



Модель параллельных вычислений для оценки масштабируемости итерационных численных алгоритмов

д.ф.-м.н., профессор Л.Б. Соколинский

Южно-Уральский государственный университет
(национальный исследовательский университет)
Челябинск

Цикл исследования масштабируемости алгоритма

Разработка алгоритма

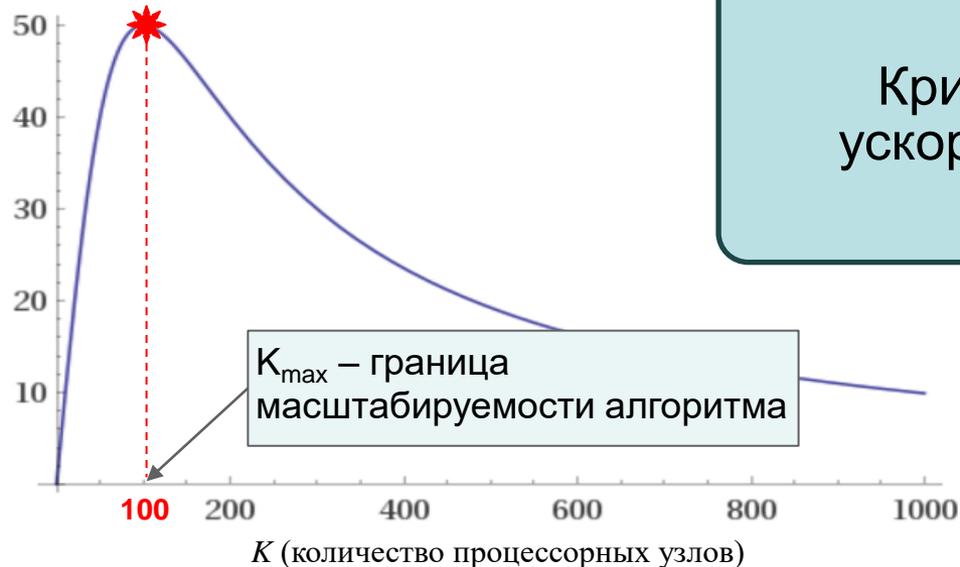
Кодирование

Тестирование

Кривая ускорения

Вычислительные эксперименты

a (ускорение)

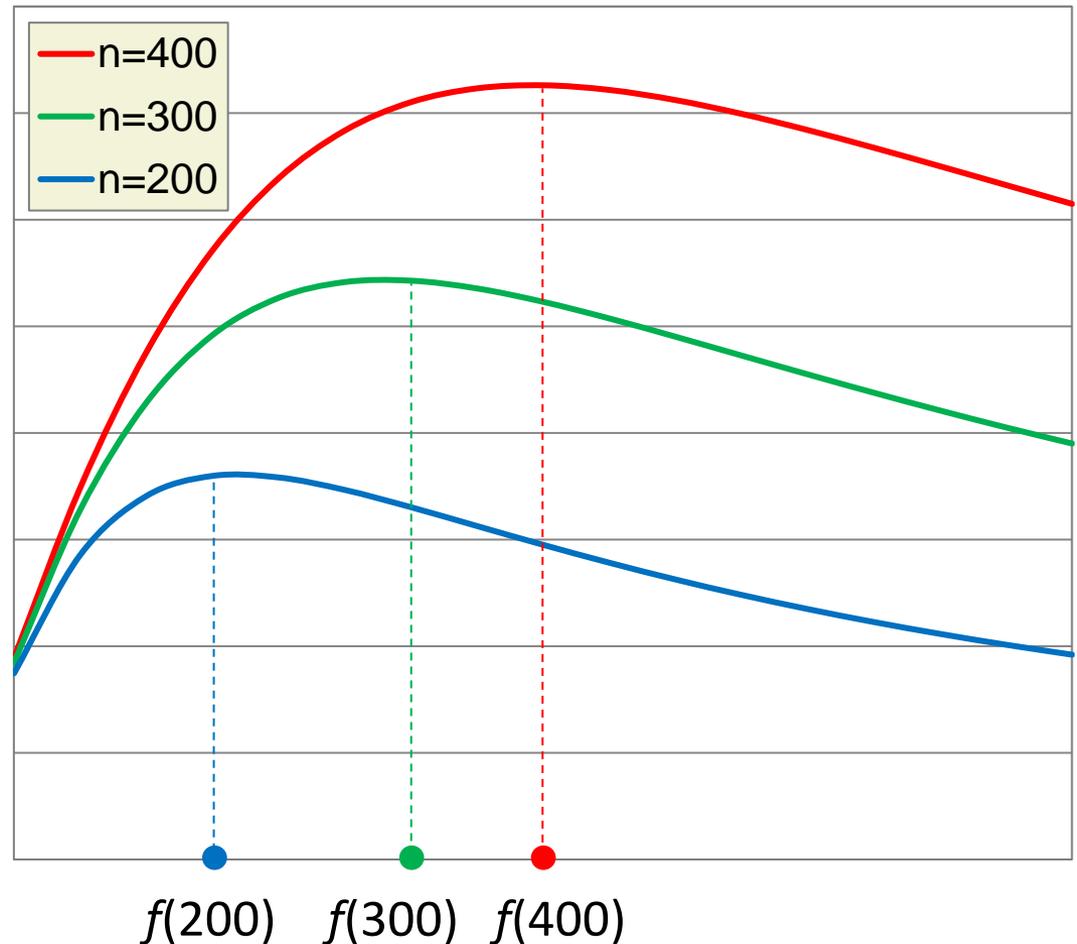


Желание

Оценить границу
масштабируемости алгоритма
ДО написания программы **БЕЗ**
вычислительных экспериментов

Масштабируемость зависит от размера задачи n

a

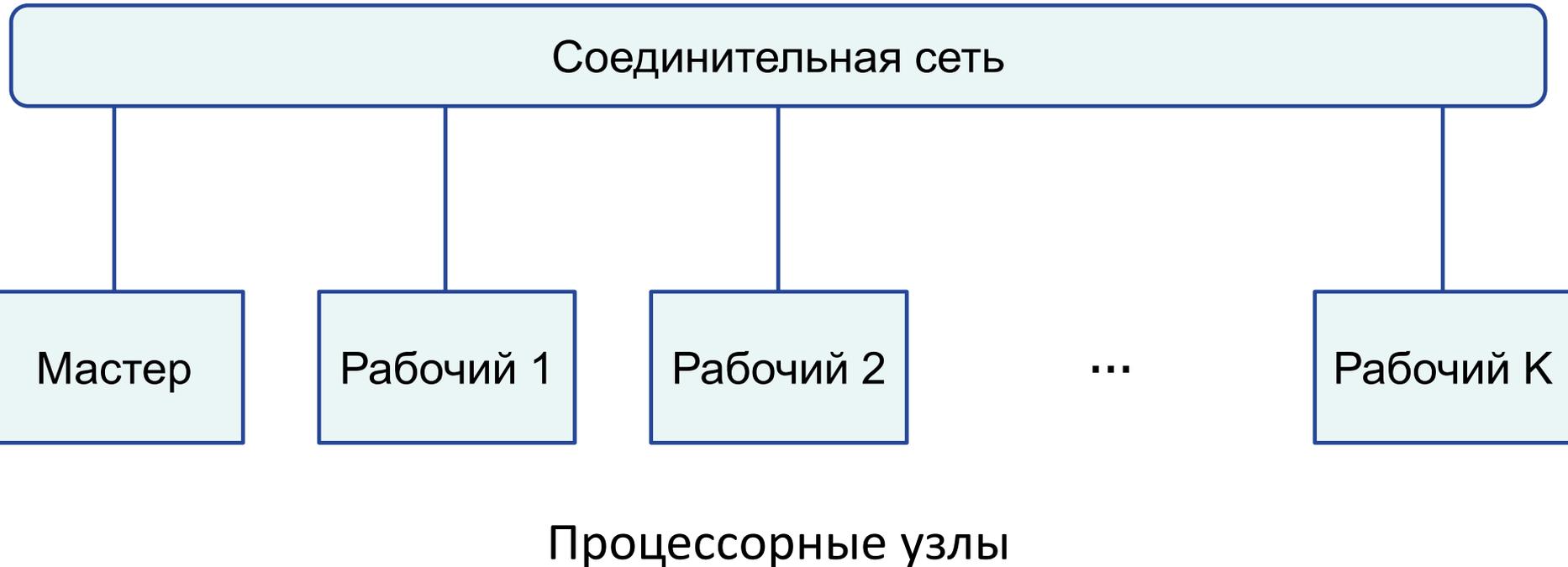


$$K_{max} = f(n)$$

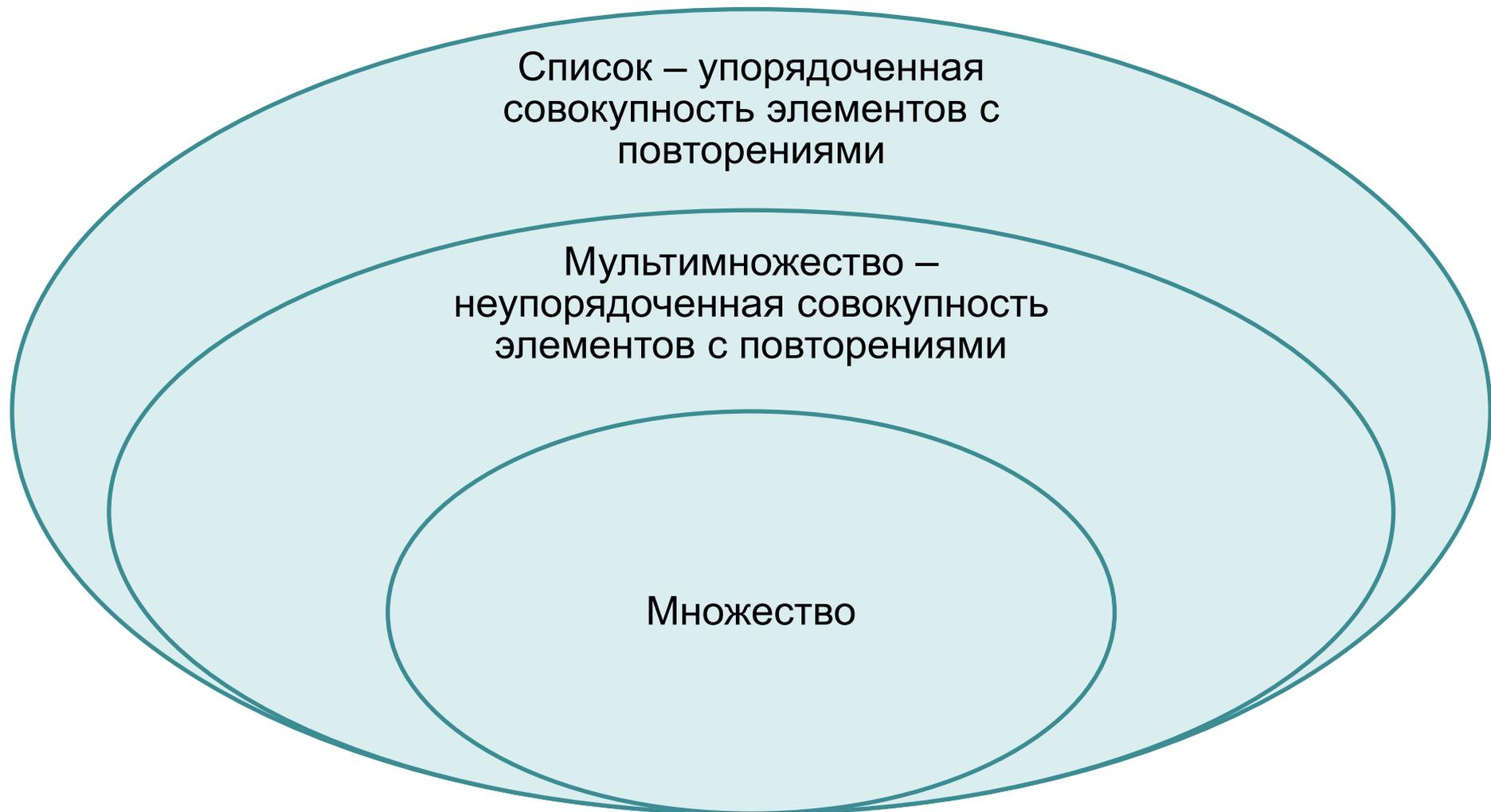
Модель параллельных вычислений BSF (Bulk-Synchronous Farm)

- Область применения:
 - Многопроцессорные системы с распределенной памятью
 - Параллельные итерационные алгоритмы с высокой вычислительной сложностью
- Позволяет предсказать:
 - ускорение параллельного алгоритма
 - **границу масштабируемости параллельного алгоритма**
(уникальное качество модели BSF)
- Требуется:
 - Организация многопроцессорной системы по схеме Мастер/рабочие
 - Алгоритм должен быть представлен в виде операций над списками с использованием функций высшего порядка Map/Reduce

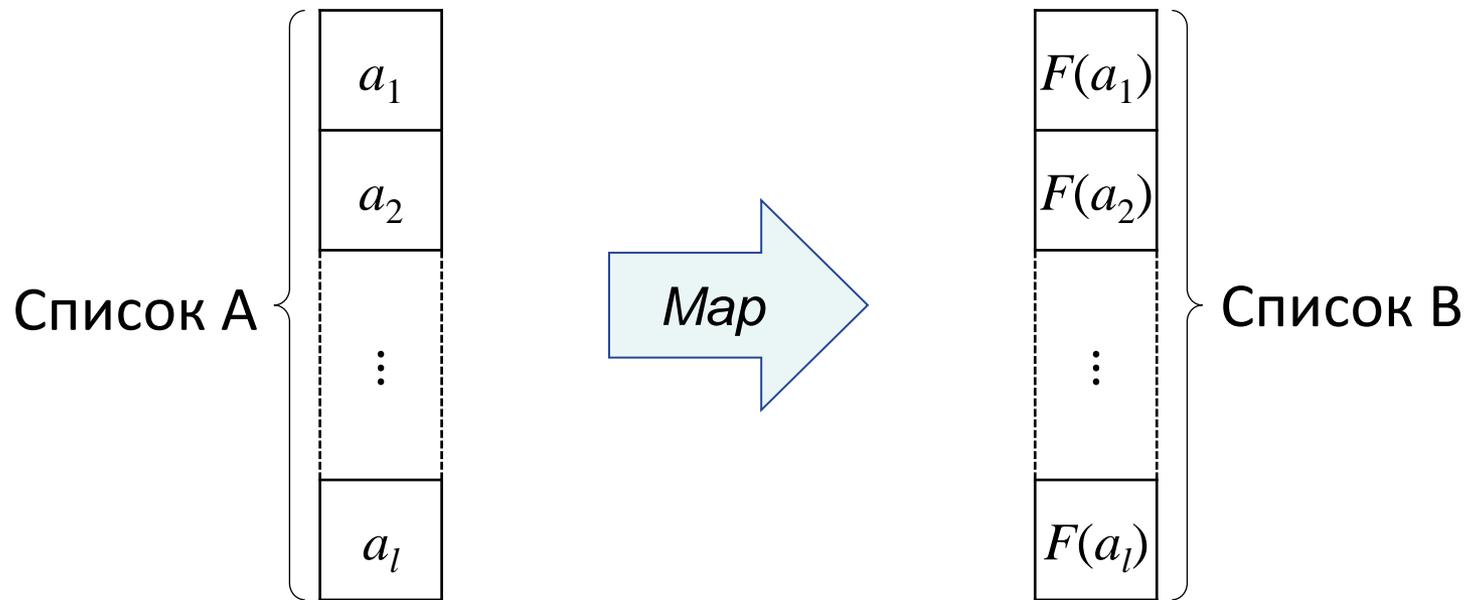
BSF-компьютер



Представление алгоритма в виде операций над списками



Функция высшего порядка *Map*

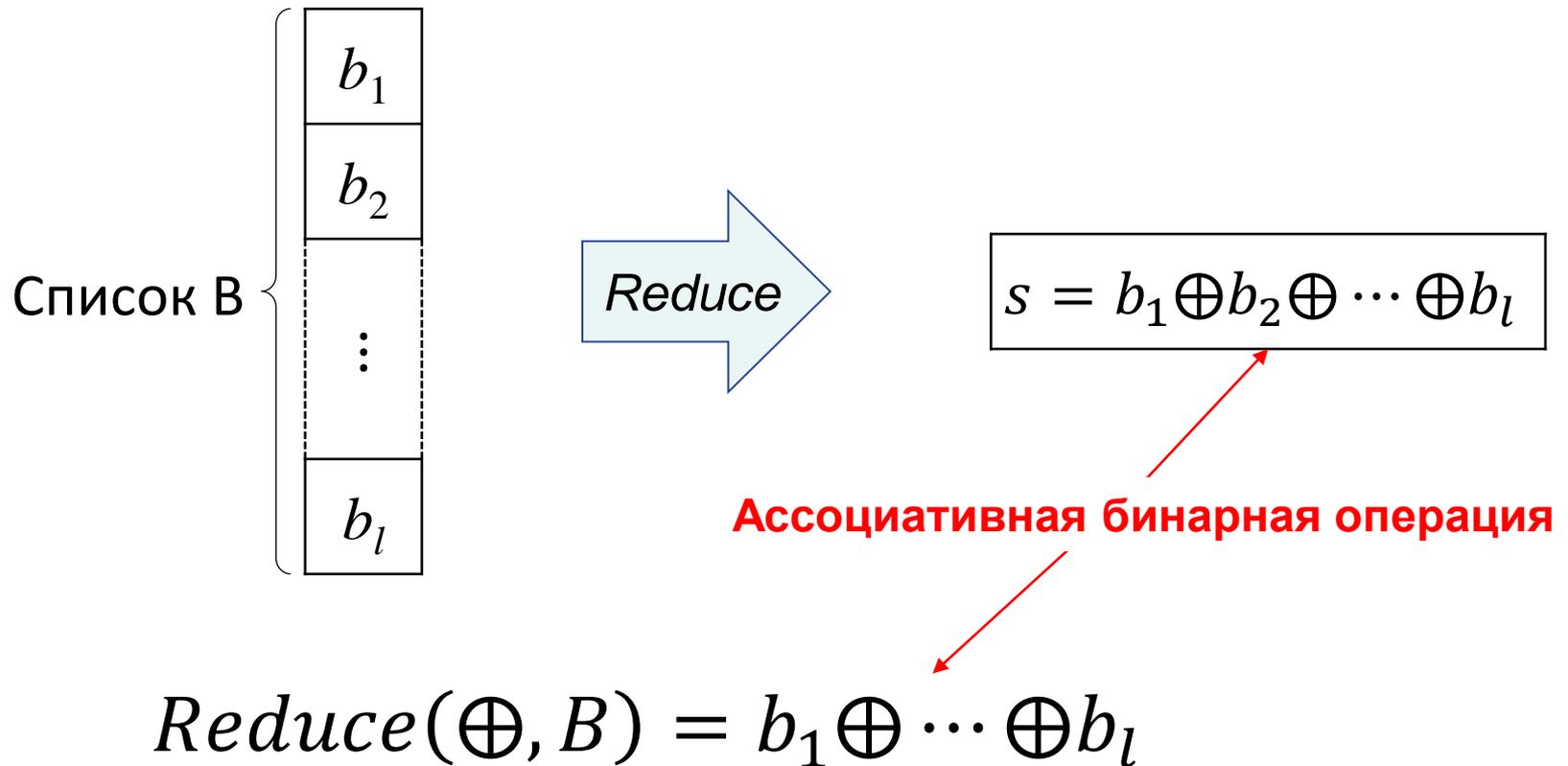


$$\text{Map}(F, A) = B$$

Ограничение применимости для множеств

BSF-модель HE применима для множеств в случае, когда функция *Map* порождает дубликаты

Функция высшего порядка *Reduce*

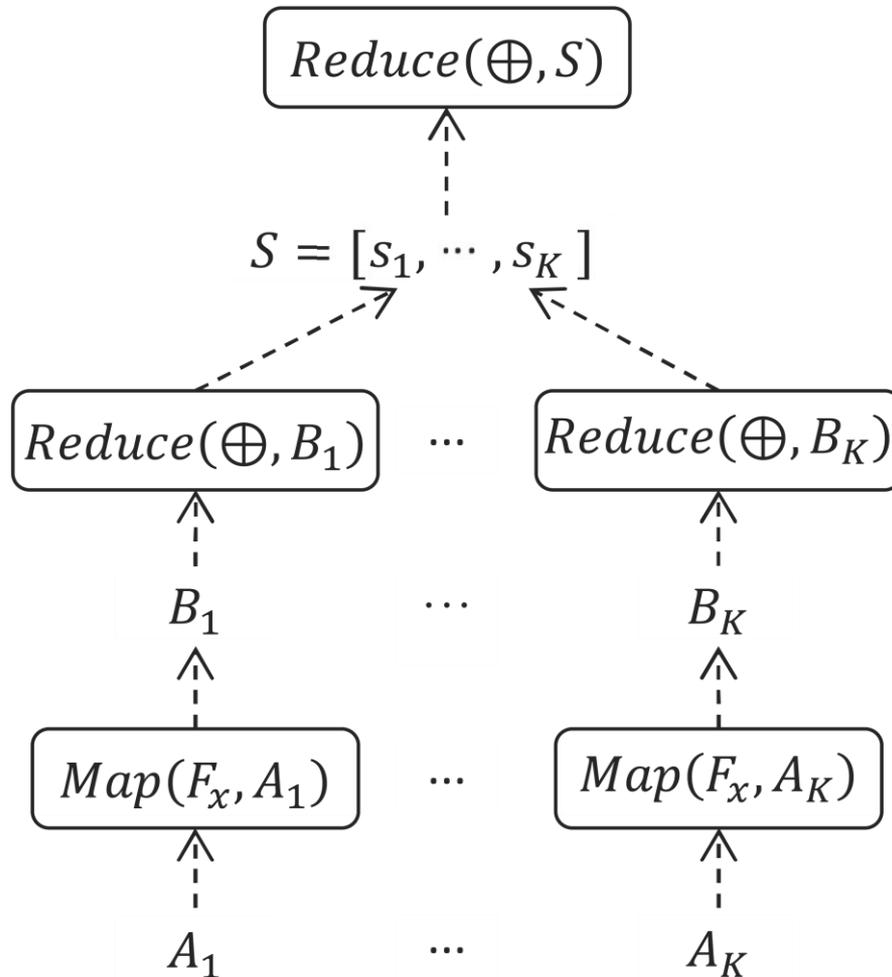


Шаблон последовательного BSF-алгоритма

1. *Input*(A, x_0)
2. $i := 0$
3. $B := \text{Map}(F_{x^{(i)}}, A)$ ←-----
4. $s := \text{Reduce}(\oplus, B)$
5. $x^{(i+1)} := \text{Compute}(x^{(i)}, s);$
6. $i := i + 1$
7. **if** *StopCond*($x^{(i+1)}, x^{(i)}$) **go to** 9
8. **go to** 3 -----
9. *Output*($x^{(i)}$);
10. **stop**

i	- номер итерации
$A \in [\mathcal{A}]$	- список исходных элементов данных
x_0	- начальное приближение
$F_x: \mathcal{A} \rightarrow \mathcal{B}$	- параметризованная функция
$B \in [\mathcal{B}]$	- список результирующих элементов
\oplus	- ассоциативная операция
x_i -----	- i -тое приближение

Схема распараллеливания BSF-алгоритма



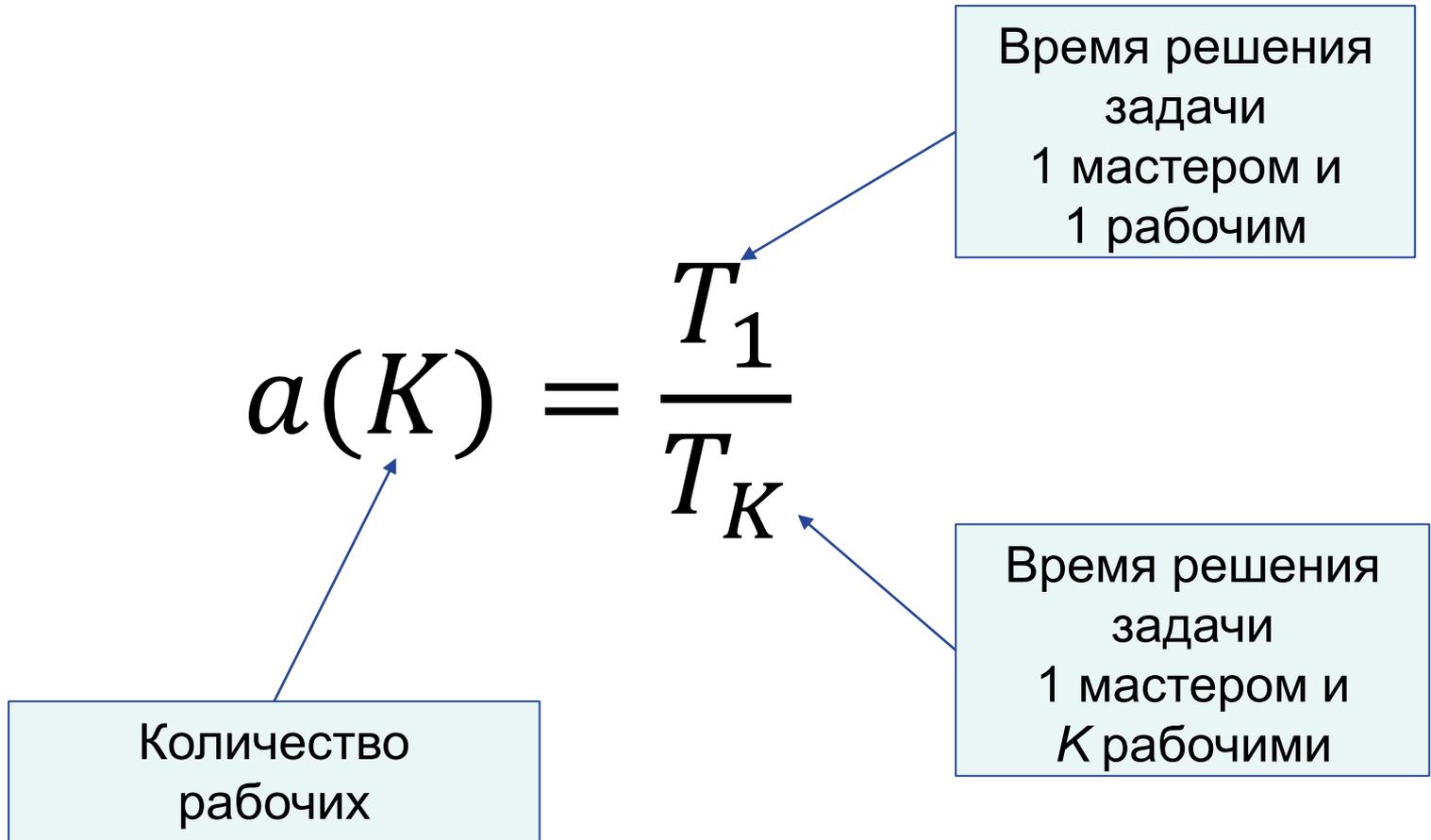
Шаблон параллельного алгоритма в модели BSF

Шаг	Мастер	Рабочий j (j=1,...,K)
1.	$i := 0; \text{Input}(x^{(0)})$	$\text{Input}(A_j)$
2.	$\text{SendToAllWorkers}(x^{(i)})$	$\text{RecvFromMaster}(x^{(i)})$
3.		$B_j := \text{Map}(F_{x^{(i)}}, A_j)$
4.		$s_j := \text{Reduce}(\oplus, B_j)$
5.	$\text{RecvFromAllWorkers}(s_1, \dots, s_K)$	$\text{SendToMaster}(s_j)$
6.	$s := \text{Reduce}(\oplus, [s_1, \dots, s_K])$	
7.	$x^{(i+1)} := \text{Compute}(x^{(i)}, s)$	
8.	$i := i + 1$	
9.	$\text{exit} := \text{StopCond}(x^{(i+1)}, x^{(i)})$	
10.	$\text{SendToAllWorkers}(\text{exit})$	$\text{RecvFromMaster}(\text{exit})$
11.	if not exit goto 2	if not exit goto 2
12.	$\text{Output}(x^{(i)})$	
13.	stop	stop

Стоимостная метрика модели BSF

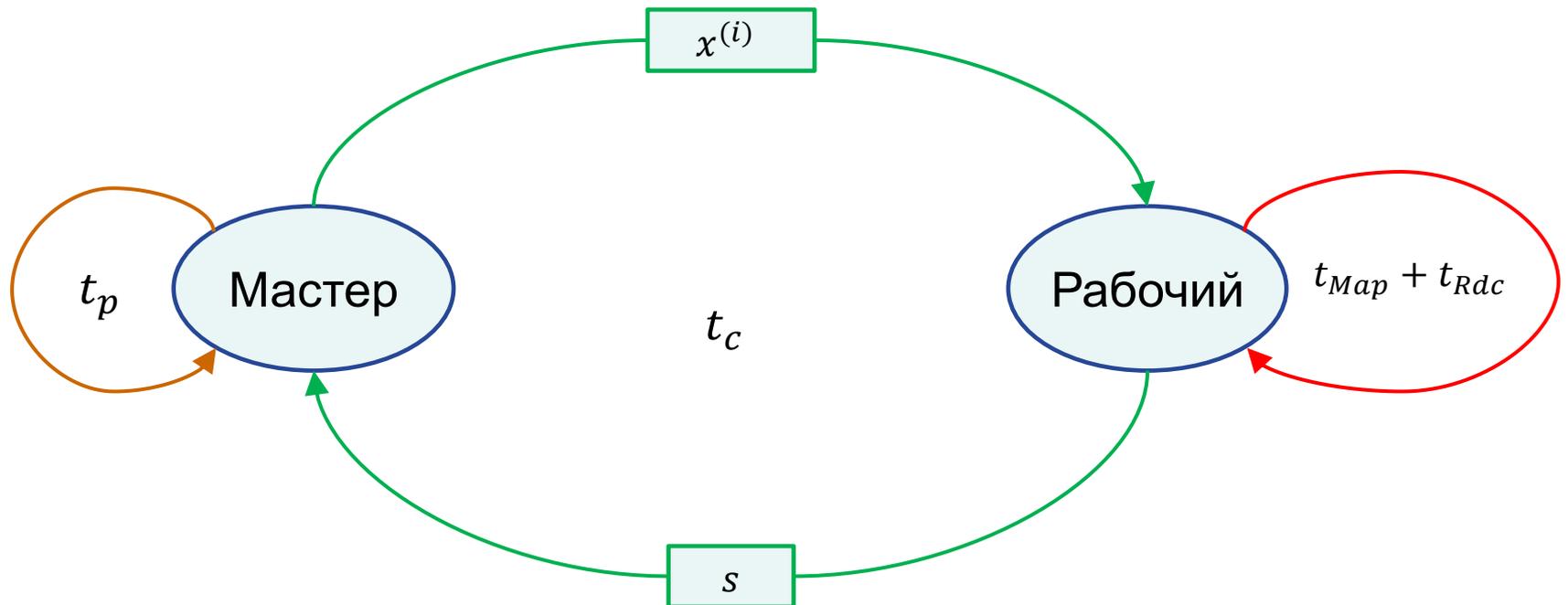
- K – количество рабочих процессорных узлов
- l – длина списков A и B
- L – латентность (время посылки сообщения длиной в 1 байт)
- t_c – время, которое тратит мастер на пересылку текущего приближения рабочему и на получение от него Reduce-свертки с учетом латентности
- t_{Map} – время выполнения функции Map для всего списка A
- t_{Rdc} – время выполнения функции $Reduce$ для всего списка B
- t_p – время на обработку результатов итерации и вычисления следующего приближения
- t_a – время выполнения операции \oplus : $t_a = t_{Rdc}/(l - 1)$

Ускорение в модели BSF



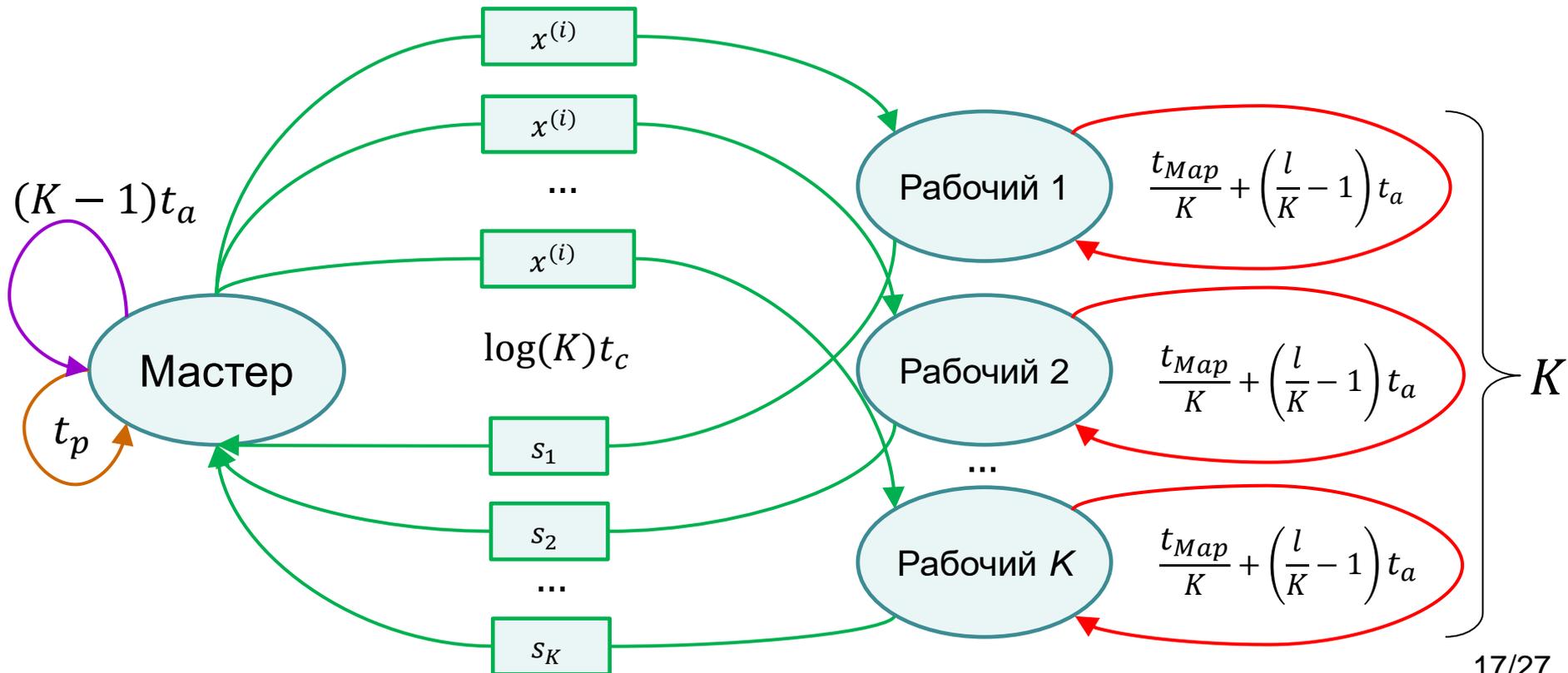
Оценка времени T_1

$$T_1 = t_p + t_c + t_{Map} + t_{Rdc}$$



Оценка времени T_K

$$T_K = (K - 1)t_a + t_p + (\log_2(K) + 1)t_c + \frac{t_{Map} + (l - K)t_a}{K}$$



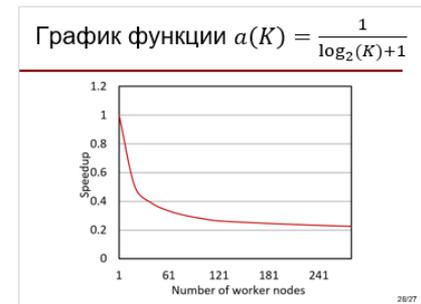
Ускорение для модели BSF

$$a_{BSF}(K) = \frac{t_p + t_c + t_{Map} + t_{Rdc}}{(K - 1)t_a + t_p + (\log_2(K) + 1)t_c + \frac{t_{Map} + (l - K)t_a}{K}}$$

Свойства:

$$a_{BSF}(1) = 1$$

$$a_{BSF}(K) > 0$$

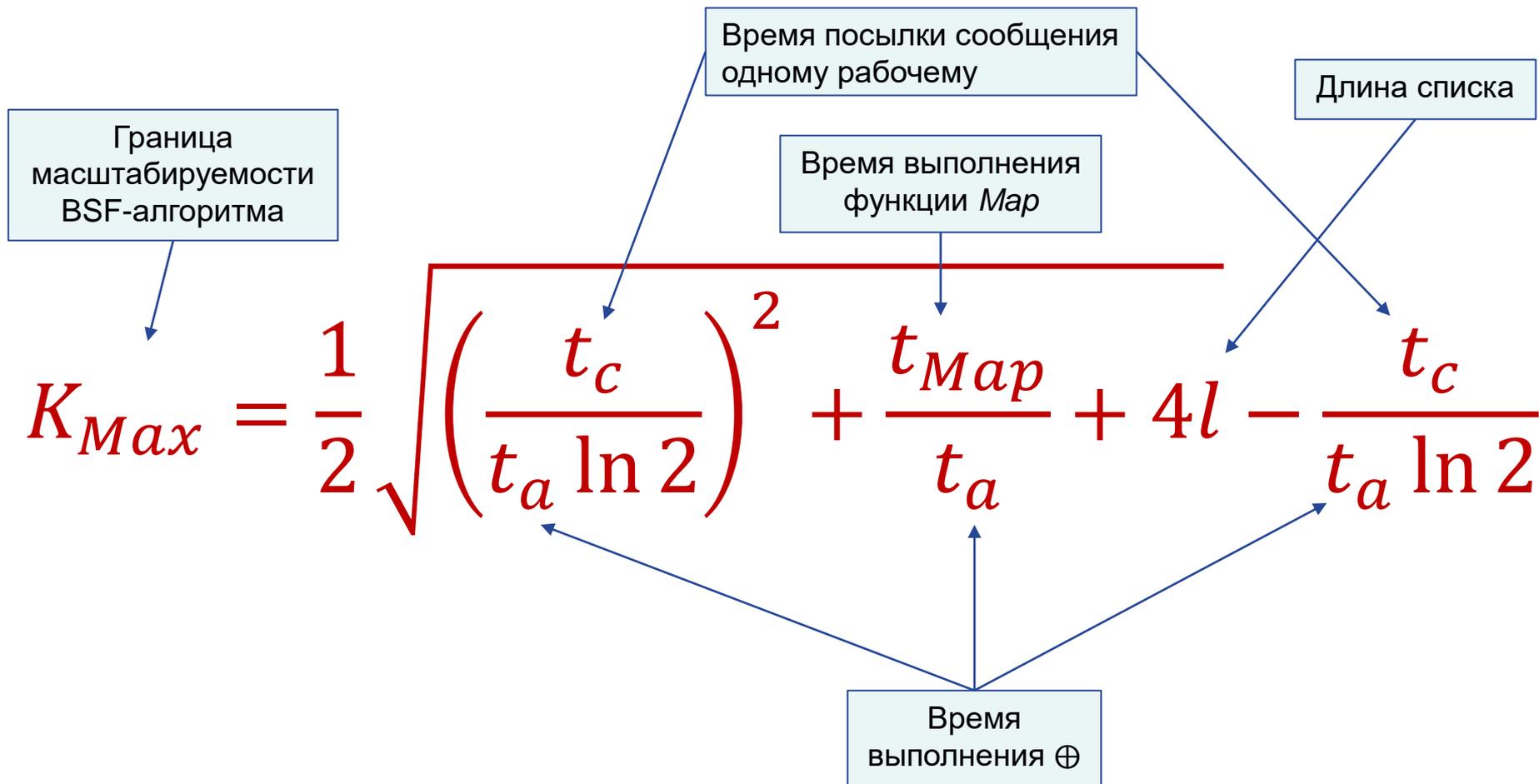


$$\lim_{t_{comp} \rightarrow 0} a_{BSF}(K) = \frac{1}{\log_2(K)+1} \text{ — монотонно убывает на } [1; +\infty)$$

$$\underbrace{t_{Map} + t_{Rdc} + t_p}_{t_{Rdc} = (l - 1)t_a}$$

$$t_{Rdc} = (l - 1)t_a$$

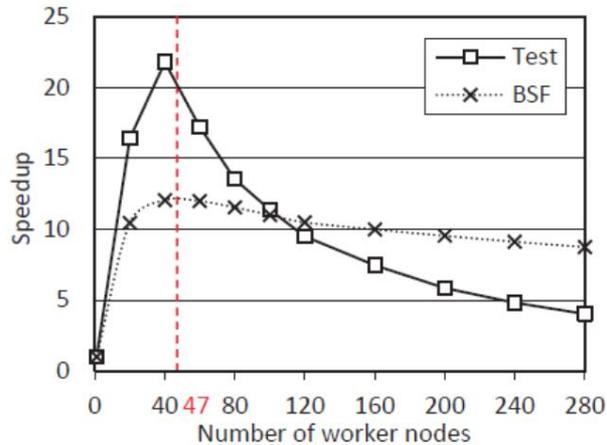
Граница масштабируемости BSF-алгоритма



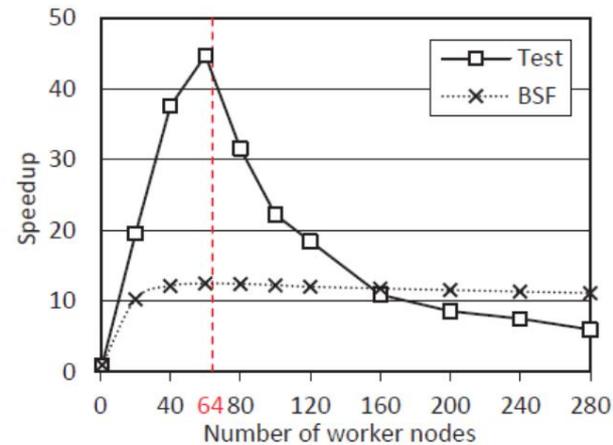
Алгоритм Якобі для приближенного решения СЛАУ

$$K_{Max} = O(n)$$

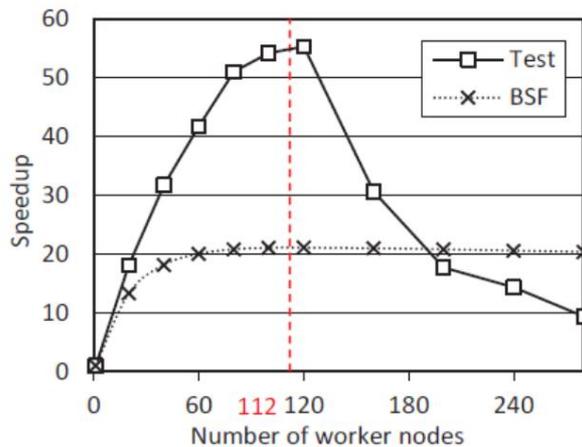
Алгоритм Якобі для приближенного решения СЛАУ



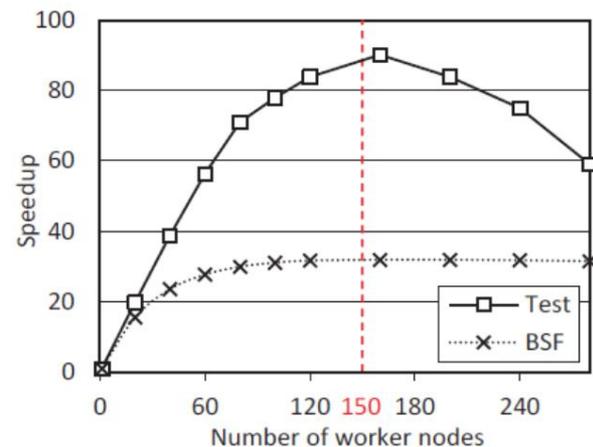
(a) $n = 1500$



(b) $n = 5000$



(c) $n = 10000$



(c) $n = 16000$

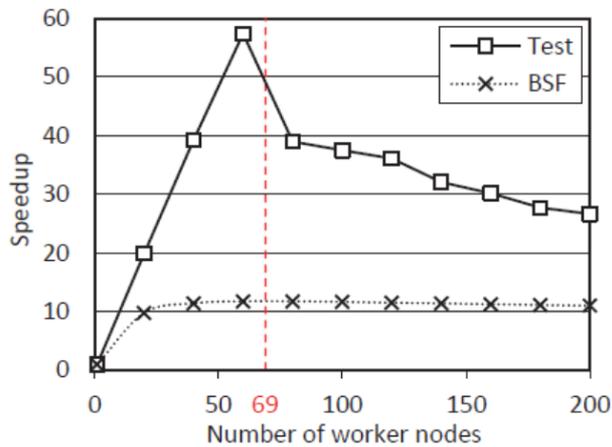
Ошибка для алгоритм Jacobi

n	1 500	5 000	10 000	16 000
K_{BSF}	47	64	112	150
K_{test}	40	60	120	160
$Error$	0.15	0.06	0.07	0.06

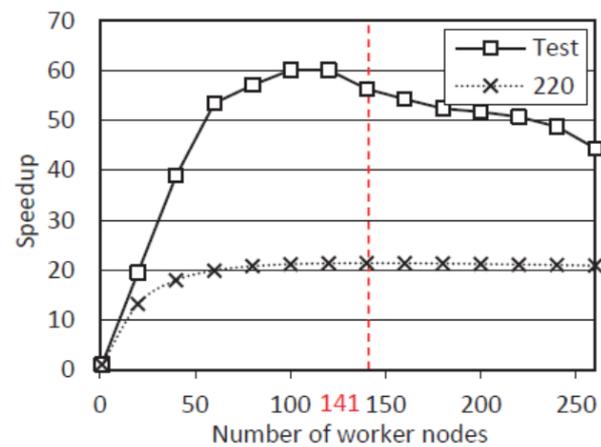
Задача n-тел

$$K_{Max} = O(n)$$

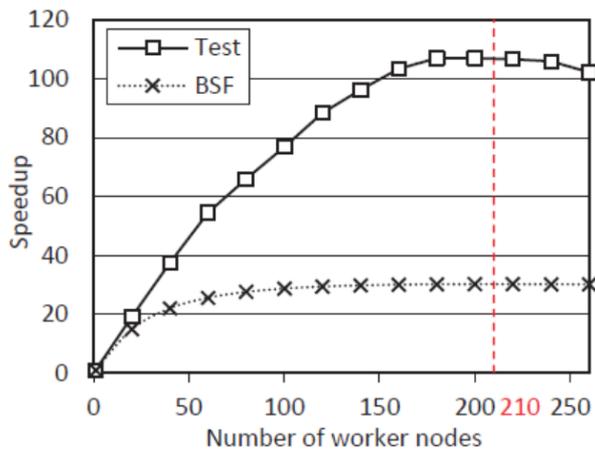
Задача n-тел



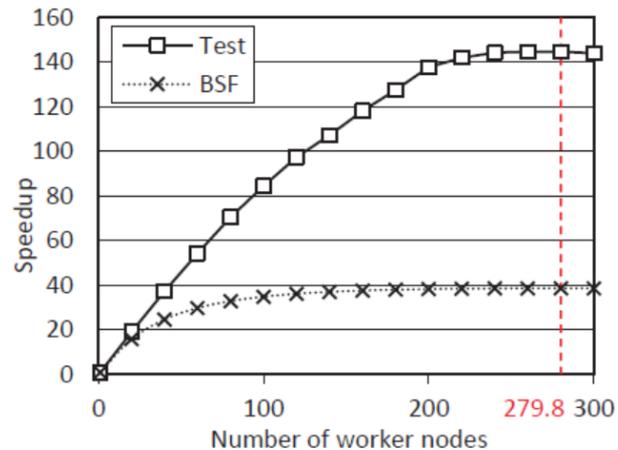
(a) $n = 1500$



(b) $n = 5000$



(c) $n = 10000$



(c) $n = 16000$

Ошибка для задачи n-тел

n	300	600	900	12 000
K_{BSF}	69	141	210	279.1
K_{test}	60	140	200	280
$Error$	0.13	0.01	0.05	3.6E-4

Параллельный BSF-каркас

- C++
- MPI + OpenMP
- Инкрементная компиляция
- Исходные коды + документация
<https://github.com/leonid-sokolinsky/BSF-skeleton>

Спасибо за внимание!

Вопросы?

Исходные коды + документация
<https://github.com/leonid-sokolinsky/BSF-skeleton>

График функции $a(K) = \frac{1}{\log_2(K)+1}$

