

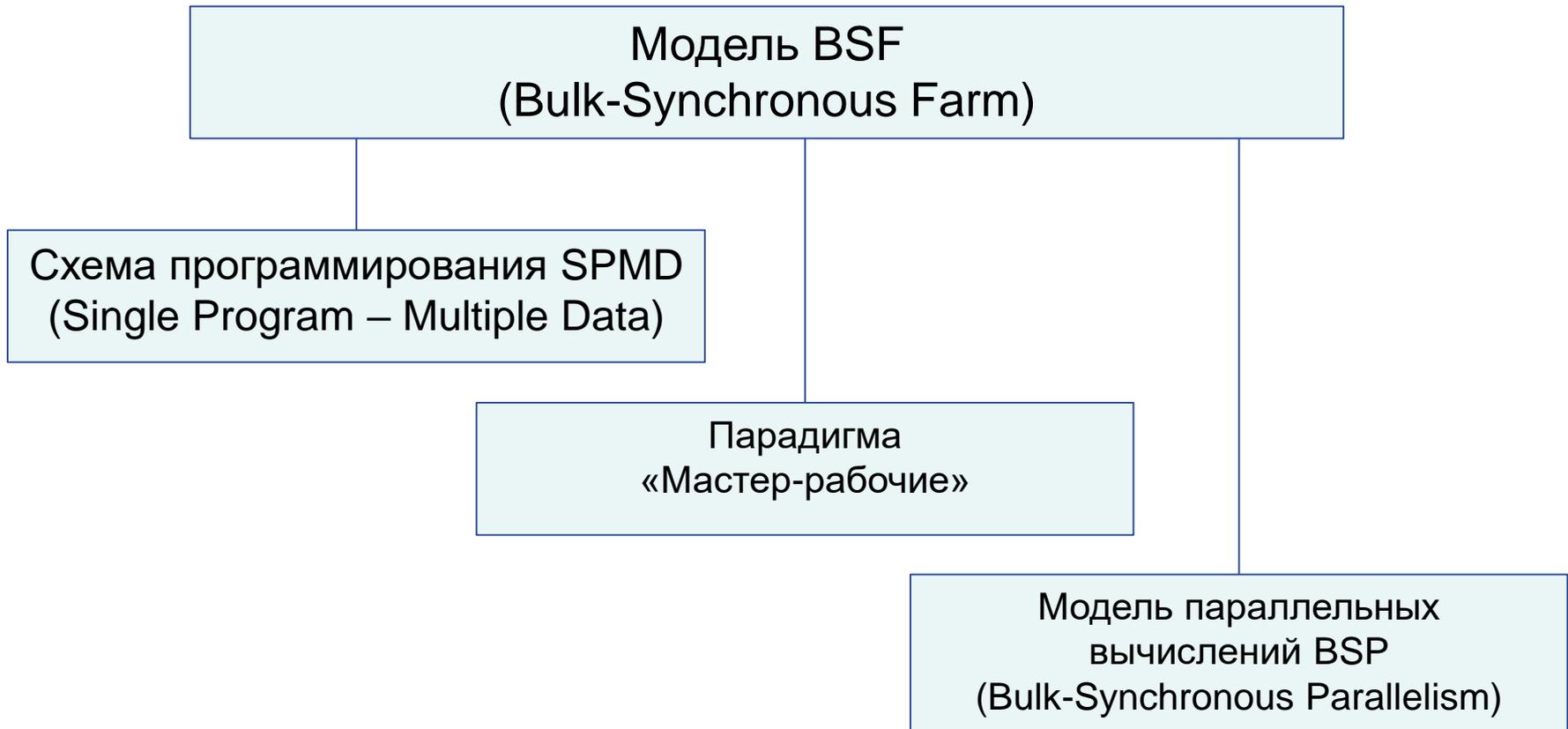


BSF: модель параллельных вычислений для многопроцессорных систем с распределенной памятью

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

Южно-Уральский государственный университет
(национальный исследовательский университет)

Фундамент модели BSF



Область применения модели BSF

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

Модель BSF позволяет предсказать:

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

BSF-компьютер (архитектурный компонент)



Процессорные узлы

Структура BSF-программы (спецификационный компонент)



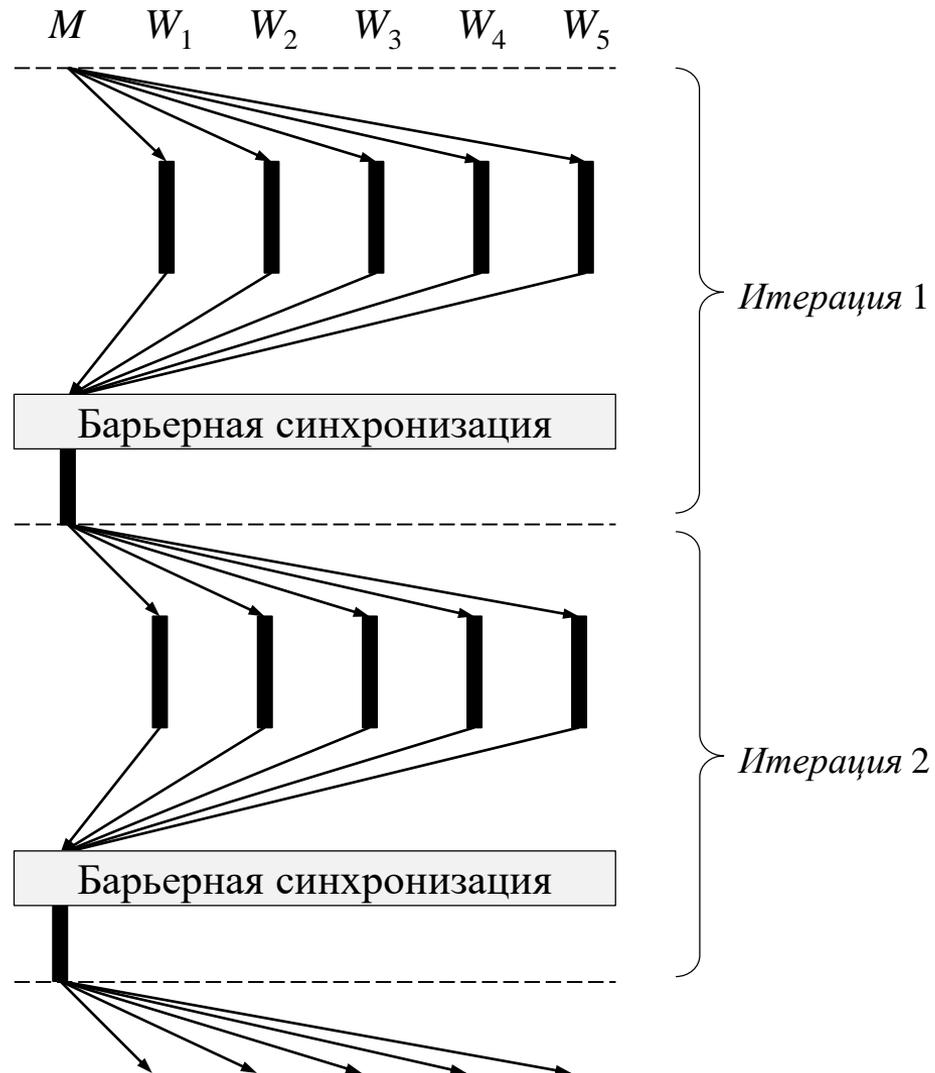
Деление программы на секции мастера и рабочего

```
// 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
```

Схема работы *BSF*-алгоритма (КОМПОНЕНТ ВЫПОЛНЕНИЯ)



Параметры BSF-модели (СТОИМОСТНЫЙ КОМПОНЕНТ)

- 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_p – время обработки мастером результатов, полученных от рабочих

Вычисление ускорения (СТОИМОСТНЫЙ КОМПОНЕНТ)

Параметры BSF-модели (стоимостный компонент)

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_p – время обработки мастером результатов, полученных от рабочих

10/34

$$T_1 = 2L + t_s + t_r + t_p + t_w$$

$$T_K = 2L \cdot K + t_s \cdot K + t_r + t_p + t_w/K$$

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

Исследование формулы

$$a(K) = \frac{K(2L + t_s + t_r + t_p + t_w)}{K^2(2L + t_s) + K(t_r + t_p) + t_w}$$

$$a(1) = \frac{1 \cdot (2L + t_s + t_r + t_p + t_w)}{1^2 \cdot (2L + t_s) + 1 \cdot (t_r + t_p) + t_w} = 1$$

Параметризация формулы

$$a(K) = \frac{K(2L + t_s + t_r + t_p + t_w)}{K^2(2L + t_s) + K(t_r + t_p) + t_w}$$

$$v = \lg(t_w/t_s)$$

$$t_s = 10^{-v} t_w$$

Параметры BSF-модели (стоимостный компонент)

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_p – время обработки мастером результатов, полученных от рабочих

10/34

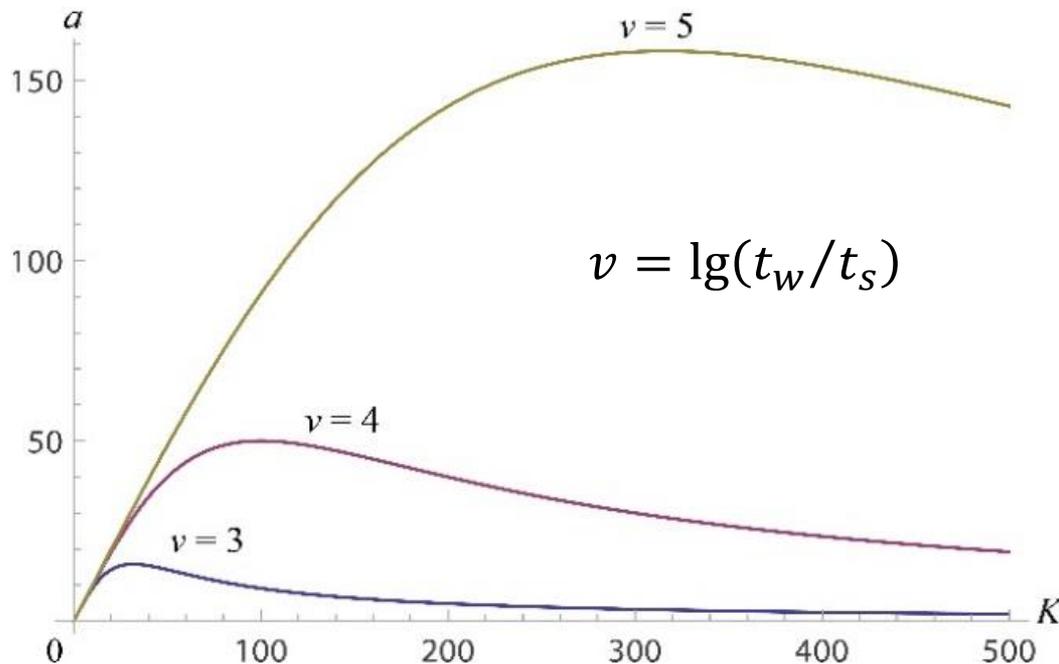
$$a(K) = \frac{K(2L + 10^{-v} t_w + t_r + t_p + t_w)}{K^2(2L + 10^{-v} t_w) + K(t_r + t_p) + t_w}$$

Пример использования формулы

$$a(K) = \frac{K(2L + 10^{-\nu}t_w + t_r + t_p + t_w)}{K^2(2L + 10^{-\nu}t_w) + K(t_r + t_p) + t_w}$$

Задача: $n = 10^4$, $t_w = n^3 = 10^{12}$, $t_p = t_r = n = 10^4$, $L = 0.5$

$$a(K) = \frac{K(1 + 10^{12-\nu} + 2 \cdot 10^4 + 10^{12})}{K^2(1 + 10^{12-\nu}) + 2K \cdot 10^4 + 10^{12}} \approx \frac{K(10^{8-\nu} + 2 + 10^8)}{10^{8-\nu}K^2 + 2K + 10^8}$$



Параметры BSF-модели (стоимостный компонент)

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_p – время обработки мастером результатов, полученных от рабочих

10/34

Параллельная эффективность BSF-алгоритма

$$e(K) = \frac{T_1}{K \cdot T_K} = \frac{2L + t_s + t_r + t_p + t_w}{K^2(2L + t_s) + K(t_r + t_p) + t_w} =$$
$$= \frac{2L + t_s}{K^2(2L + t_s) + K(t_r + t_p) + t_w} + \frac{t_r + t_p}{K^2(2L + t_s) + K(t_r + t_p) + t_w} + \frac{t_w}{K^2(2L + t_s) + K(t_r + t_p) + t_w}$$

$$K \gg 1$$

$$e(K) \approx \frac{1}{1 + \left(K^2(2L + t_s) + K(t_r + t_p) \right) / t_w}$$

Параметризация формулы

$$e(K) \approx \frac{1}{1 + \left(K^2(2L + t_s) + K(t_r + t_p) \right) / t_w}$$

$$v = \lg(t_w / t_s)$$

$$t_s = 10^{-v} t_w$$

Параметры BSF-модели (стоимостный компонент)

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_p – время обработки мастером результатов, полученных от рабочих

10/34

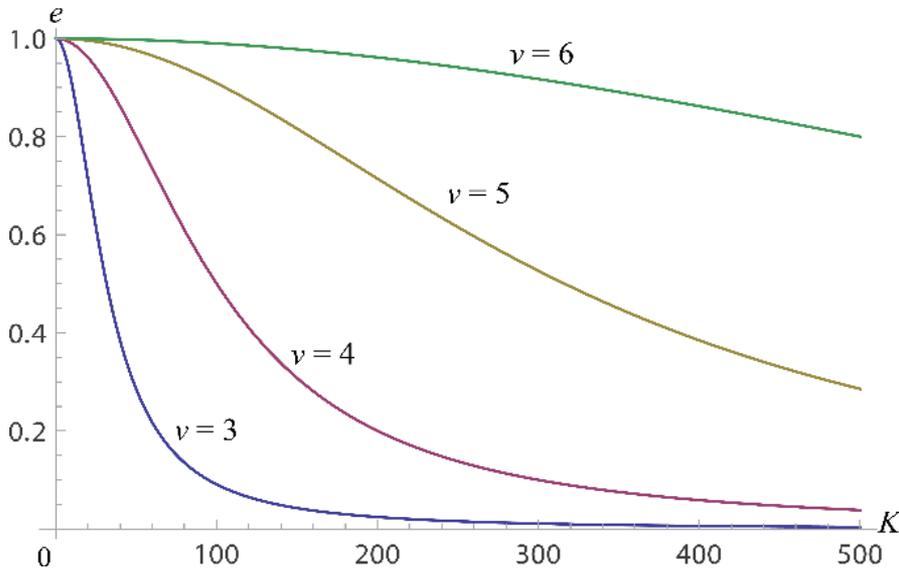
$$e(K) \approx \frac{1}{1 + \left(K^2(2L + 10^{-v} t_w) + K(t_r + t_p) \right) / t_w}$$

Пример использования формулы

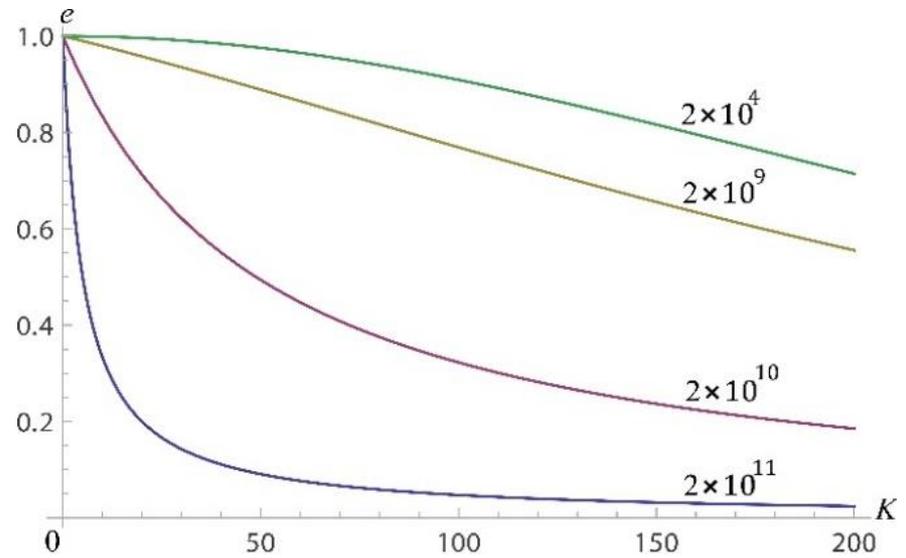
$$e(K) \approx \frac{1}{1 + \left(K^2(2L + 10^{-\nu}t_w) + K(t_r + t_p) \right) / t_w}$$

Задача: $n = 10^4$, $t_w = n^3 = 10^{12}$, $t_p = t_r = n = 10^4$, $L = 0.5$

$$e(K) \approx \frac{1}{1 + \left(K^2(1 + 10^{12-\nu}) + 2K \cdot 10^4 \right) / 10^{12}}$$



$$\nu = \lg(t_w/t_s)$$



Влияние $t_p + t_r$ на эффективность распараллеливания ($\nu = 5$)

Вычисление границы масштабируемости BSF-алгоритма

Найдем корни уравнения $a'(K) = 0$

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

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

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

Вычисление ускорения
(стоимостный компонент)

$$T_1 = 2L + t_s + t_r + t_p + t_w$$

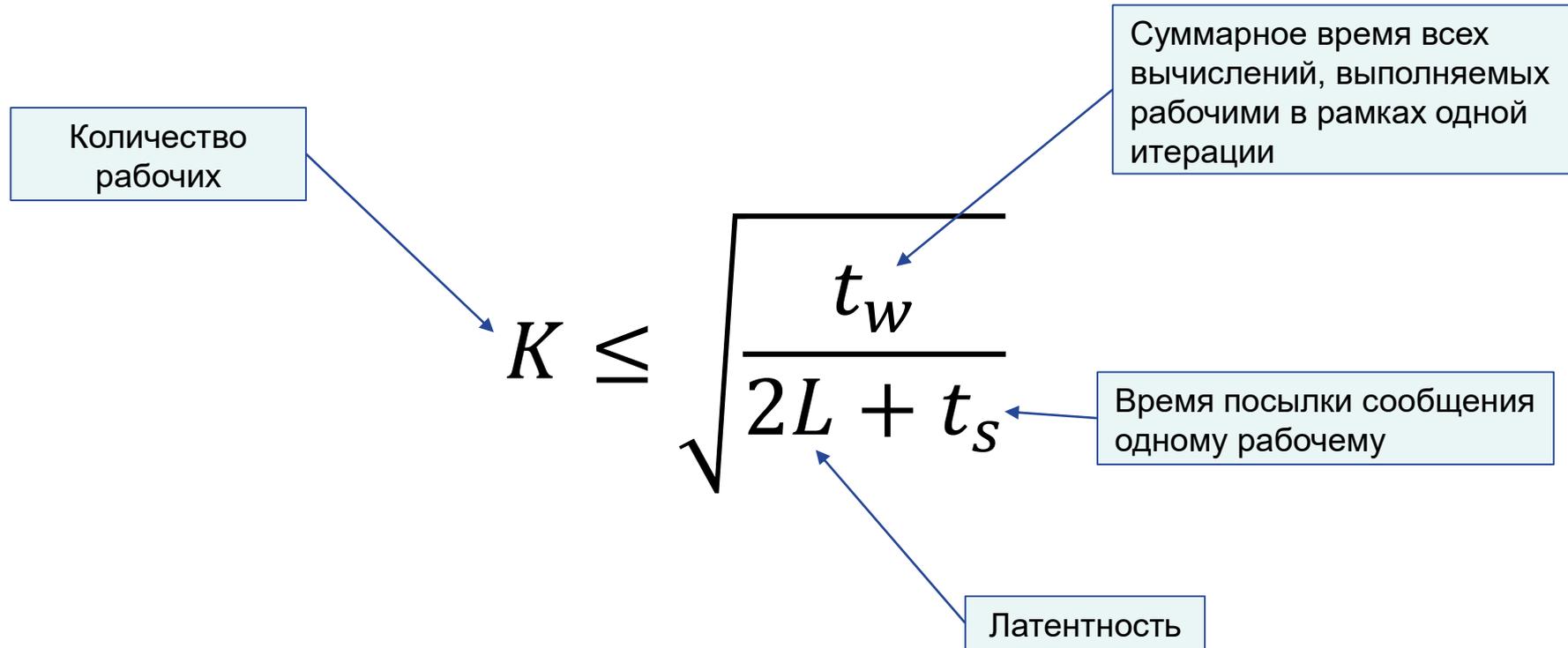
$$T_K = 2L \cdot K + t_s \cdot K + t_r + t_p + t_w/K$$

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

Параметры BSF-модели
(стоимостный компонент)

1. время, необходимое для поиска оптимального решения;
2. время, необходимое для поиска оптимального решения;
3. время, необходимое для поиска оптимального решения;
4. время, необходимое для поиска оптимального решения;
5. время, необходимое для поиска оптимального решения;

Граница масштабируемости BSF-алгоритма



Граница масштабируемости не зависит от:

t_r – время передачи результатов мастеру от рабочих

t_p – время обработки мастером результатов, полученных от рабочих

Эмулятор BSF-программ

C++ & MPI

Исходные тексты:

<https://github.com/nadezhda-ezhova/BSF-simulator>

Супершаг	Мастер	Рабочий
Посылка задания рабочим	MPI_Isend MPI_Waitall	MPI_Recv
Выполнение задания рабочими		usleep(t_w/K)
Барьерная синхронизация	MPI_Barrier	MPI_Barrier
Посылка результатов мастеру	MPI_Irecv	MPI_Send
Обработка результатов мастером	usleep(t_p)	

Параметры эксперимента по эмуляции выполнения BSF-программы

$$v = \lg \left(\frac{t_w}{t_s} \right)$$

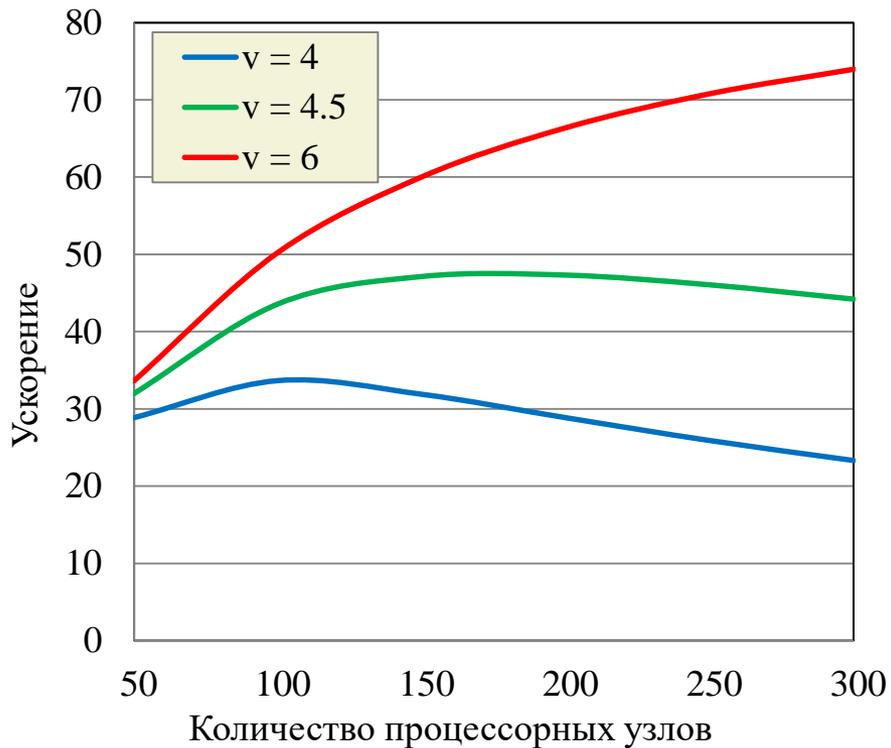
$$L = 2 \cdot 10^{-5} \text{ sec}$$

$$t_w = 500 \text{ sec}$$

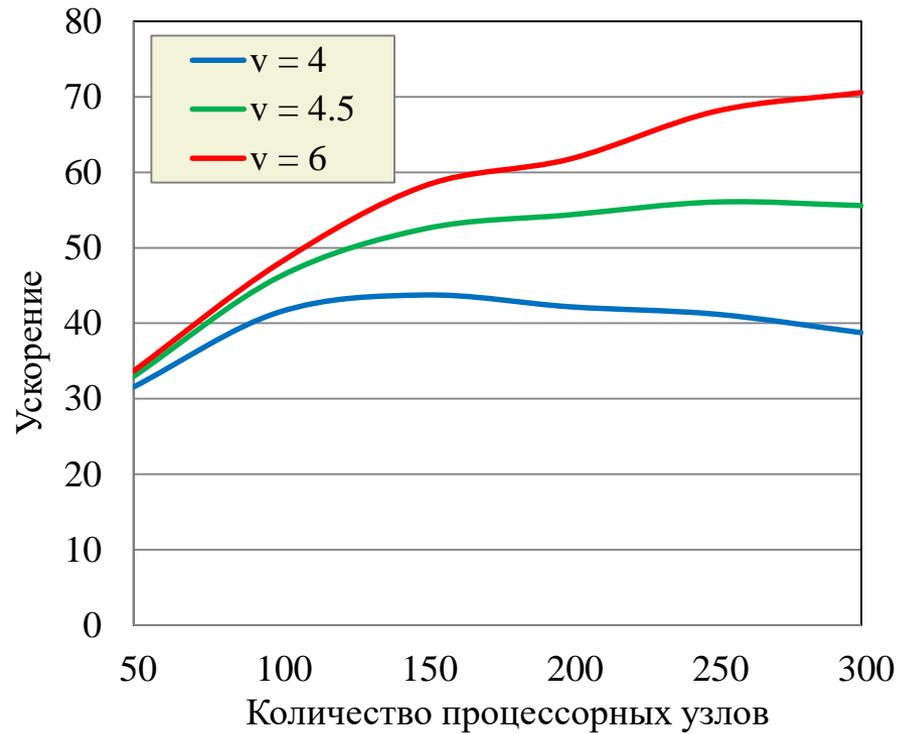
$$t_r = 0.01 \text{ sec}$$

$$t_p = 5 \text{ sec}$$

Ускорение



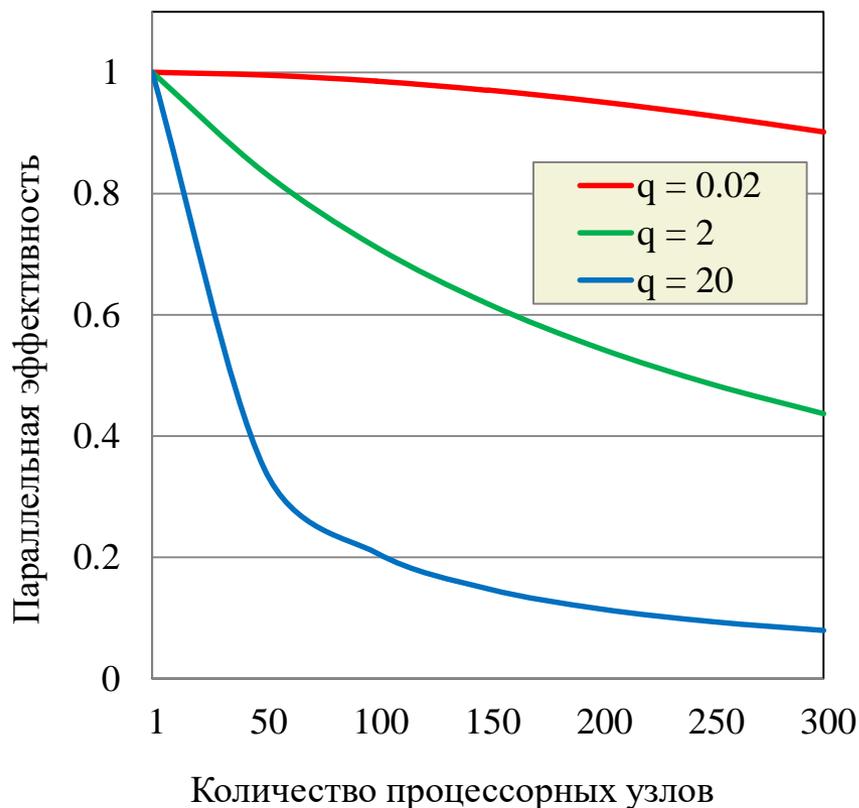
BSF-модель



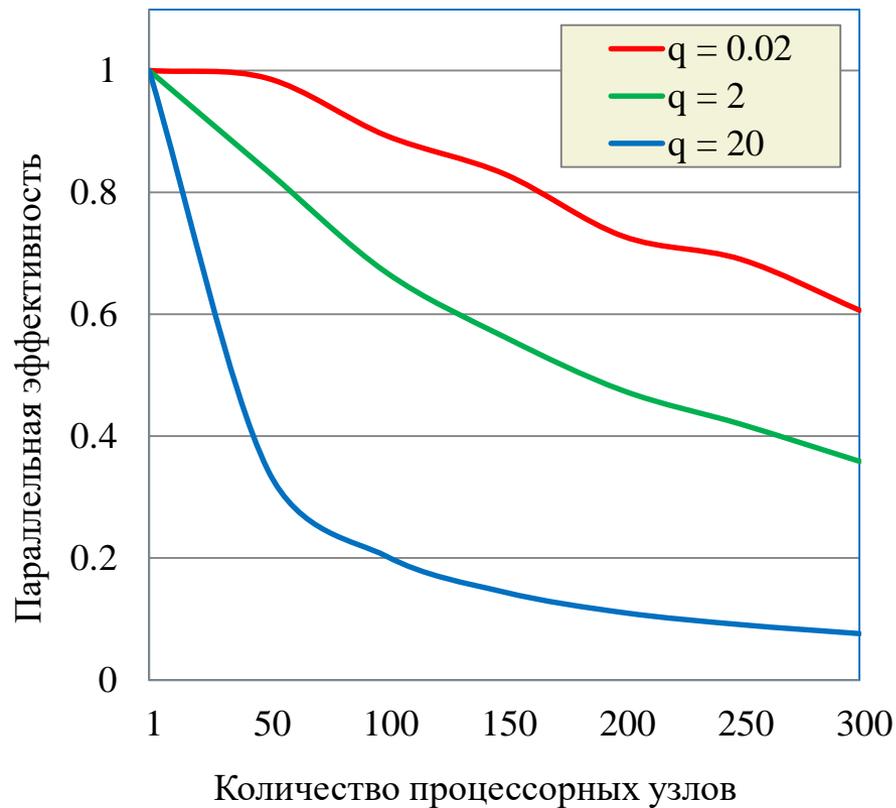
Эксперимент

Параллельная эффективность

$$(q = t_r + t_p)$$



BSF-модель



Эксперимент

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

Меняются все элементы матрицы A :

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

Меняется один элемент матрицы A :

$$K \leq O(n)$$

K – количество рабочих узлов

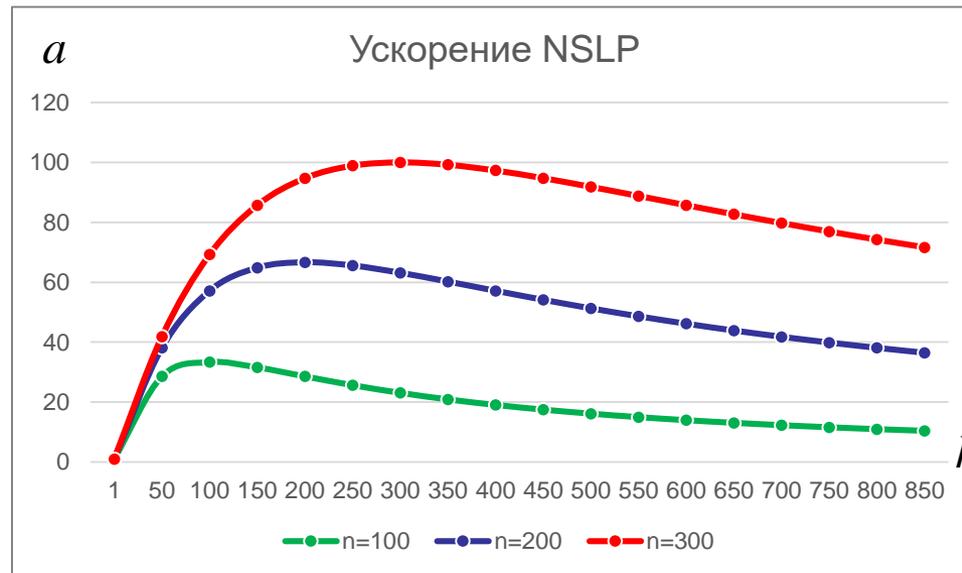
n – размерность задачи



Ускорение для BSF-реализации алгоритма NSLP

$$a(K) = \frac{O(n^3)K}{O(n)K^2 + O(n^2)K + O(n^3)}$$

(меняется один элемент матрицы A)



K – количество рабочих узлов

n – размерность задачи

Параллельная эффективность BSF-реализации алгоритма NSLP

Меняются все элементы матрицы A :

$$e(K) = \frac{1}{1 + K^2/O(n)}$$

Меняется один элемент матрицы A :

$$e(K) = \frac{1}{1 + K^2/O(n^2) + K/O(n)}$$

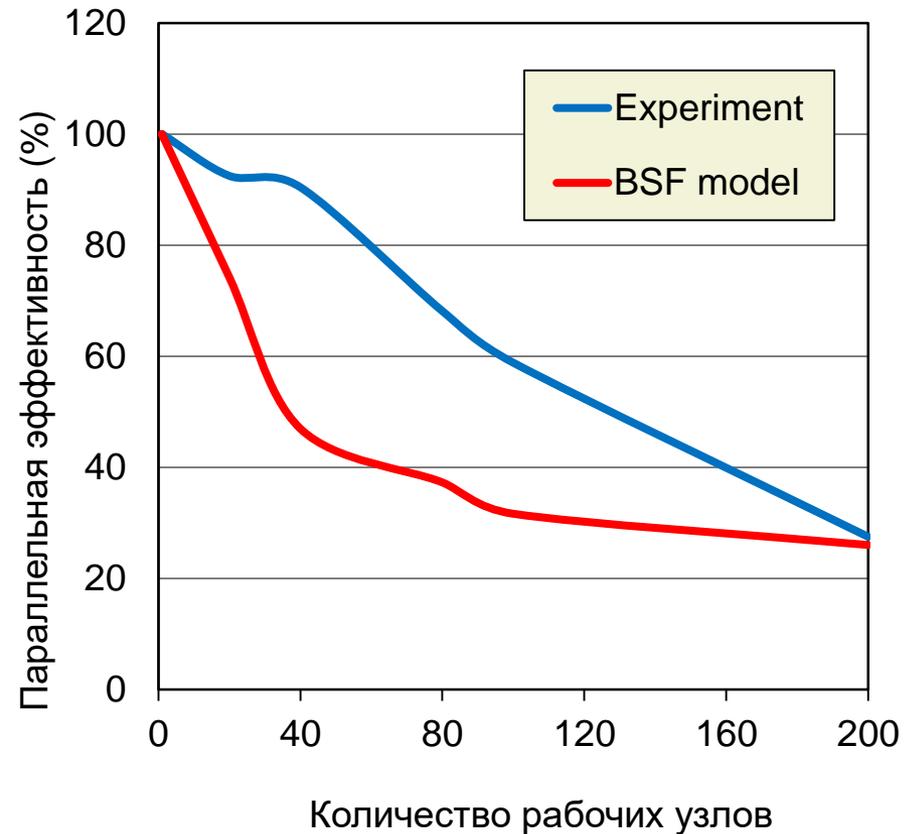
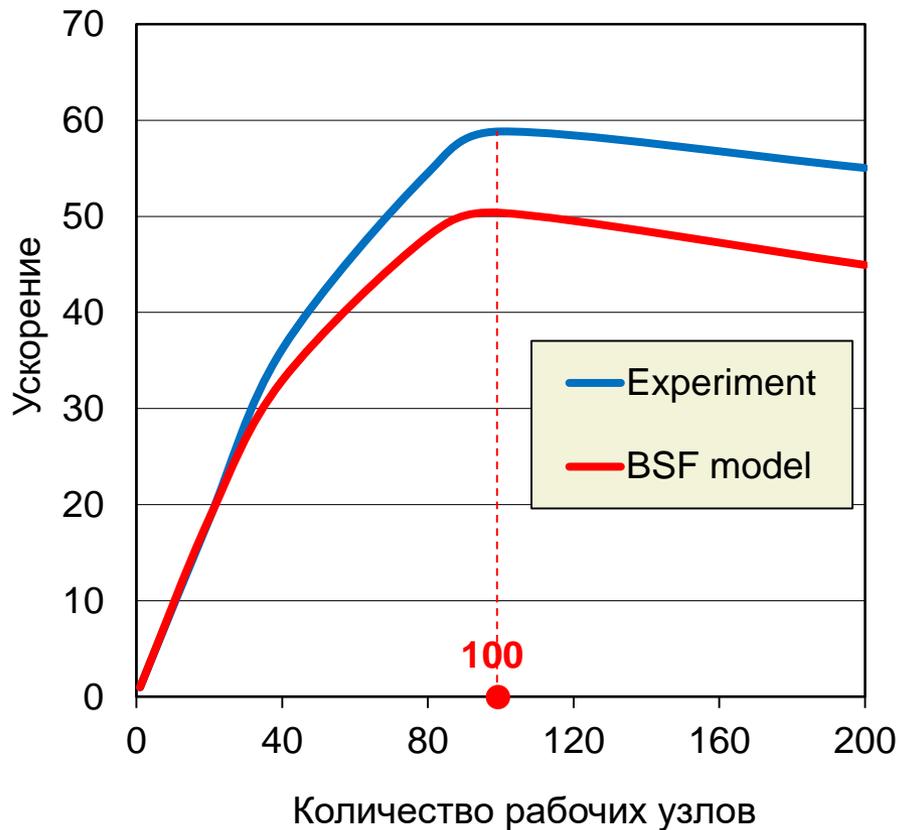
K – количество рабочих узлов

n – размерность задачи

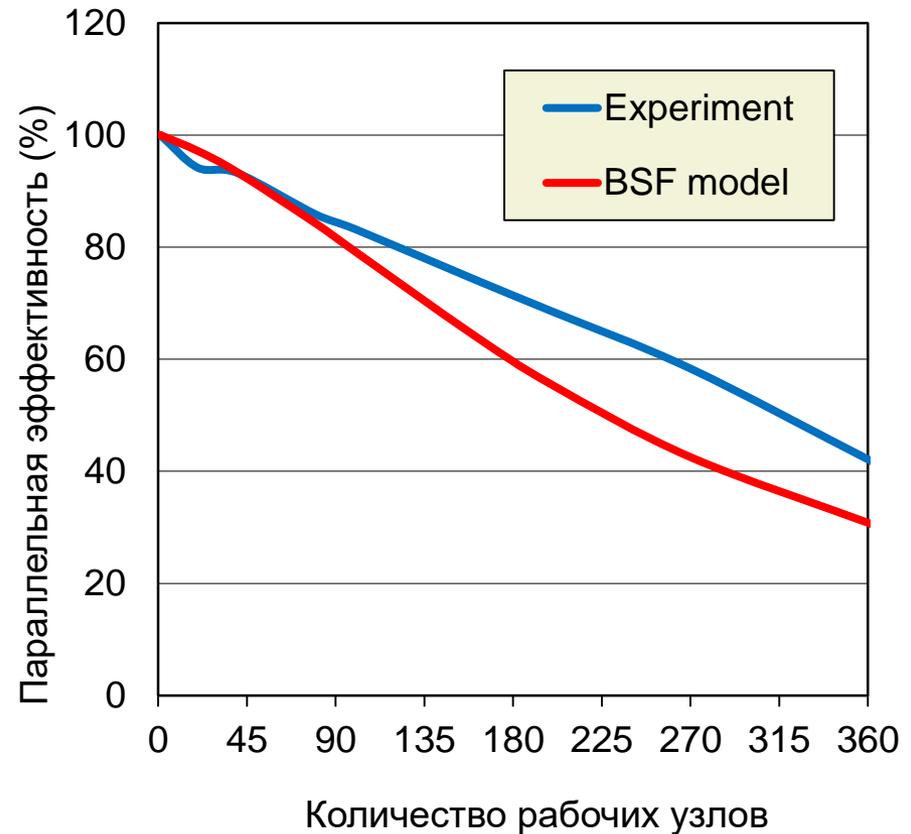
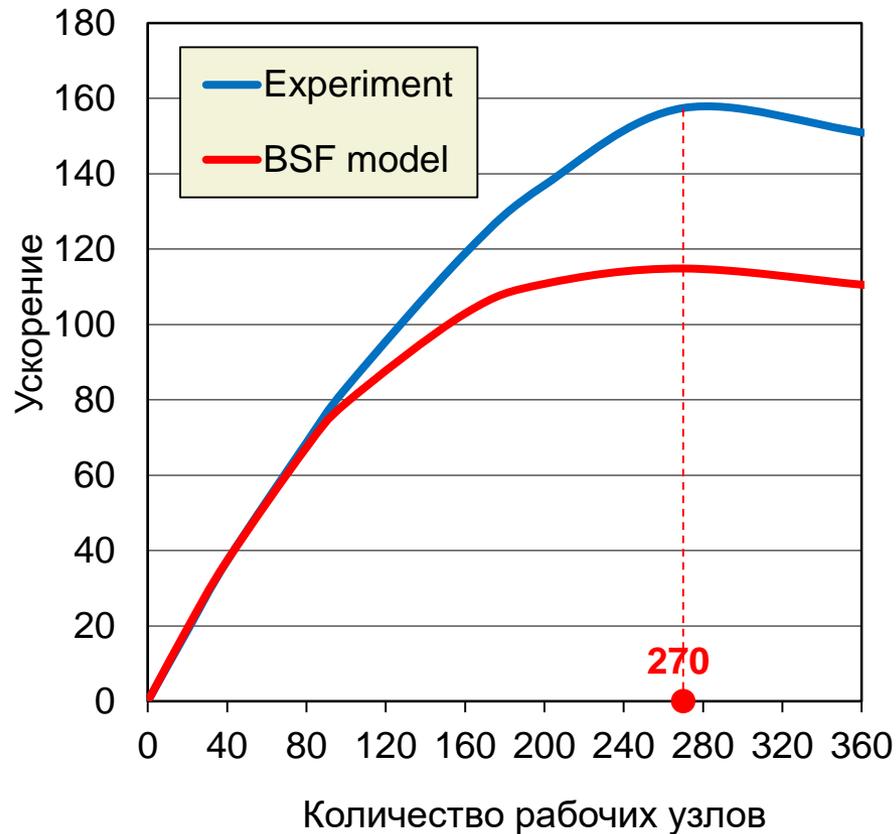
BSF-реализация алгоритма NSLP на C++ и MPI

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

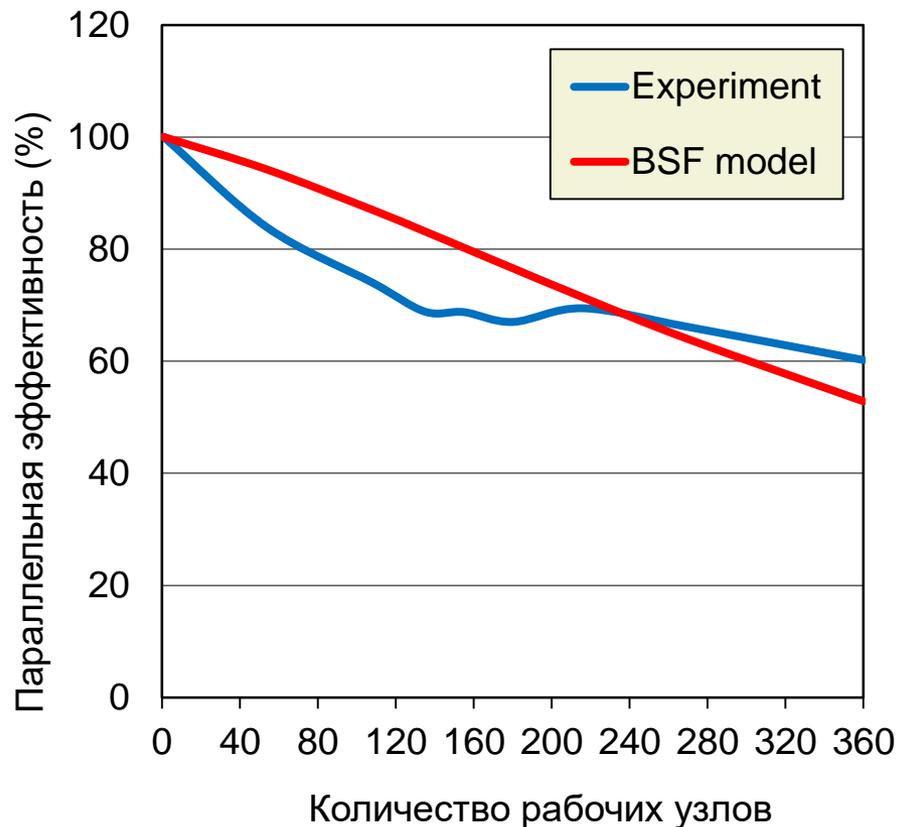
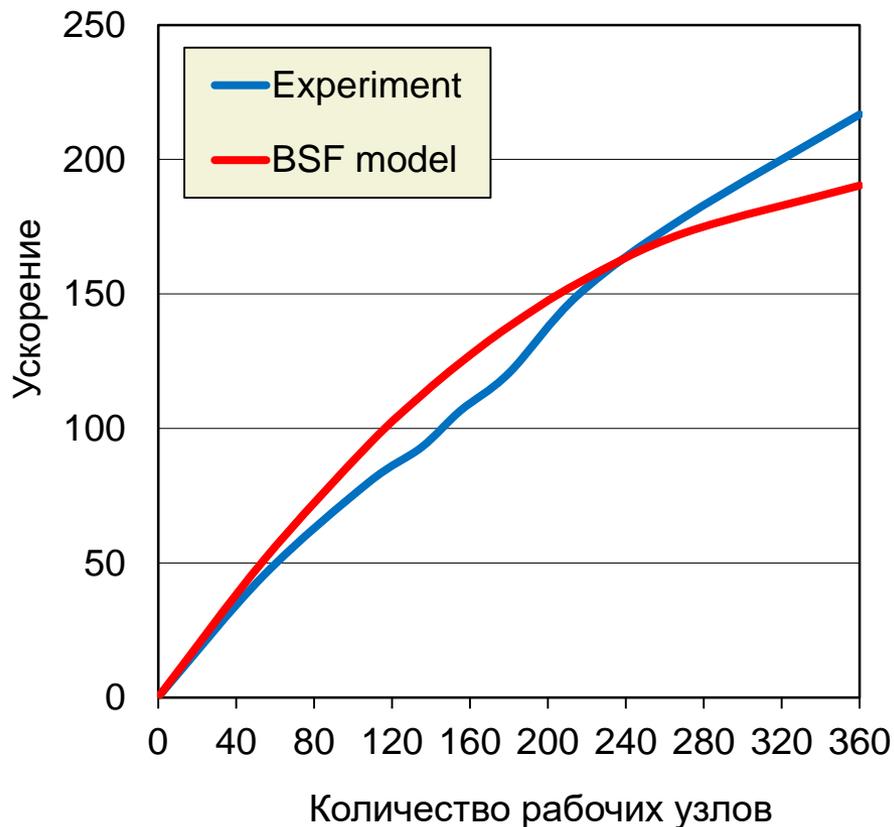
Сравнение теории с экспериментом ($n = 400$)



Сравнение теории с экспериментом ($n = 800$)



Сравнение теории с экспериментом ($n = 1080$)



План дальнейших исследований

- Разработка программного BSF-каркаса
- Верификация BSF-модели на большом количестве численных итерационных алгоритмов
- Разработка интеллектуального конструктора BSF-программ «BSF Studio»

Библиография

1. *Sokolinsky L.B.* Analytical Estimation of the Scalability of Iterative Numerical Algorithms on Distributed Memory Multiprocessors // Lobachevskii Journal of Mathematics. 2018. Vol. 39, no. 4. <https://arxiv.org/pdf/1710.10490.pdf>
2. *Sokolinsky L.B.* Analytical study of the “master-worker” framework scalability on multiprocessors with distributed memory [Electronic resource] // arXiv:1704.05816 [cs.DC]. 2017. P. 15. (на русском). <http://arxiv.org/abs/1704.05816>
3. *Ezhova N.* Verification of BSF Parallel Computational Model // 3rd Ural Workshop on Parallel, Distributed, and Cloud Computing for Young Scientists (Ural-PDC). CEUR Workshop Proceedings. Vol. 1990. CEUR-WS.org. P. 30–39. <http://ceur-ws.org/Vol-1990/paper-04.pdf>
4. *Sokolinskaya I., Sokolinsky L.B.* Scalability Evaluation of NSLP Algorithm for Solving Non-Stationary Linear Programming Problems on Cluster Computing Systems // Communications in Computer and Information Science. 2017. Vol. 793. P. 40-53. DOI: 10.1007/978-3-319-71255-0_4. <https://arxiv.org/abs/1709.04640>

Спасибо за внимание!

Вопросы?