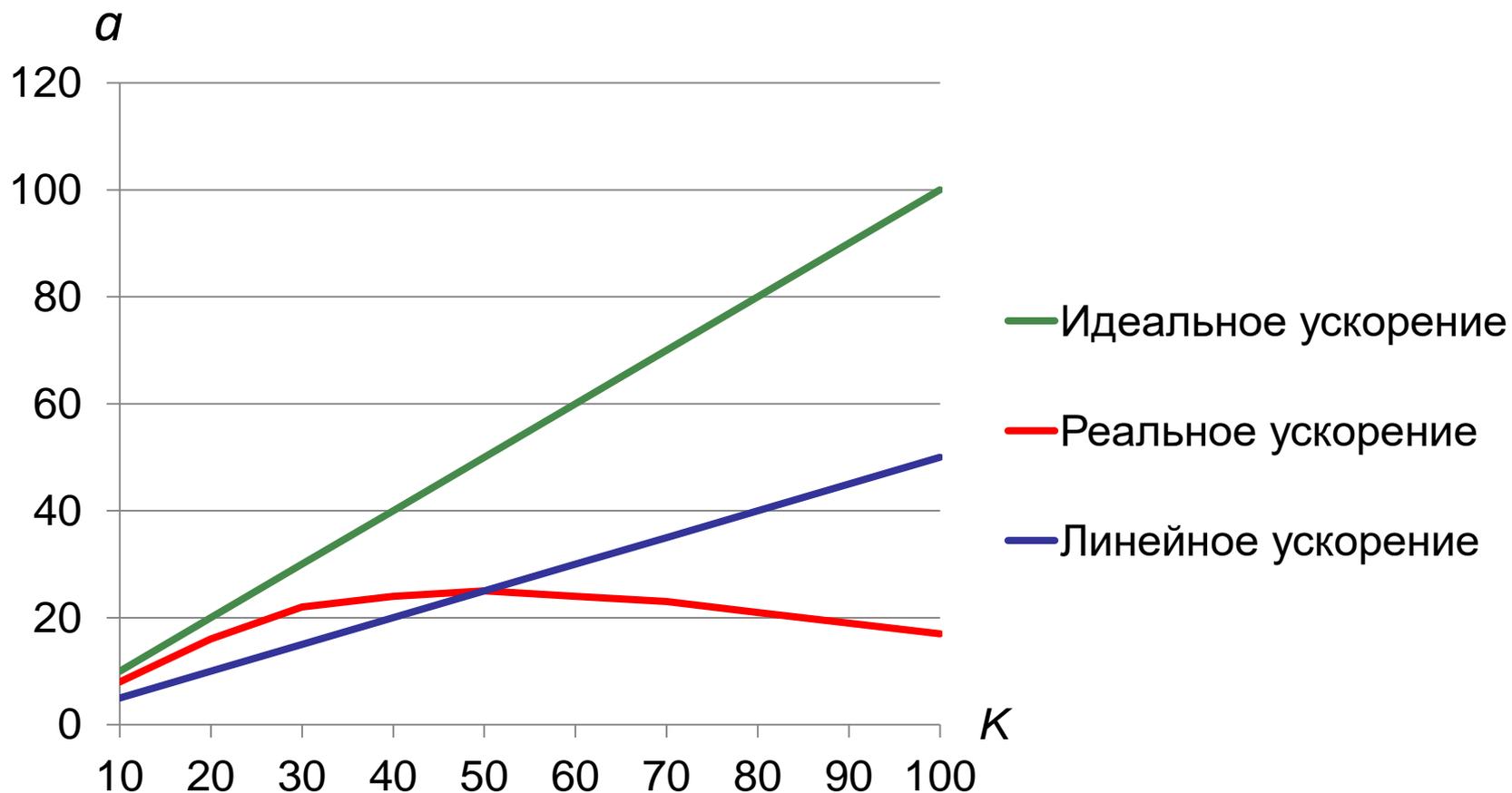
Параллельная эффективность

$$e(K) = \frac{a(K)}{K}$$

Ускорение

Количество процессорных узлов

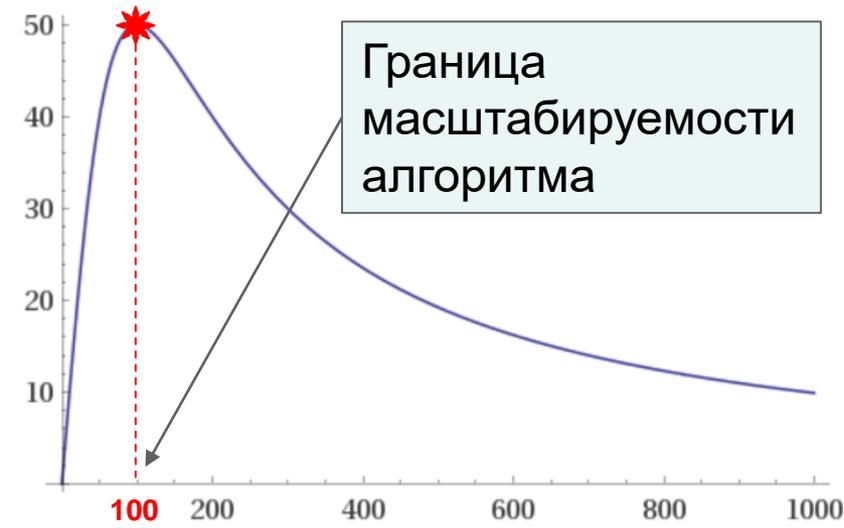
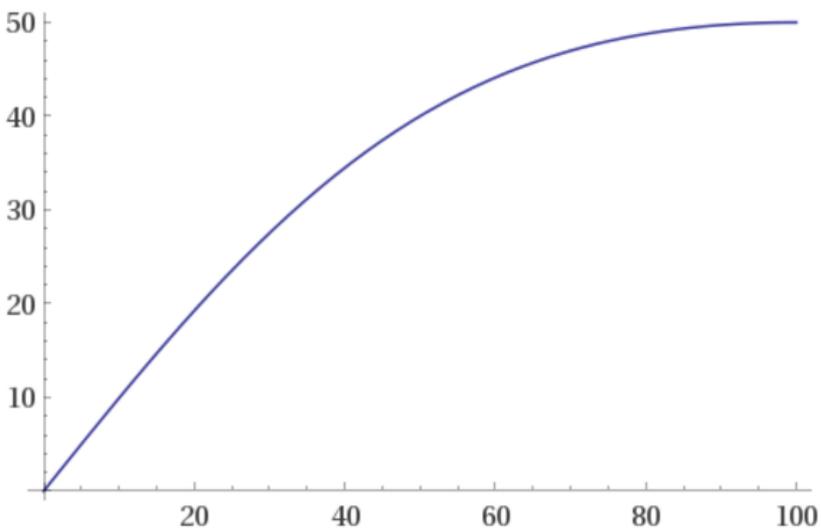
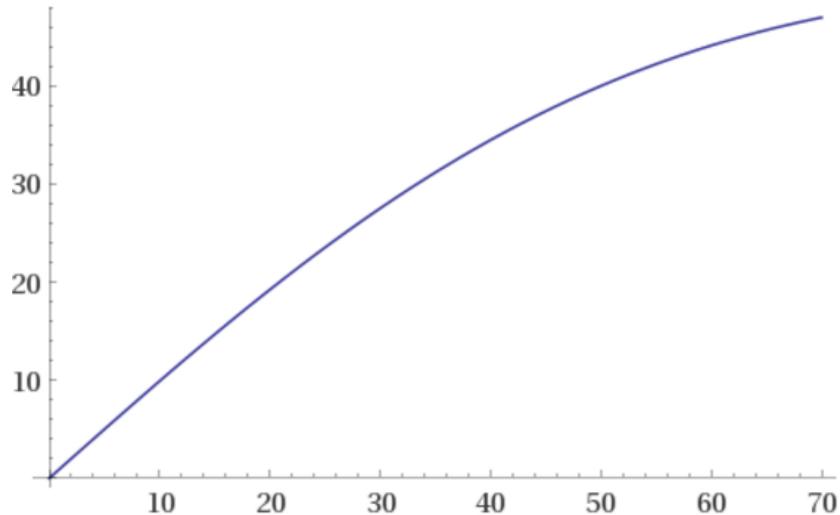
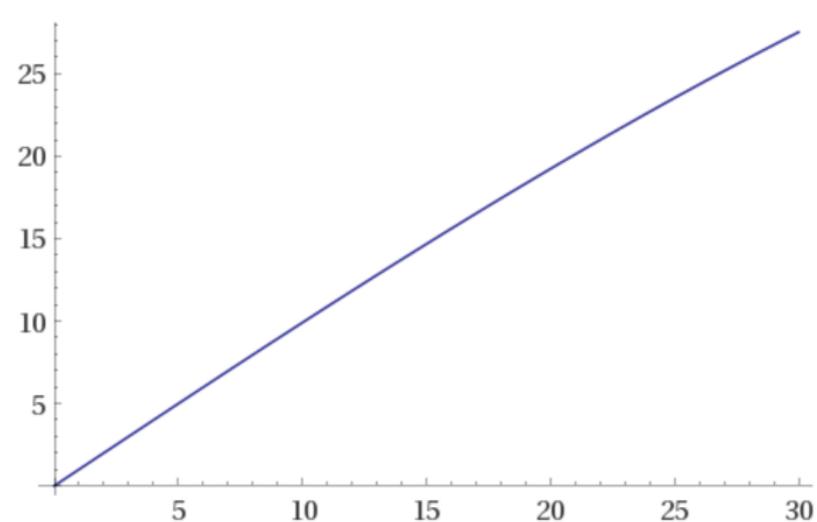
Ускорение: идеальное, линейное, реальное



Алгоритм является хорошо масштабируемым, если:

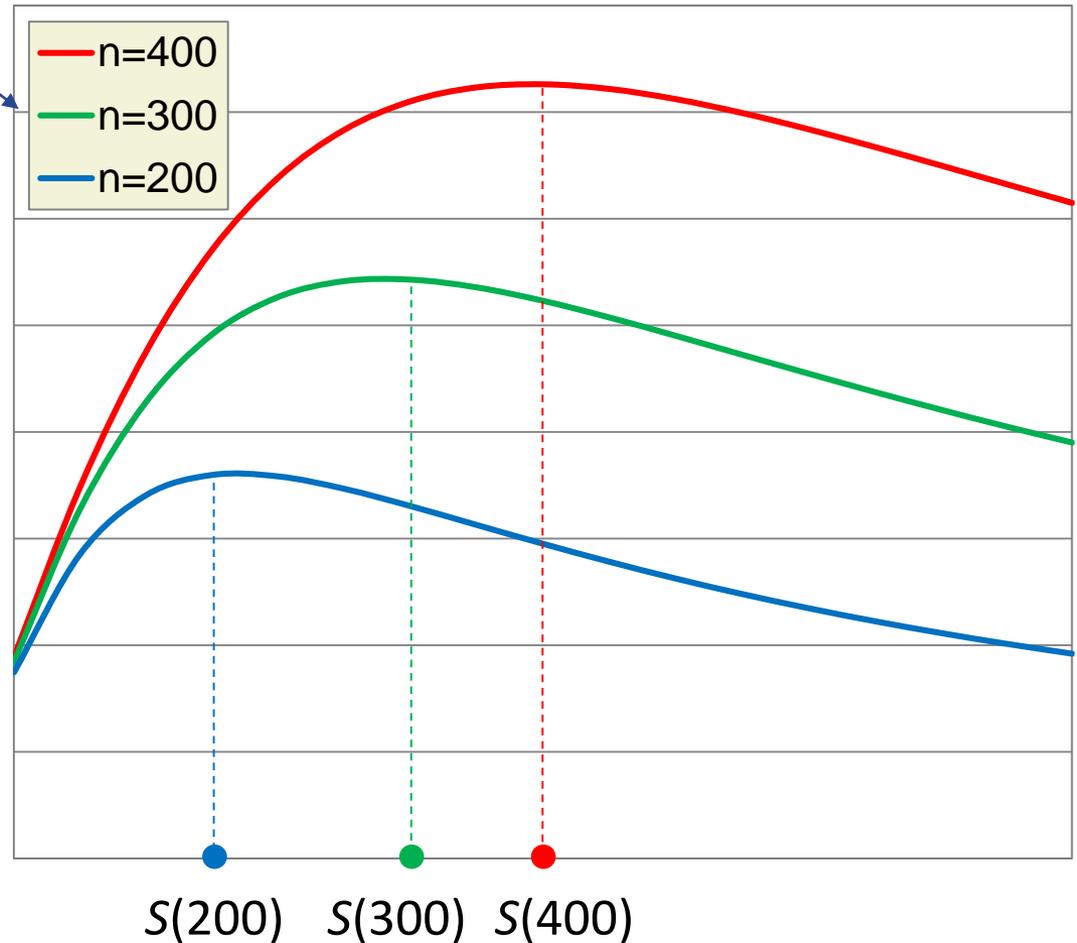
- 1) ускорение близко к линейному
- 2) параллельная эффективность > 0.5

Ускорение реальной задачи на кластерной вычислительной системе



Необходима формула для оценка масштабируемости алгоритма **ДО** написания программы

n - размер задачи



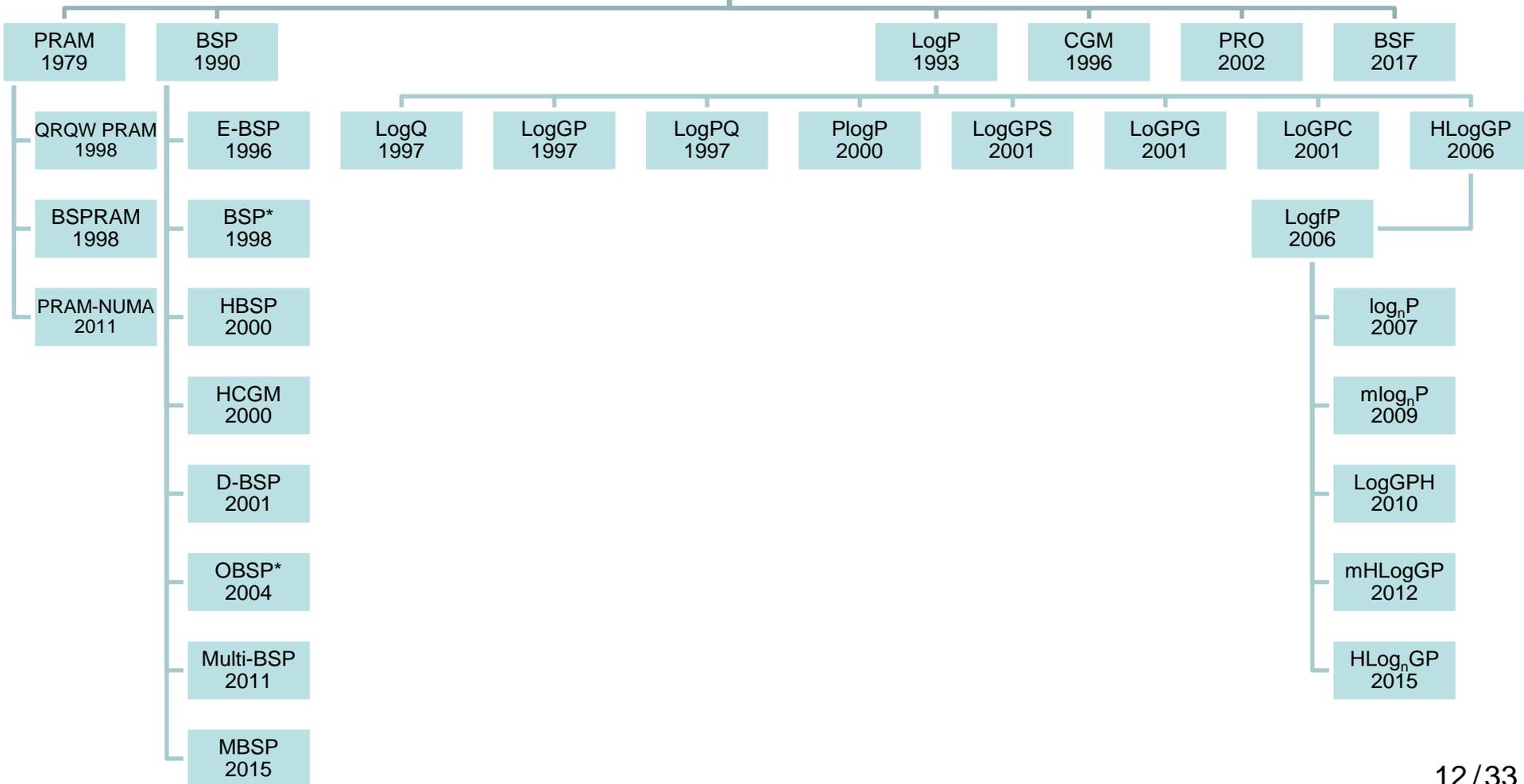
$$K \leq S(n)$$

Модель параллельных вычислений

- *Модель параллельных вычислений* – это фреймворк (система правил и ограничений) для описания и анализа параллельных алгоритмов и программ
- Модель параллельных вычислений позволяет оценить время выполнения параллельной программы (без ее реального выполнения на компьютере)

Дерево моделей параллельных вычислений

Модели параллельных вычислений



Борьба противоположностей в моделях параллельных вычислений

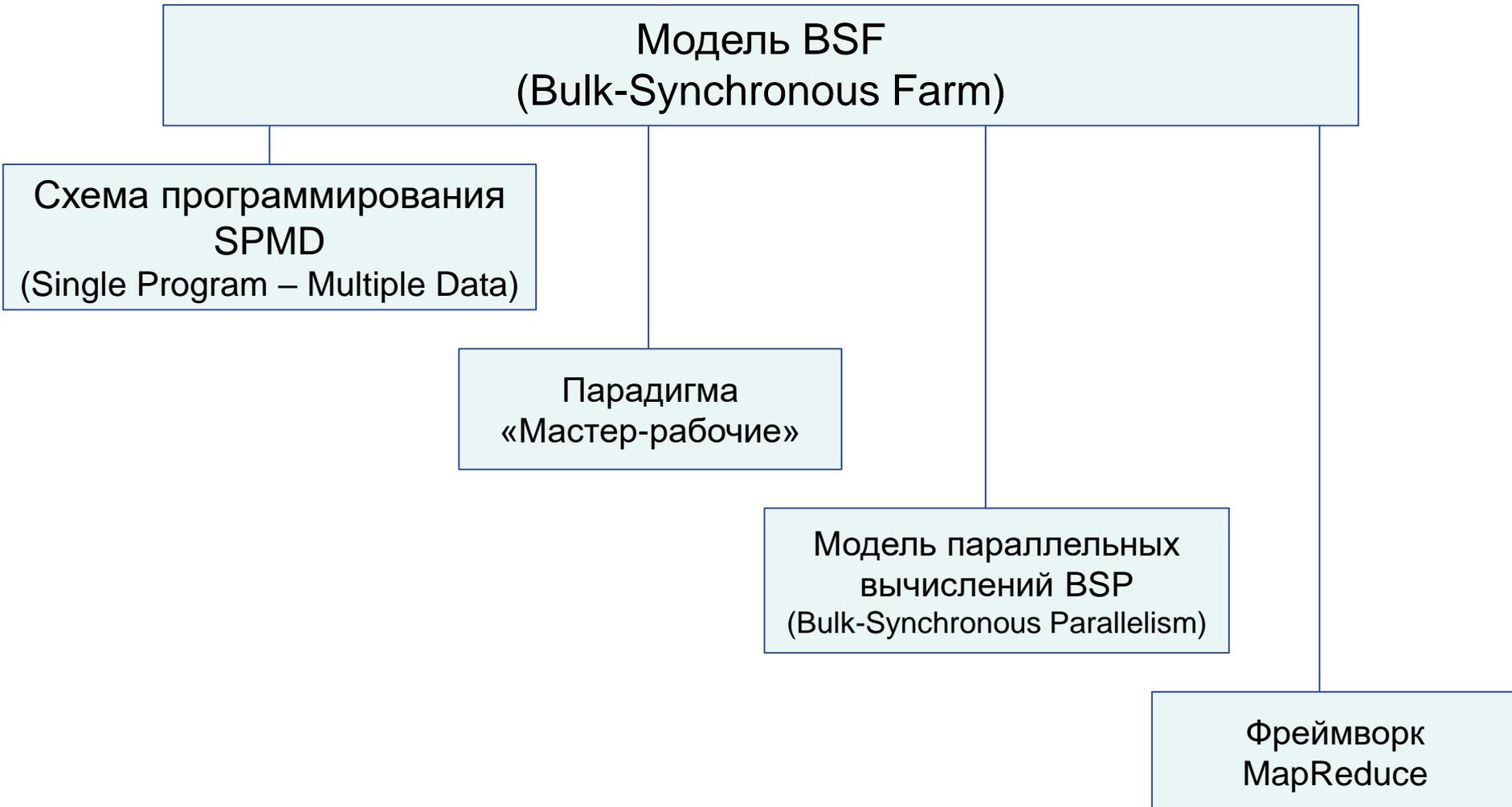
Простота

1. Необходимо сужать проблемную область: для разных областей нужны разные модели
2. Необходимо накладывать более жесткие ограничения на структуру параллельной программы

Универсальность

Точность

Фундамент модели BSF



Область применения модели BSF

- Многопроцессорные системы с распределенной памятью
- Параллельные итерационные алгоритмы с высокой вычислительной сложностью

Модель BSF позволяет предсказать:

- границу масштабируемости параллельного алгоритма
- ускорение алгоритма
- параллельную эффективность алгоритма

BSF-компьютер

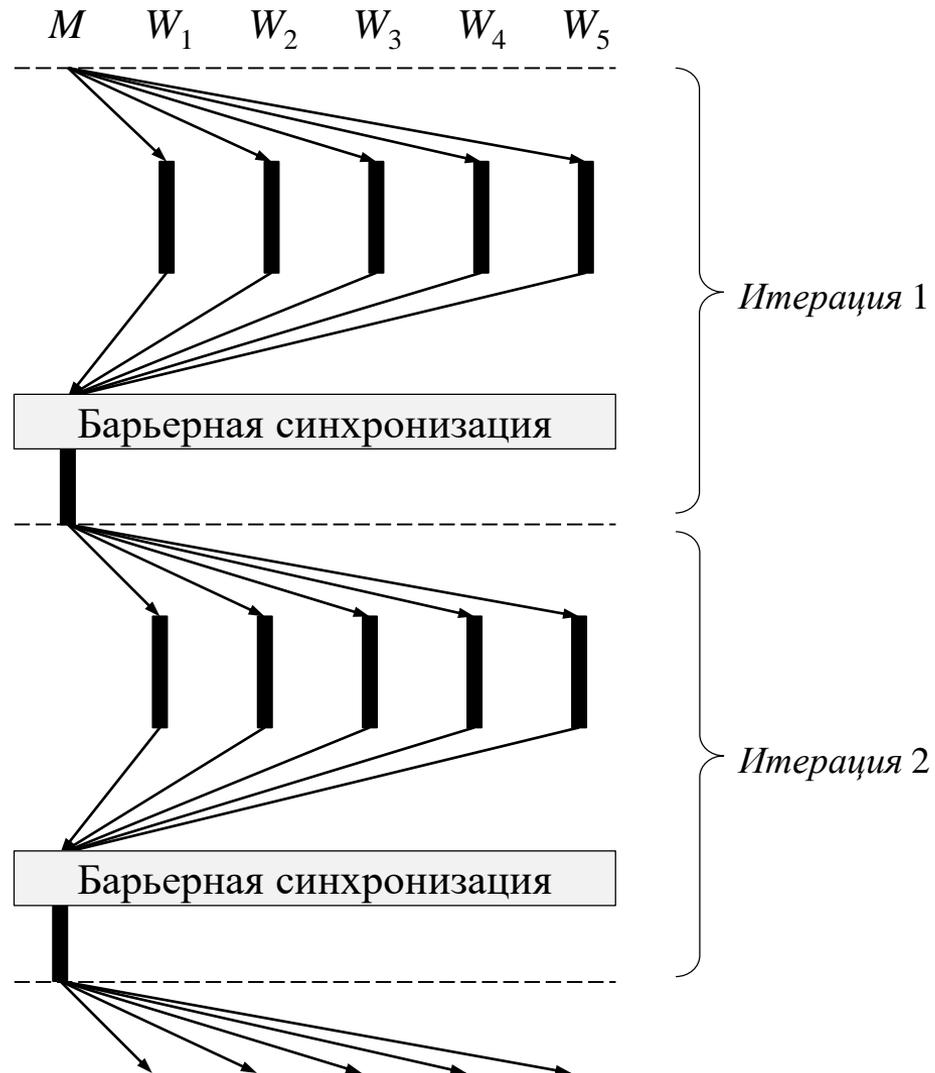


Процессорные узлы

Структура BSF-программы



Схема работы *BSF*-алгоритма

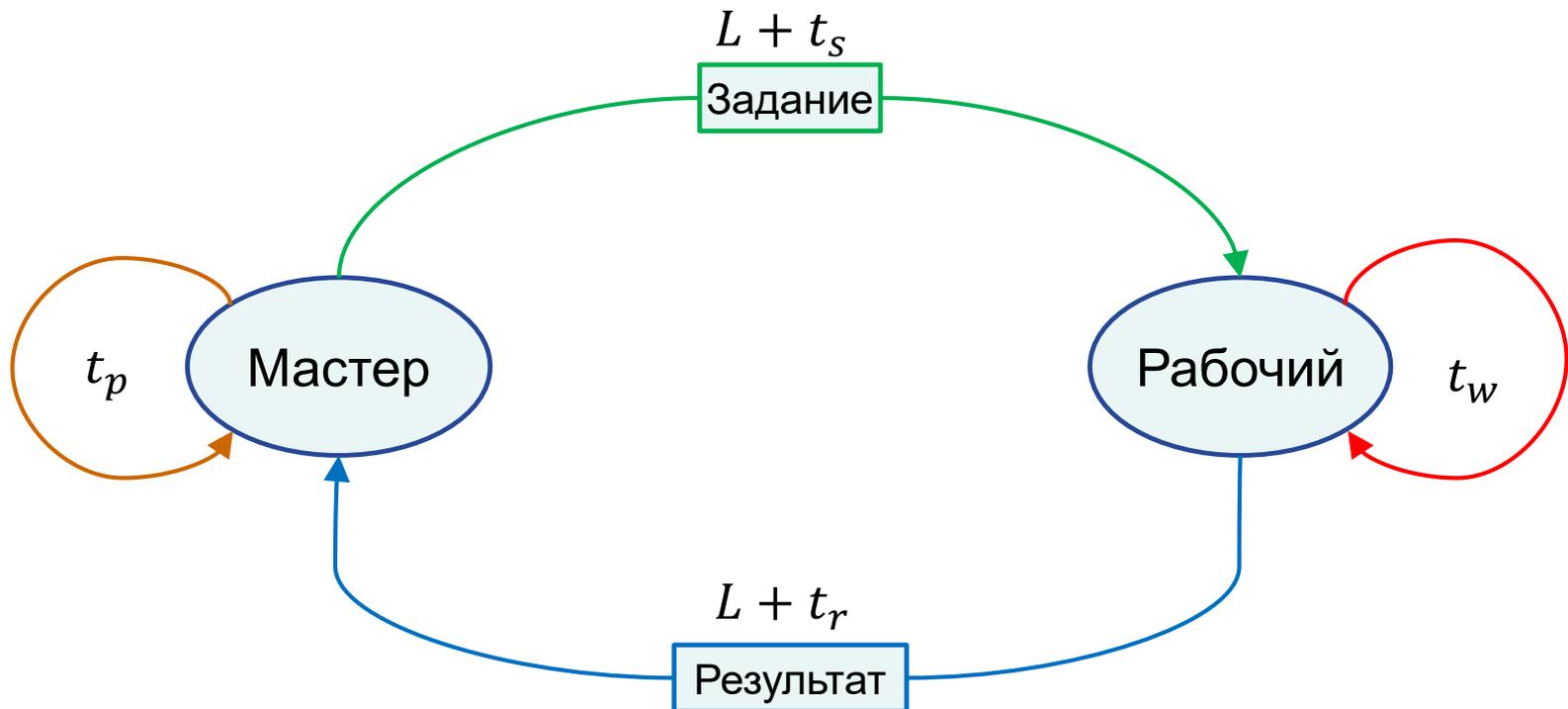


Параметры BSF-модели

- t_s – время, необходимое для посылки сообщения одному рабочему
- L – латентность (время посылки сообщения длиной в 1 байт)
- t_w – суммарное время всех вычислений, выполняемых рабочими в рамках одной итерации
- t_r – время, необходимое для передачи результатов мастеру от рабочих
- t_a – время, необходимое для сложения двух результатов
- t_p – время проверки условия завершения

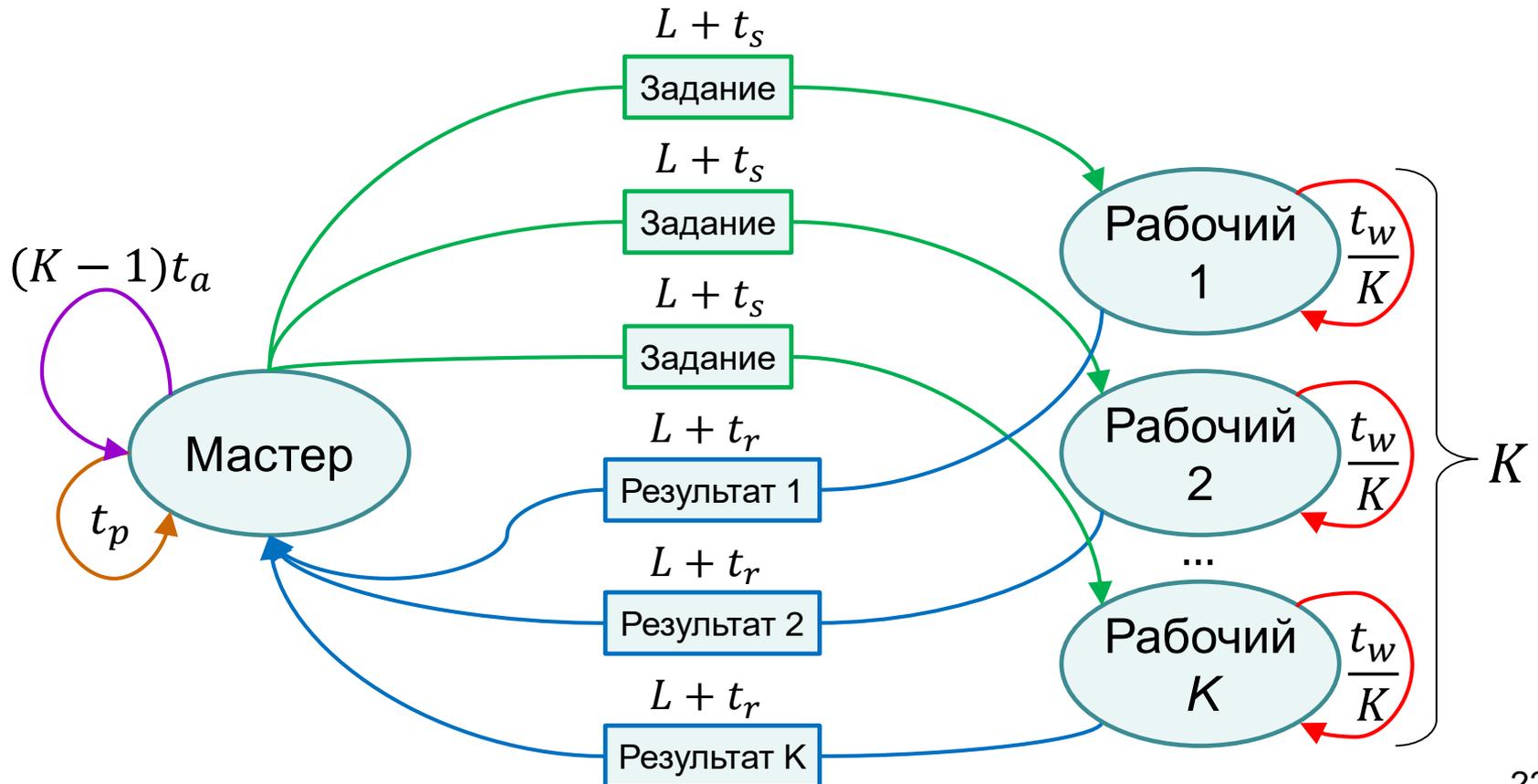
Время решения задачи системой из мастера и одного рабочего

$$T_1 = L + t_s + t_w + L + t_r + t_p$$



Время решения задачи системой из одного мастера и K рабочих

$$T_K = K(L + t_s) + t_w/K + K(L + t_r) + (K - 1)t_a + Kt_p$$



Ускорение BSF-алгоритма

$$a(K) = \frac{T_1}{T_K} = \frac{2L + t_s + t_r + t_p + t_w}{K(2L + t_s + t_r + t_a) + t_p - t_a + t_w/K}$$

Параллельная эффективность BSF-алгоритма

$$e(K) = \frac{a(K)}{K} \approx \frac{1}{1 + (K^2(2L + t_s + t_r + t_a) + Kt_p)/t_w}$$

Вычисление границы масштабируемости BSF-алгоритма

Найдем корни уравнения $a'(K) = 0$

$$\frac{(2L + t_s + t_r + t_p + t_w)(t_w/K^2 - 2L - t_s - t_r - t_a)}{(K(2L + t_s + t_r + t_a) - t_a + t_p + t_w/K)^2} = 0$$

$$t_w/K^2 - 2L - t_s - t_r - t_a = 0$$

Параметры BSF-модели

- t_s – время, необходимое для отправки сообщения одному рабочему
- L – латентность (время отправки сообщения длиной в 1 байт)
- t_w – суммарное время всех вычислений, выполняемых рабочими в рамках одной итерации
- t_r – время, необходимое для передачи результатов мастеру от рабочих
- t_a – время, необходимое для сложения двух результатов
- t_p – время проверки условия завершения

20/33

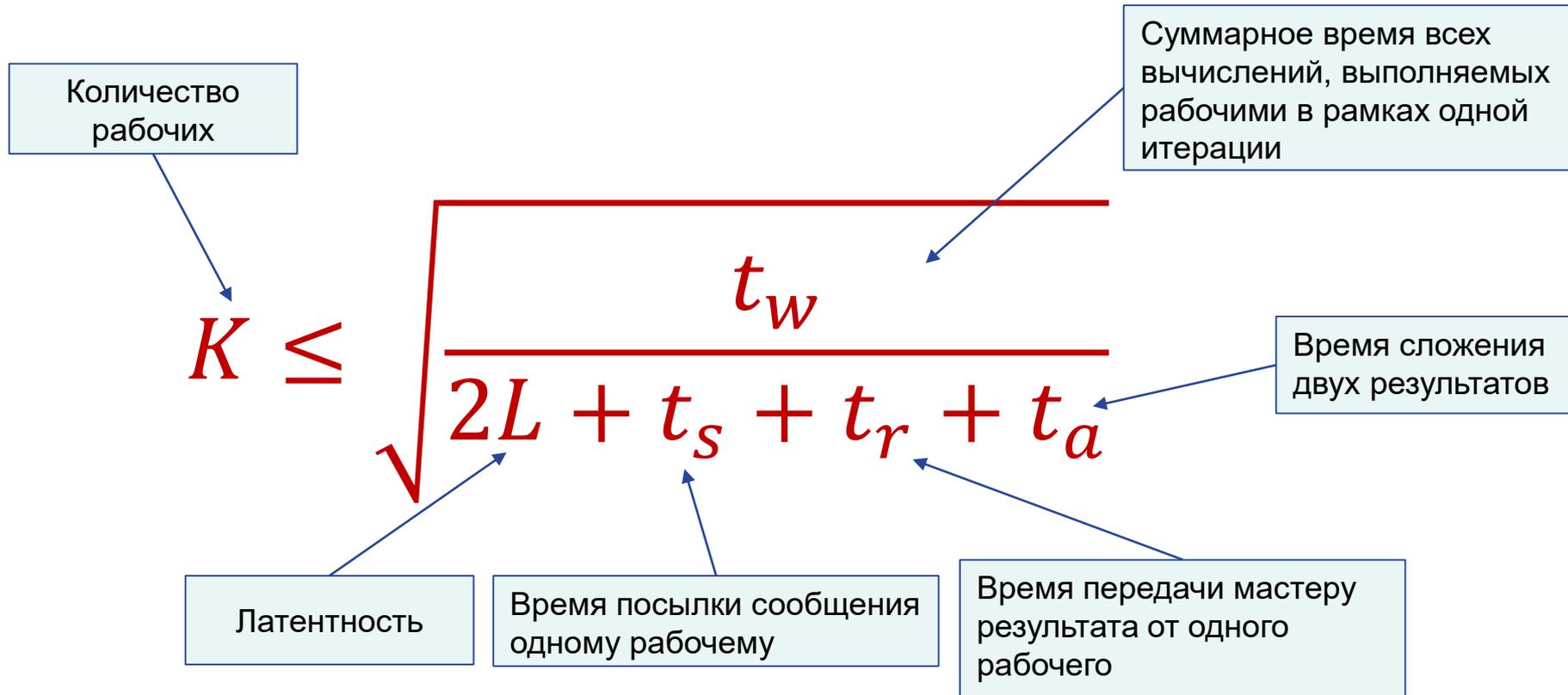
$$K = \sqrt{\frac{t_w}{2L + t_s + t_r + t_a}}$$

Ускорение BSF-алгоритма

$$a(K) = \frac{T_1}{T_K} = \frac{2L + t_s + t_r + t_p + t_w}{K(2L + t_s + t_r + t_a) + t_p - t_a + t_w/K}$$

23/33

Граница масштабируемости BSF-алгоритма



Граница масштабируемости не зависит от:

t_p – время проверки условия завершения

Пример применение модели BSF для исследования алгоритма NSLP

Нестационарная задача линейного программирования

$$\max\{\langle c_t, x \rangle \mid A_t x \leq b_t, x \geq 0\}$$

- $x \in \mathbb{R}_n$
- A_t – матрица $m \times n$
- c_t, b_t – векторы размерности n
- $t \in \mathbb{R}_{\geq 0}$ – время

Граница масштабируемости BSF-реализации алгоритма NSLP

В каждой итерации меняются все элементы матрицы A :

$$K \leq O(\sqrt{n})$$

В каждой итерации меняется один элемент матрицы A :

$$K \leq O(n)$$

K – количество рабочих узлов

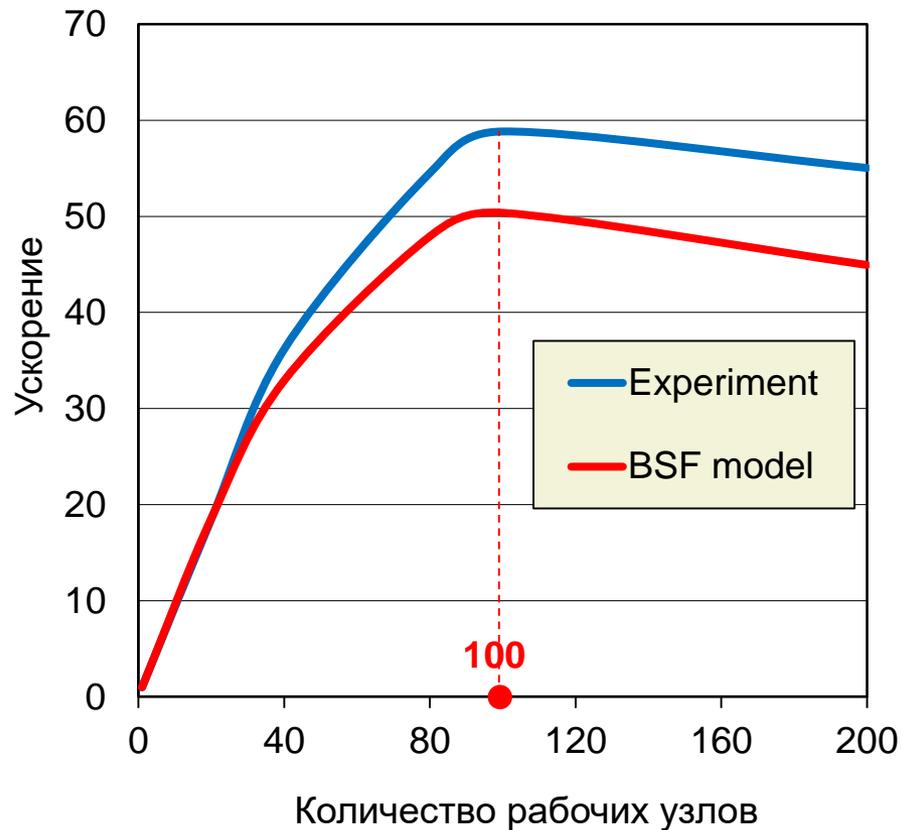
n – размерность задачи

BSF-реализация алгоритма NSLP на C++ и MPI

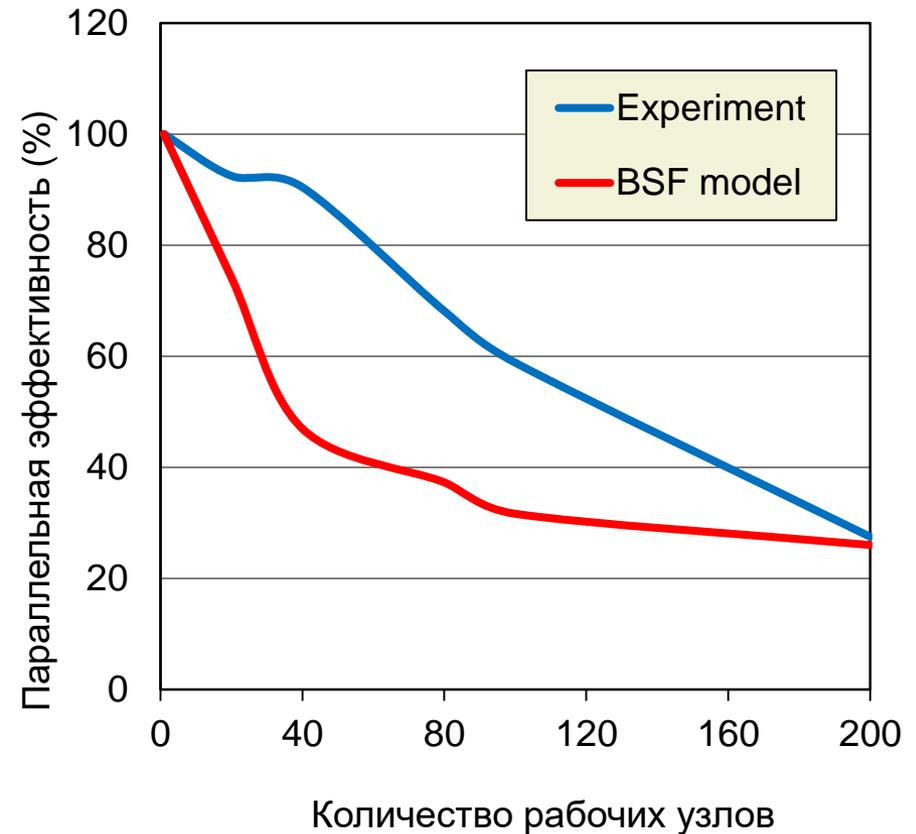
<https://github.com/leonid-sokolinsky/BSF-NSLP>

Сравнение теории с экспериментом: $n = 400$

Ускорение

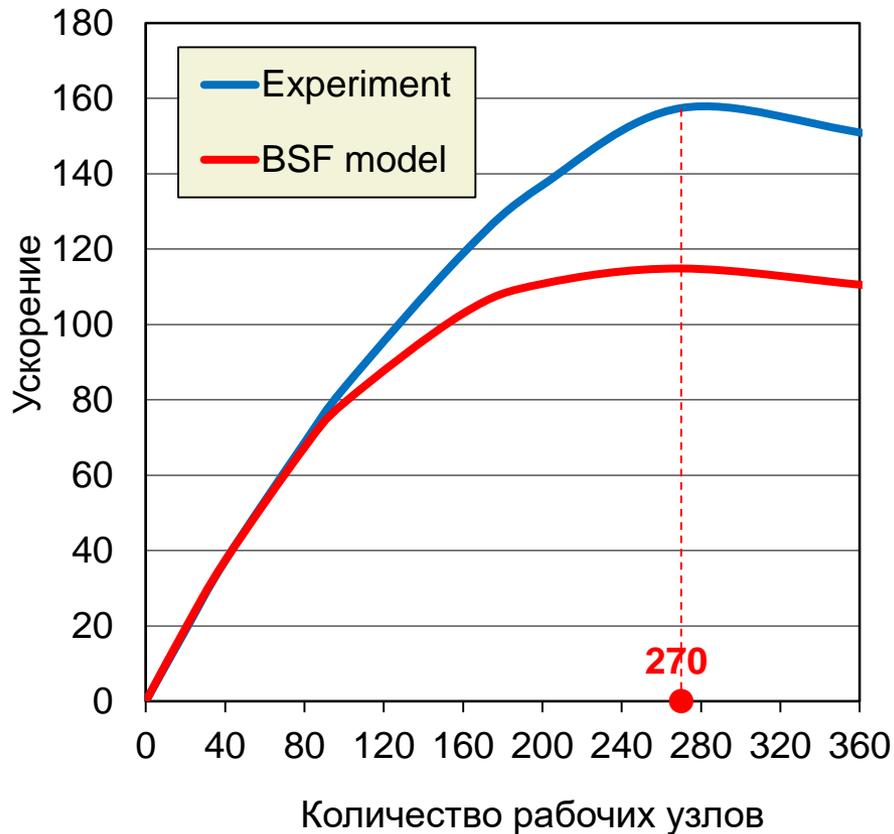


Эффективность

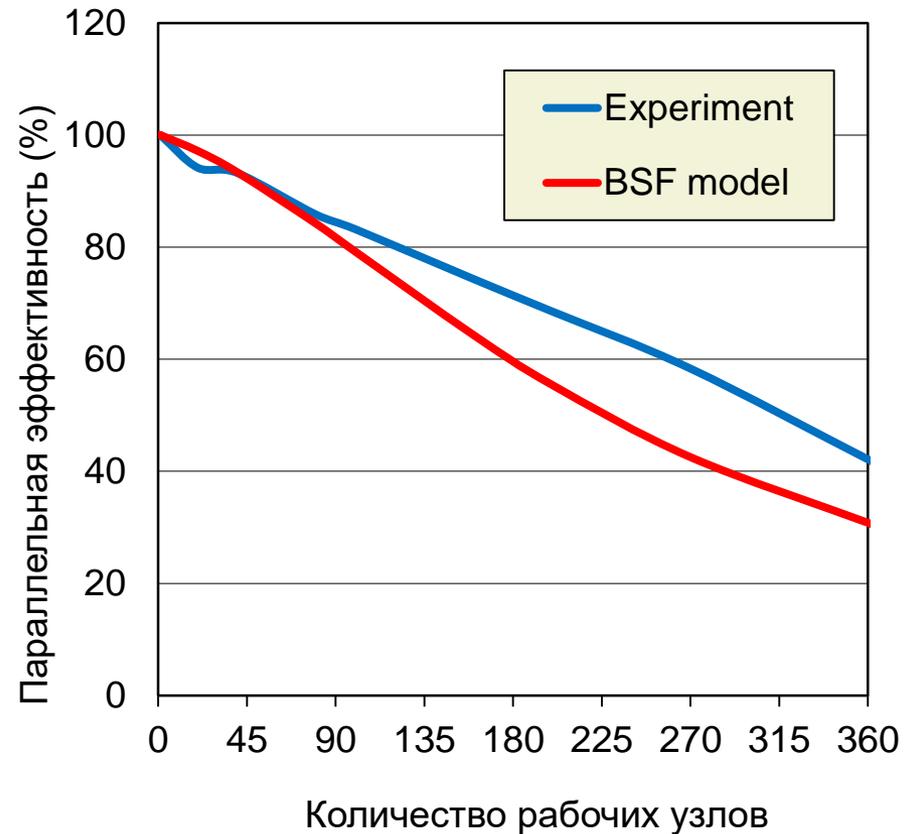


Сравнение теории с экспериментом: $n = 800$

Ускорение

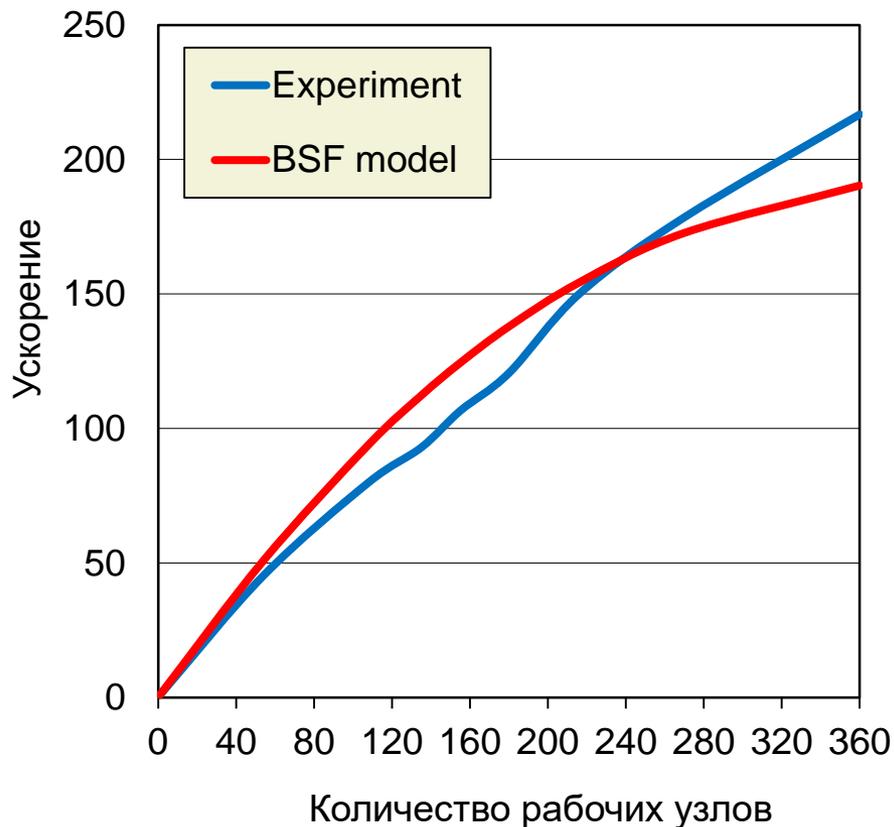


Эффективность

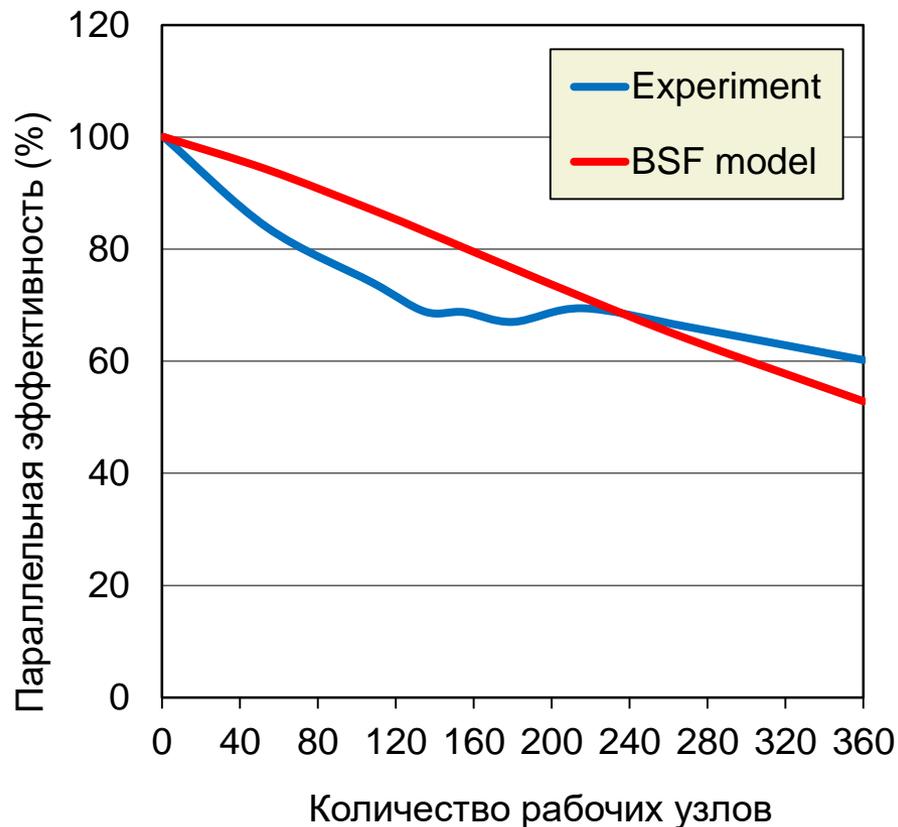


Сравнение теории с экспериментом: $n = 1080$

Ускорение



Эффективность

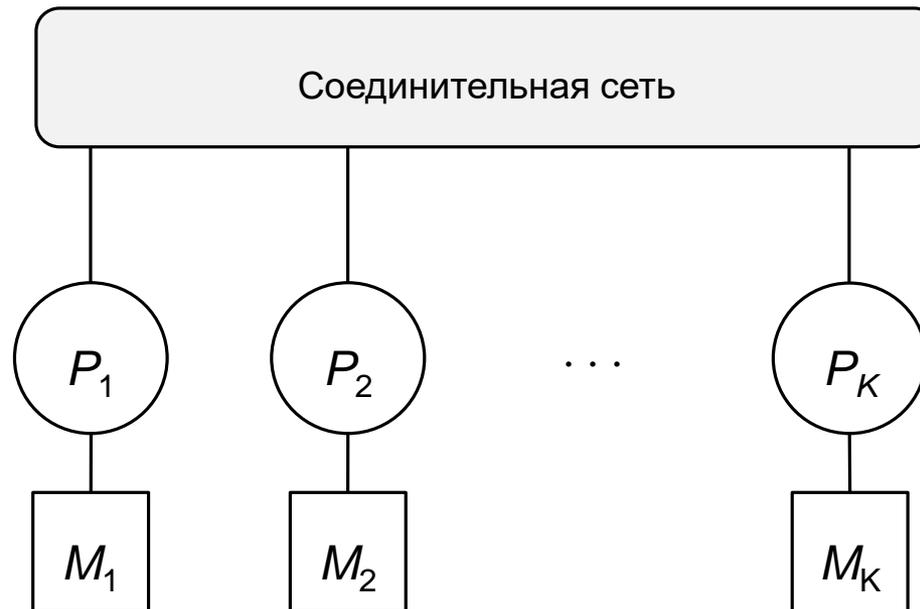


Спасибо за внимание!

Вопросы?

Дополнительные слайды

BSP-компьютер



Параметры BSP-модели

g – время, необходимое для передачи по сети одного машинного слова

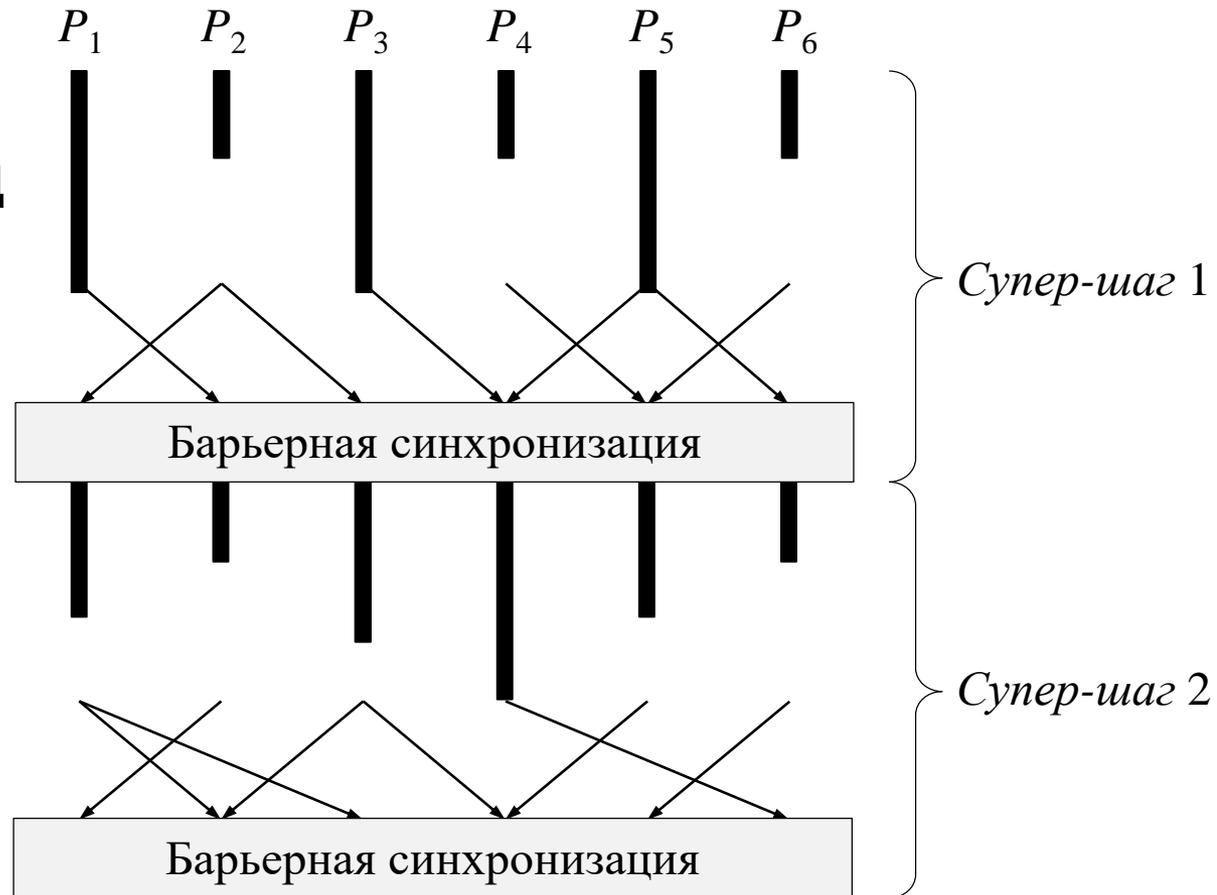
L – время, необходимое для выполнения глобальной синхронизации

h -сессия: каждый процессор передает не более h машинных слов и получает не более h машинных слов

$h \cdot g$ – время выполнения h -сессии

Схема выполнения BSP-программы

- *BSP-программа* состоит из n потоков команд
- Каждый поток команд делится на последовательные *супершаги*
- Супершаг:
 - 1) Вычисления на каждом процессоре с использованием только локальных данных
 - 2) h-сессия
 - 3) Глобальная барьерная синхронизация



Стоимостная функция модели BSP

- Время измеряется в машинных тактах
- BSP-программа состоит из S супершагов
- Каждый процессор на i -том супершаге выполняет максимум w_i тактов в ходе локальных вычислений
- Время выполнения i -того супершага:

$$t_i = w_i + h \cdot g + L$$

- Время выполнения всей программы:

$$T = h \cdot g \cdot S + L \cdot S + \sum_{i=1}^S w_i$$

Деление BSF-программы на секции мастера и рабочего

```
// К - число рабочих
// Номер мастера: К
// Номера рабочих: 0..(К-1)
input
    MyNodeNo // номер моего узла
begin
    if MyNodeNo = К then
        // Секции мастера
        ...
    else
        // Секции рабочего
        ...
    end if
end
```

Секция рабочего «Выполнение задания»

```
// Суммирование элементов массива M
input
  K // число рабочих
  M[0..n*K-1] // массив для суммирования
  MyNodeNo // номер моего узла
output
  S[0..K-1] // массив частичных сумм
begin
  S[MyNodeNo] := 0
  for i = 0 + MyNodeNo*n to (MyNodeNo+1)*n-1 do
    S[MyNodeNo] := S[MyNodeNo] + M[i]
  end for
end
```