«Цифровая индустрия: состояние и перспективы развития 2018» (ЦИСП’2018)
(Челябинск, 13-15 ноября 2018 г.)

Исследование масштабируемости итерационных алгоритмов в суперкомпьютерном моделировании физических процессов

Н. А. Ежова

Л.Б. Соколинский

Южно-Уральский государственный университет
(национальный исследовательский университет)

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-00352а, Правительства РФ в соответствии с Постановлением №211 от 16.03.2013 г. (соглашение № 02.A03.21.0011) и Министерства образования и науки РФ (государственное задание 2.7905.2017/8.9).

Постановка задачи

Цифровые двойники

- компьютерная реализация физической модели

Системы диф. уравнений

- применяются численные методы решения

СЛАУ

- высокой вычислительной сложности

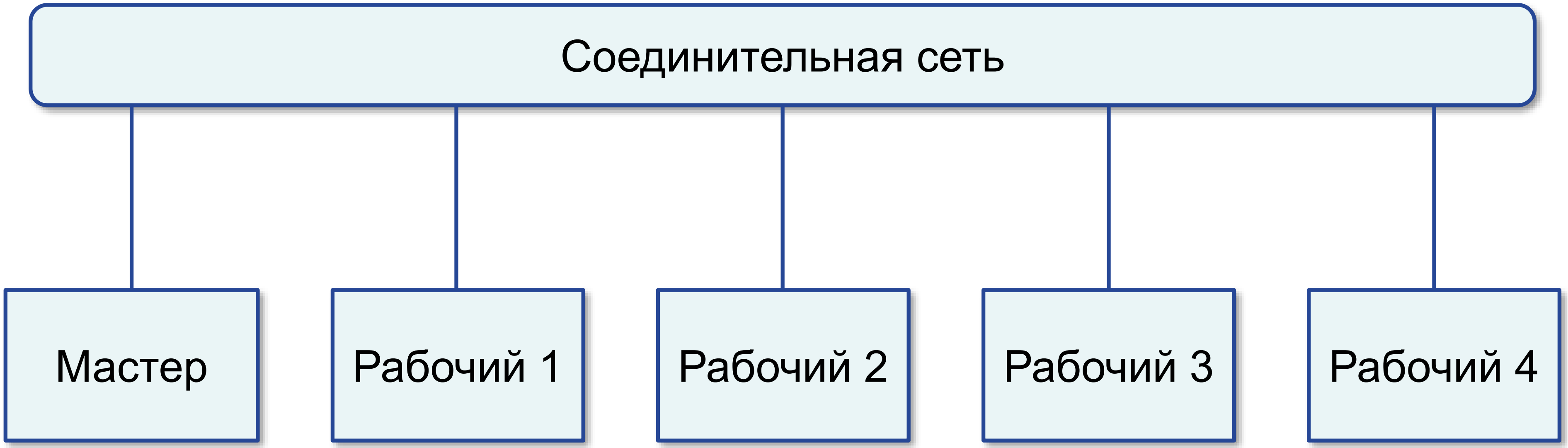
Решение

Разработана методика исследования масштабируемости ресурсоемких итерационных алгоритмов, применяемых в моделировании сложных физических процессов на суперкомпьютерных системах

Модель параллельных вычислений BSF (Bulk-Synchronous Farm)

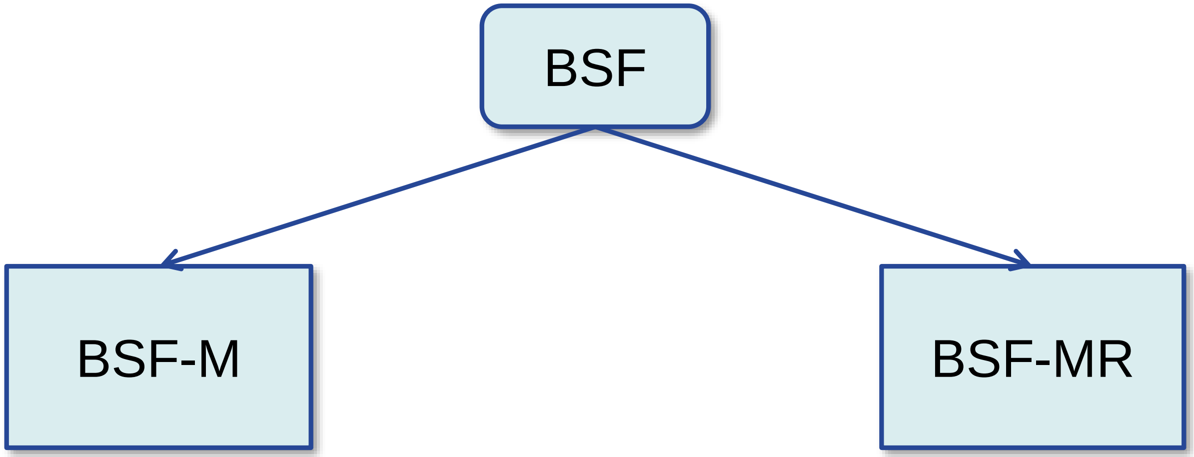
- *Модель BSF* – фреймворк (система правил и ограничений) для описания и анализа параллельных алгоритмов и программ
- Область применения:
  – Многопроцессорные системы с распределенной памятью
  – Параллельные итерационные алгоритмы с высокой вычислительной сложностью
- Позволяет предсказать:
  – границу масштабируемости параллельного алгоритма
  – ускорение параллельного алгоритма

BSF-компьютер



Процессорные узлы

Разновидности модели BSF



Модель BSF-M

Используется функция высшего порядка *Map*

Параметры BSF-M

K – количество рабочих

t_s – время, затрачиваемое мастером на передачу сообщения одному рабочему (без учета латентности)

L – латентность (время посылки сообщения длиной в 1 байт)

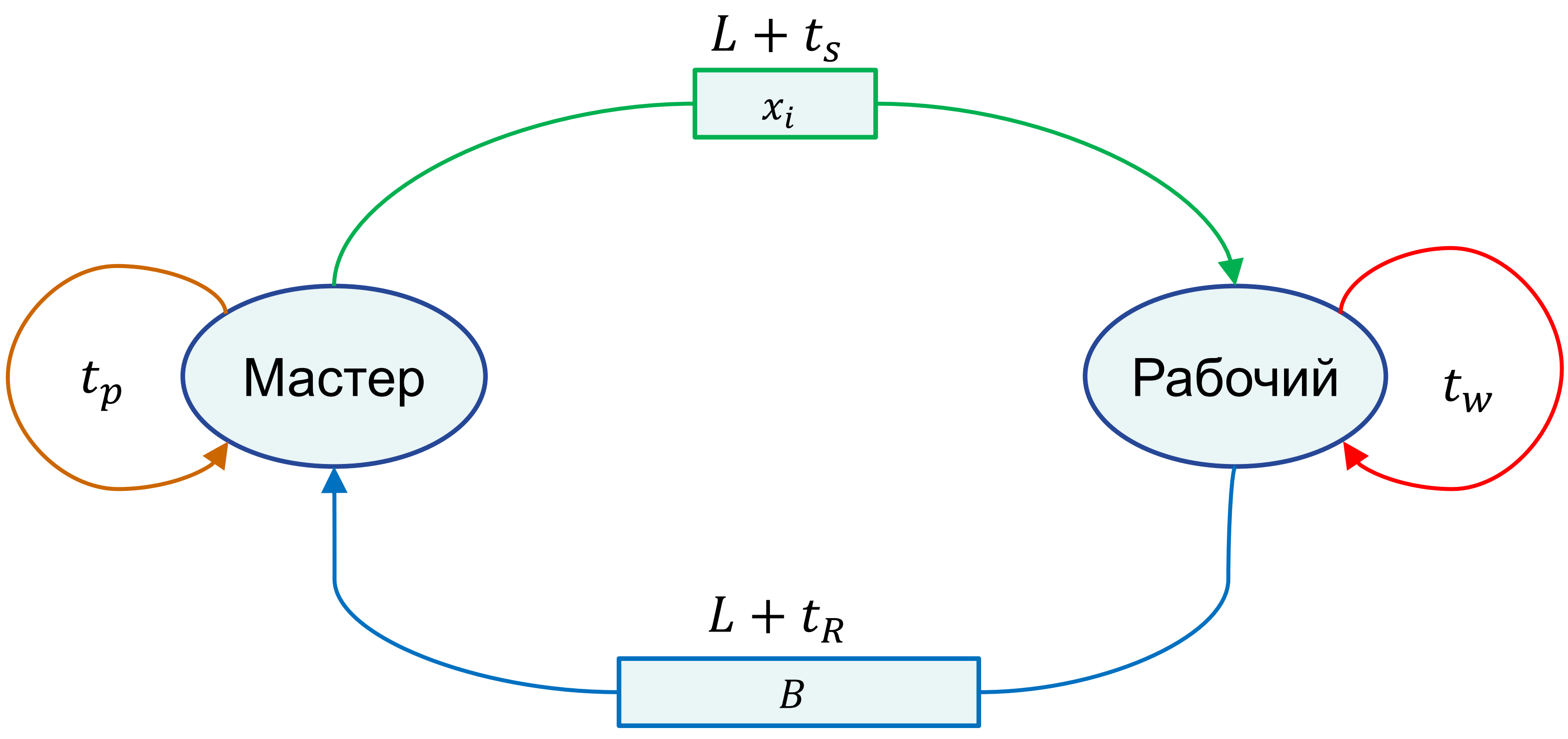
t_w – время выполнения задания бригадой из одного рабочего в рамках одной итерации

t_R – время, затрачиваемое мастером на получение результатов от всех рабочих (без учета латентности)

t_p – время, затрачиваемое мастером на обработку полученных результатов и проверку условия завершения

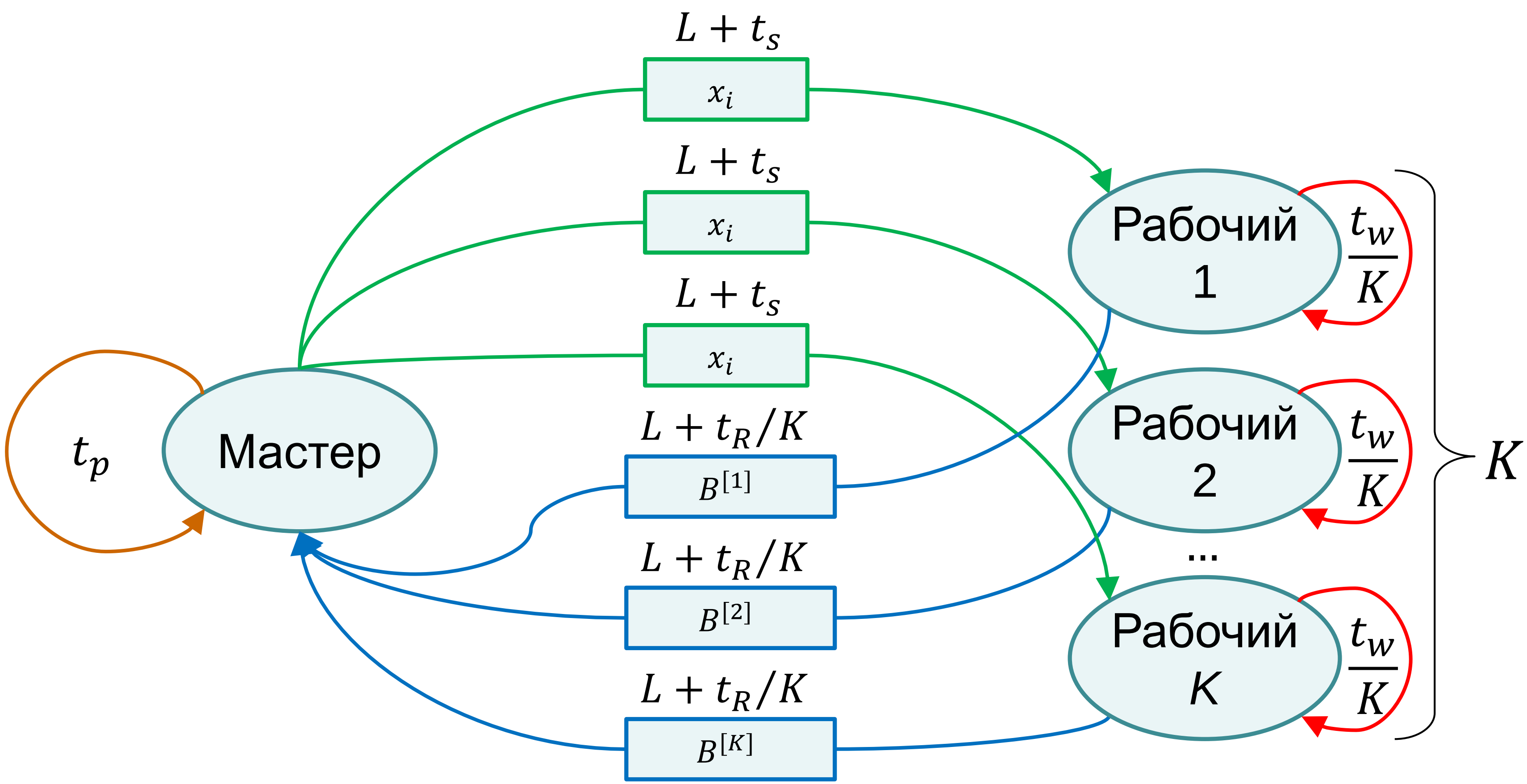
Время решения задачи системой из одного мастера и одного рабочего для BSF-M (сек.)

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



Время решения задачи системой из одного мастера и *K* рабочих для BSF-M (сек.)

$$T_K = K(L + t_s) + t_w/K + K(L + t_R/K) + t_p$$



Ускорение для BSF-M

$$a_{BSF-M}(K) = \frac{T_1}{T_K} = \frac{2L + t_s + t_R + t_p + t_w}{K^2(2L + t_s) + K\left(t_R + t_p\right) + t_w}$$

Граница масштабируемости для BSF-M

Максимальное количество рабочих

Время выполнения операции *Map*

$$K_M^{Max} = \sqrt{\frac{t_w}{2L + t_s}}$$

Латентность

Время посылки сообщения одному рабочему

Модель BSF-MR

Используются функции высшего порядка *Map* и *Reduce*

Параметры BSF-MR

K – количество рабочих

l – длина списка ($l = mK, m \in \mathbb{N}$)

t_s – время, затрачиваемое мастером на передачу сообщения одному рабочему (без учета латентности)

L – латентность (время посылки сообщения длиной в 1 байт)

t_w – время выполнения задания бригадой из одного рабочего в рамках одной итерации

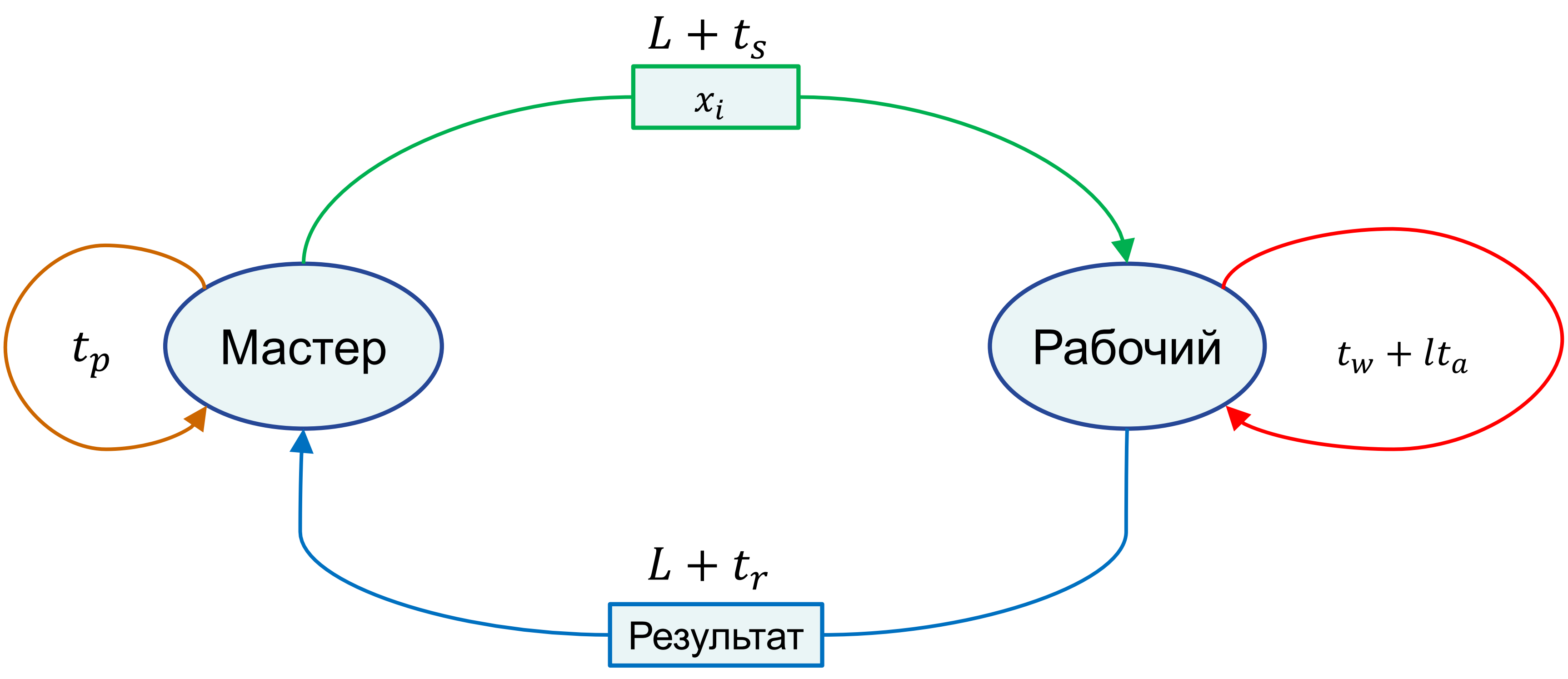
t_r – время, затрачиваемое мастером на получение сообщения от одного рабочего (без учета латентности)

t_a – время, необходимое для выполнения одной операции $\oplus$

t_p – время, затрачиваемое мастером на вычисление следующего приближения и проверку условия завершения

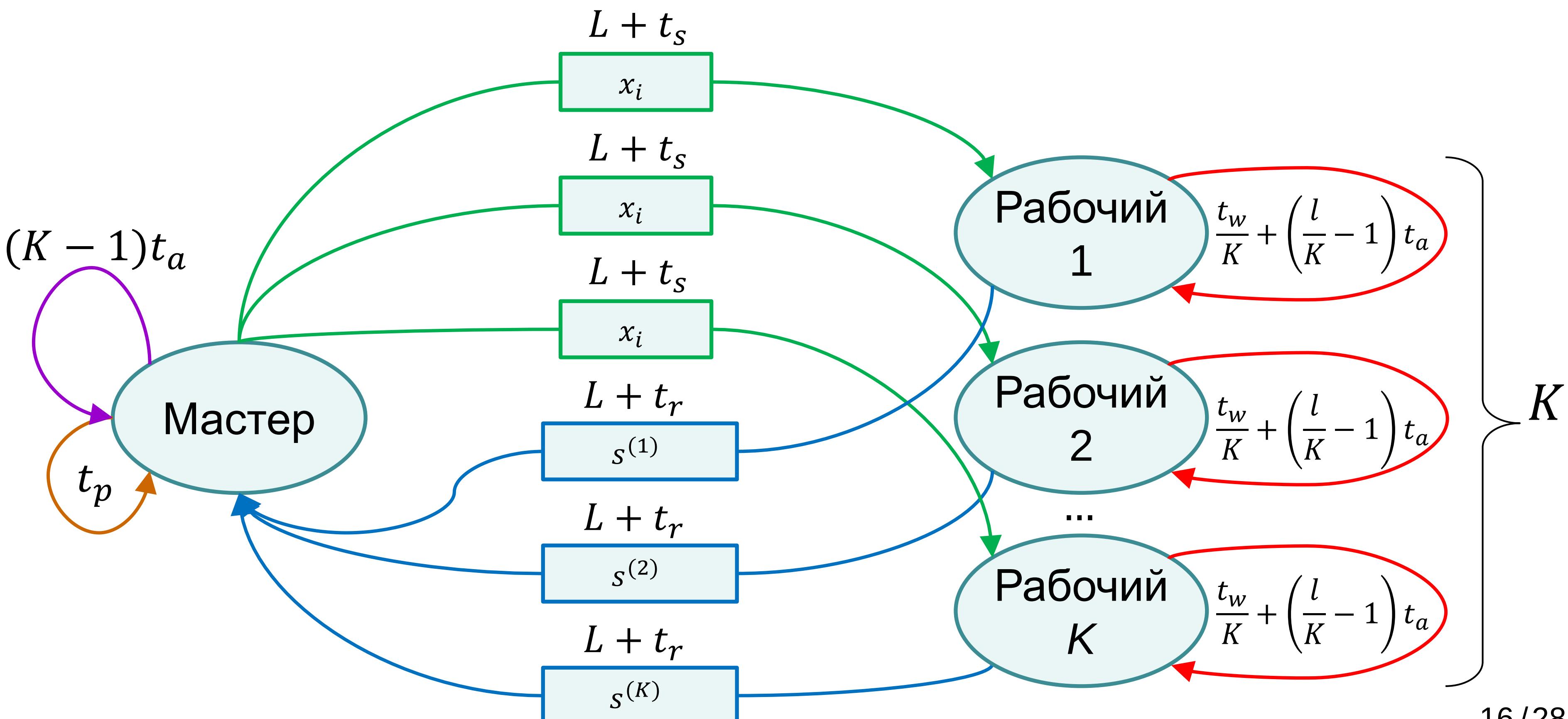
Время решения задачи системой из мастера и одного рабочего для BSF-MR

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



Время решения задачи системой из одного мастера и *K* рабочих для BSF-MR

$$T_K = K(L+t_s) + \frac{t_w}{K} + \left(\frac{l}{K} - 1\right) t_a + K(L+t_r) + (K-1)t_a + t_p$$



Ускорение для BSF-MR

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

Граница масштабируемости для BSF-MR

Максимальное количество рабочих

Время выполнения операции *Map*

Время выполнения операции Reduce

Время выполнения ⊕

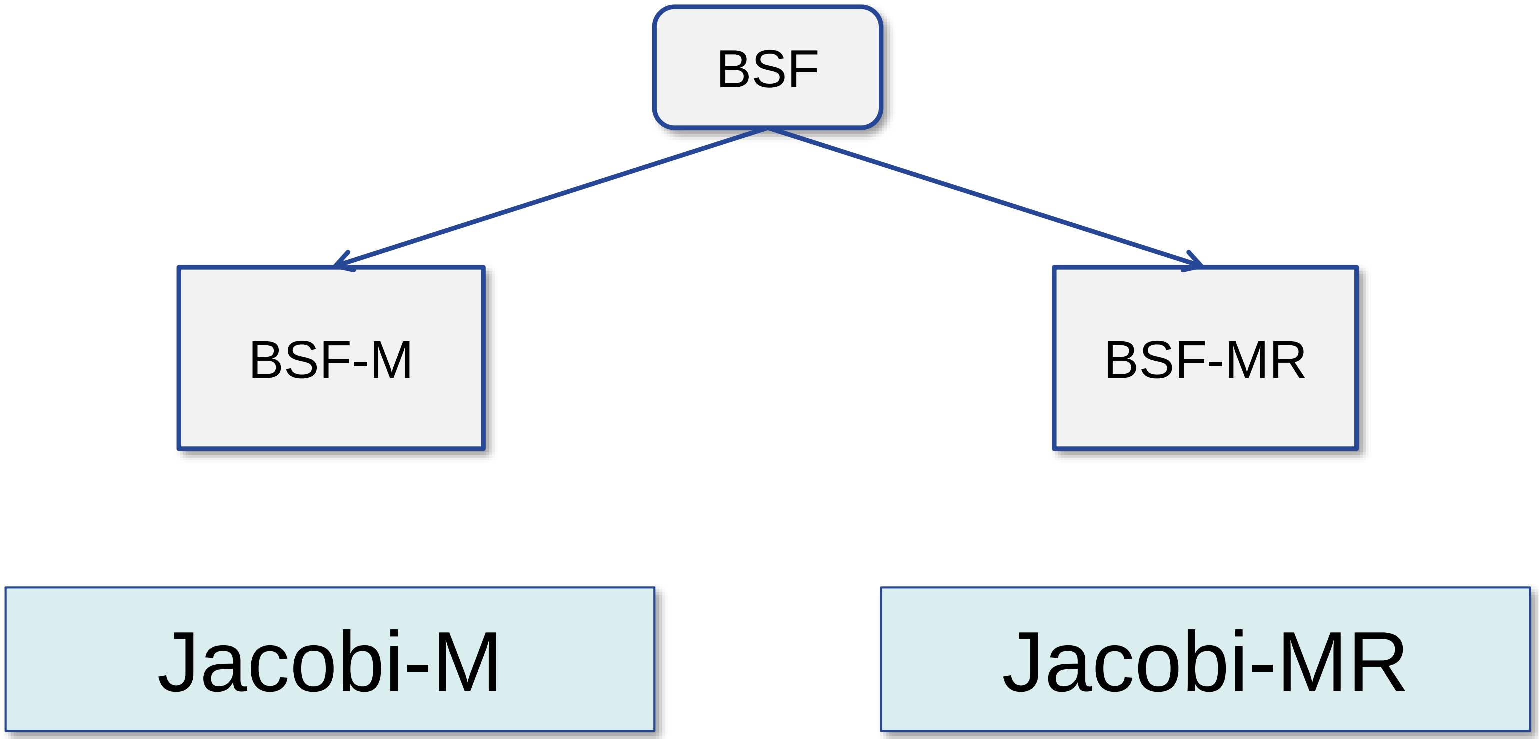
$$K_{Max} = \sqrt{\frac{t_w + l t_a}{2L + t_s + t_r + t_a}}$$

Латентность

Время посылки сообщения одному рабочему

Время получения результата от одного рабочего

Примеры применения моделей



Алгоритм Jacobi для приближенного решения СЛАУ

$$Ax = b$$

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \quad x = (x_1, \dots, x_n) \quad b = (b_1, \dots, b_n)$$

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix} \quad c_{ij} = \begin{cases} -\dfrac{a_{ij}}{a_{ii}}, \forall j \neq i \\ 0, \forall j = i \end{cases}$$

$$d = (d_1, \dots, d_n) \qquad d_i = b_i / a_{ii}$$

$$x^{(k+1)} = Cx^{(k)} + d$$

Алгоритм Jacobi-M над списками

$$F_{x^{(k)}} = d_i + \sum_{j=1}^{n} c_{ij} x_i$$

1. $Input(A, b); Comput(C, d); k \coloneqq 0; x^{(0)} \coloneqq d$
2. $C_j^{row} \coloneqq [c_{1j}, \dots, c_{nj}];\ x^{(k+1)} \coloneqq Map\left(F_{x^{(k)}}, C_j^{row}\right)$
3. $\text{Если } \left\|x^{(k+1)} - x^{(k)}\right\|^2 < \varepsilon$; перейти на шаг 5
4. $k \coloneqq k + 1$; перейти на шаг 2
5. Стоп

BSF-оценки для алгоритма Jacobi-M для многопроцессорных систем с распределенной памятью

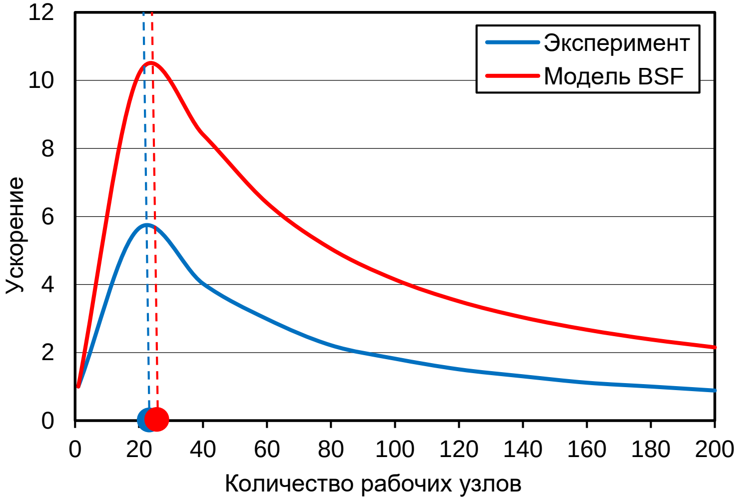
τ_{op} - время выполнения одной операции с плавающей точкой
τ_{tr} - время пересылки вещественного числа (без учета латентности)
n - количество уравнений
L - латентность

$$a_{Jacobi-M}(K) = \frac{K\left(2L + 2\tau_{tr}n + \tau_{op}\cdot(2n+2) + \tau_{op}2n^2\right)}{K^2(2L+\tau_{tr}n) + K\left(\tau_{tr}n + \tau_{op}\cdot(2n+2)\right) + \tau_{op}2n^2}$$

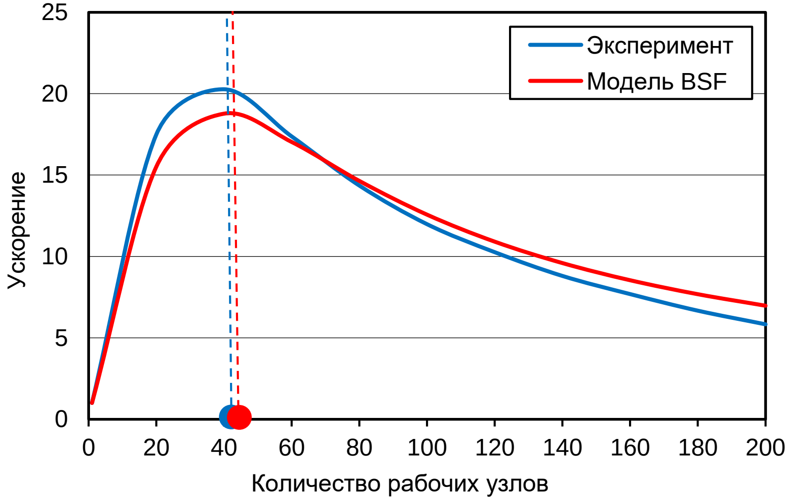
$$\boldsymbol{K_{Jacobi-M}^{Max} = O\left(\sqrt{n}\right)}$$

Ускорение алгоритма Jacobi-M: теория и практика

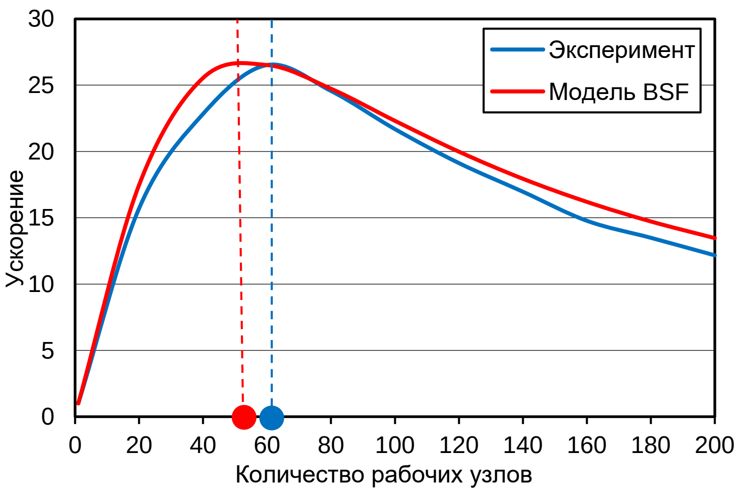
n = 1 500



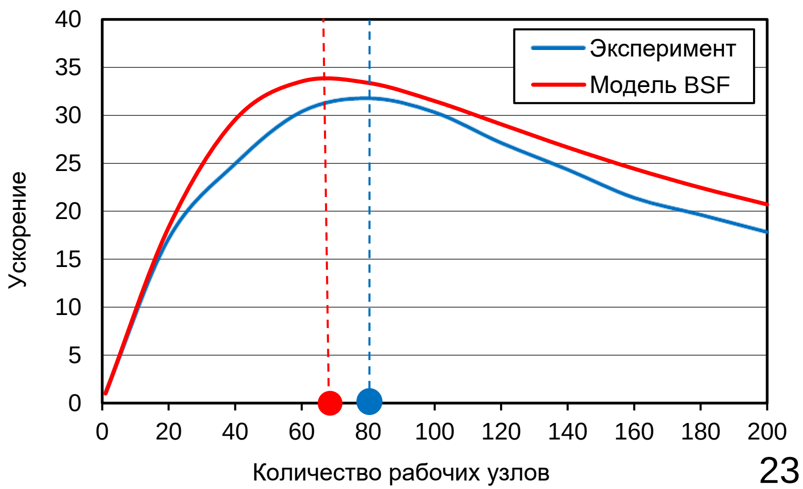
n = 5 000



n = 10 000



n = 16 000



Алгоритм Jacobi-MR над списками

$$F_x(j) = \left(c_{1j}x_j, \dots, c_{nj}x_j\right)$$

1. $Input(A, b); Comput(C, d); k := 0; x^{(0)} := d$
2. $C_i^{col} := Map\left(F_{x^{(k)}}, \left[c_{1j}, \dots, c_{nj}\right]\right)$
3. $x^{(k+1)} := Reduce\left(\oplus, C_i^{col}\right)$
4. $x^{(k+1)} := x^{(k+1)} + d$
5. Если $\left\|x^{(k+1)} - x^{(k)}\right\|^2 < \varepsilon$, перейти на шаг 7
6. $k := k + 1$; перейти на шаг 2
7. Стоп

BSF-оценки для алгоритма Jacobi-MR для многопроцессорных систем с распределенной памятью

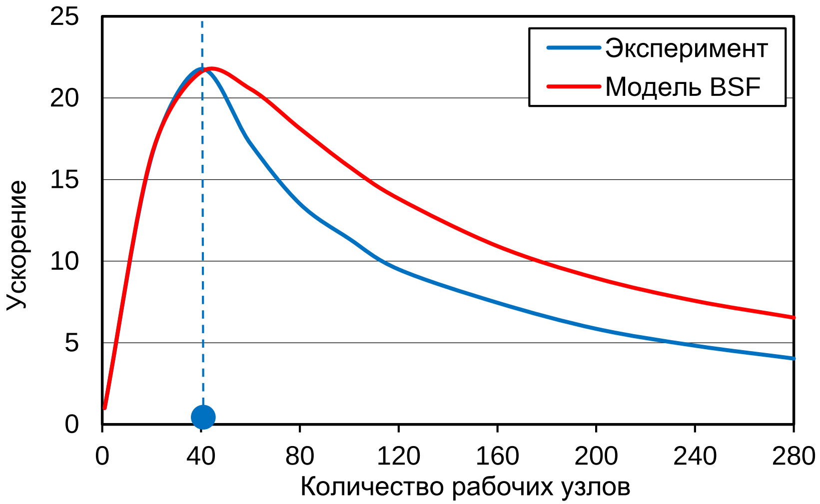
τ_{op} - время выполнения одной операции с плавающей точкой
τ_{tr} - время пересылки вещественного числа (без учета латентности)
n - количество уравнений
L - латентность

$$a_{Jacobi-MR}(K) = \frac{2(L + \tau_{tr}n) + \tau_{op}n(3n - K + 5)}{K^2\left(2(L + \tau_{op}n) + 3\tau_{op}n(n + K)\right)}$$

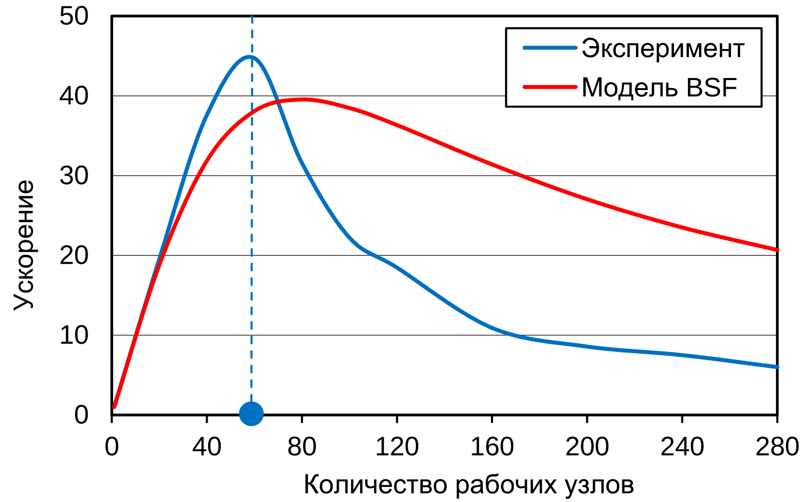
$$\boldsymbol{K_{Jacobi-MR}^{Max} = O\left(\sqrt{n}\right)}$$

Ускорение алгоритма Jacobi-MR: теория и практика

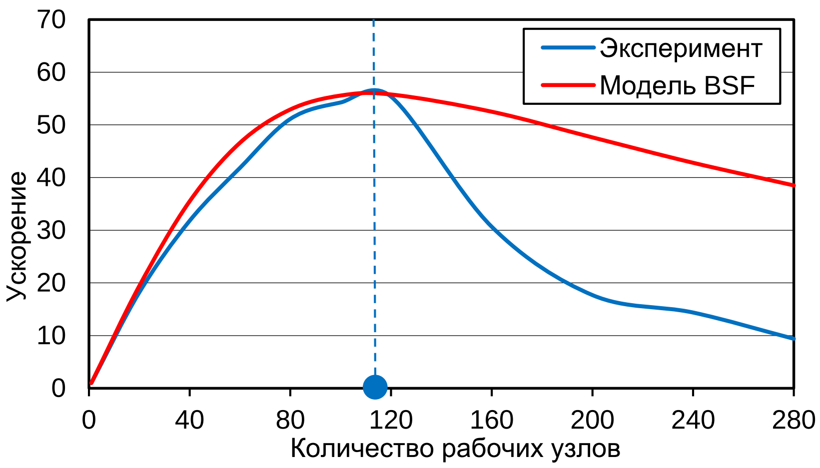
n = 1 500



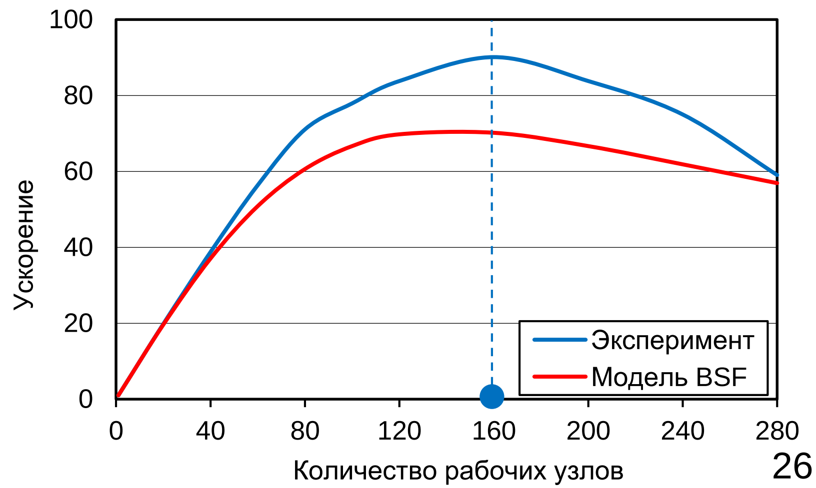
n = 5 000



n = 10 000



n = 16 000



Исходные коды

https://github.com/nadezhda-ezhova/Jacobi-MR
https://github.com/nadezhda-ezhova/Jacobi-M

Спасибо за внимание!