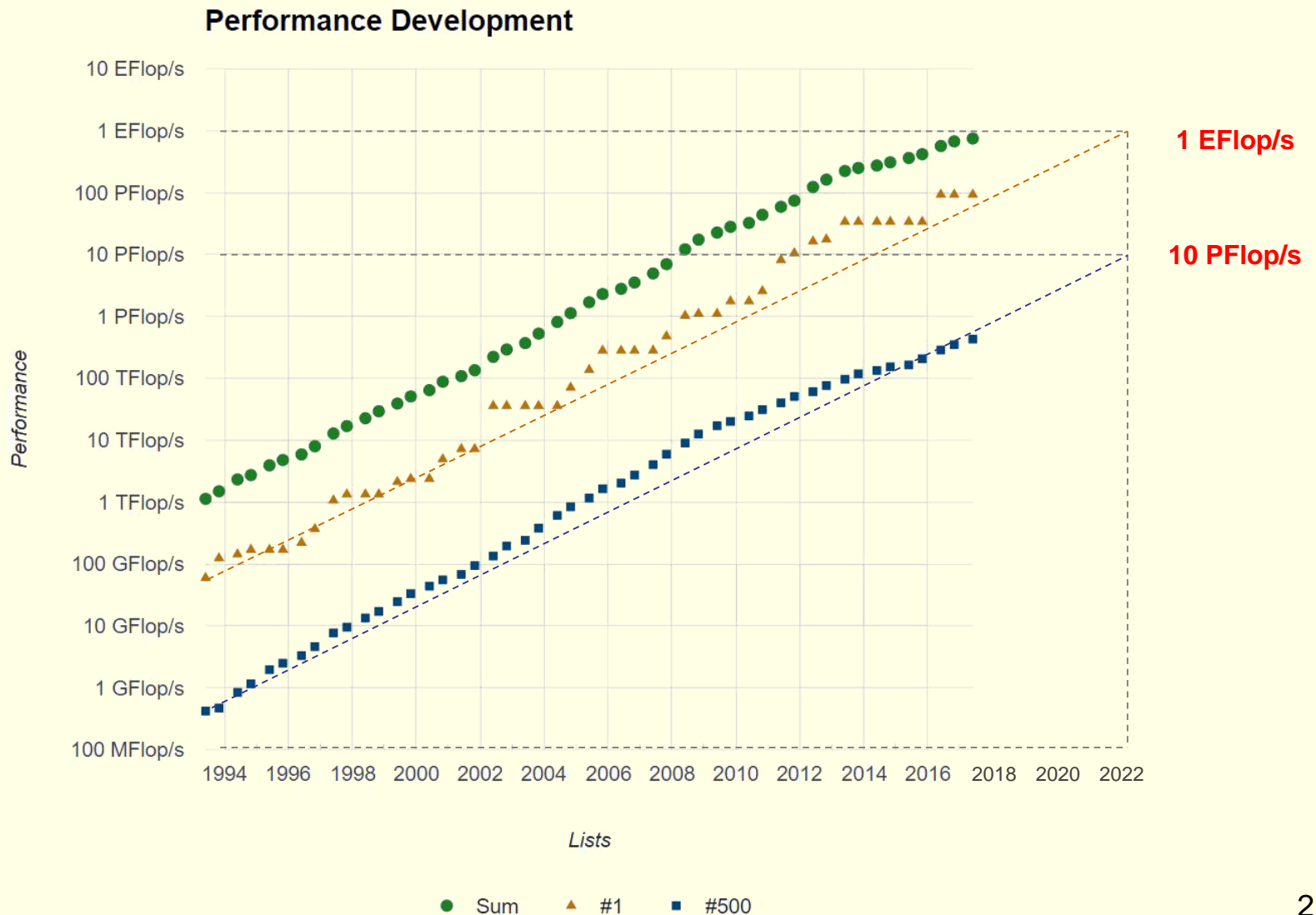




# Исследование масштабируемости алгоритма NSLP для решения нестационарных задач линейного программирования на кластерных вычислительных системах

д.ф.-м.н. Л.Б. Соколинский, к.ф.-м.н. И.М. Соколинская  
Южно-Уральский государственный университет  
(национальный исследовательский университет)

# TOP500 (июнь 2017)

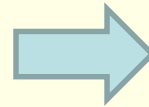


# Изменение основной парадигмы дизайна алгоритма

---

Разработать алгоритм,  
эффективно работающий  
на малых вычислительных  
ресурсах

1971

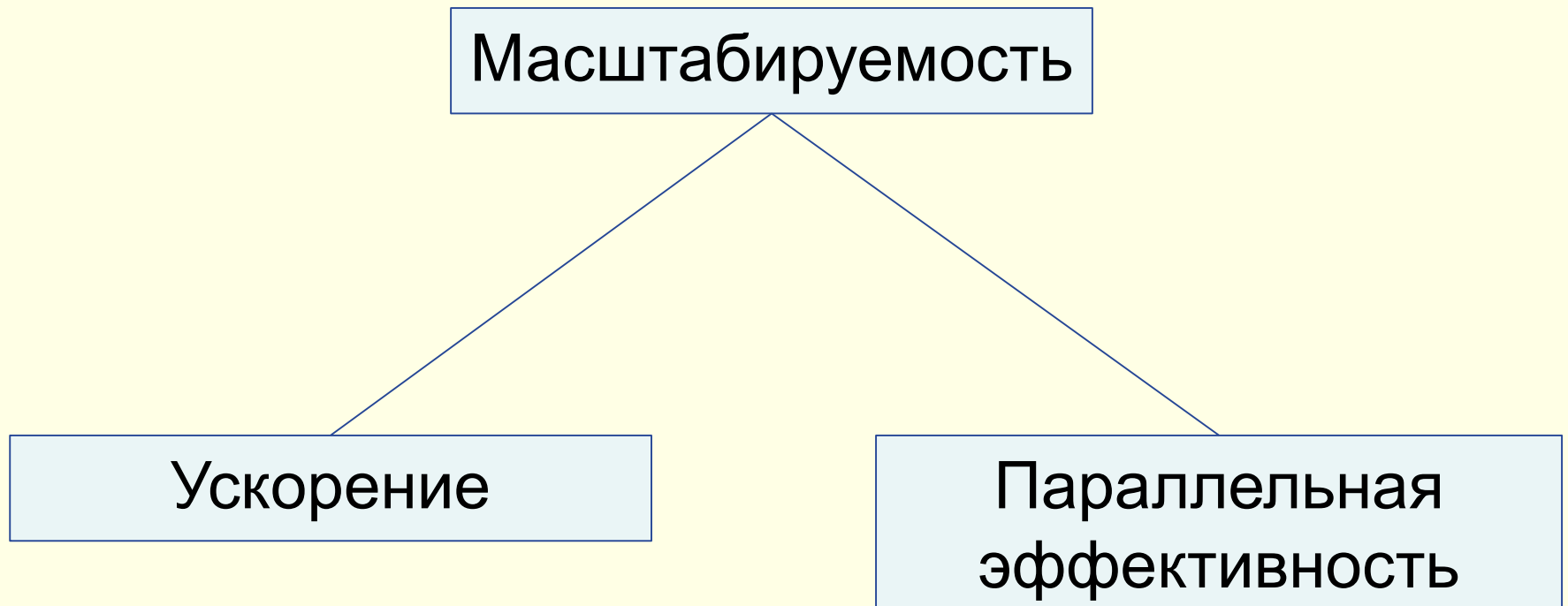


Разработать алгоритм,  
способный эффективно  
загрузить большие  
вычислительные ресурсы

2017

# Главная характеристика современного алгоритма

---



# Ускорение

---

$$a(k) = \frac{t_1}{t_k}$$

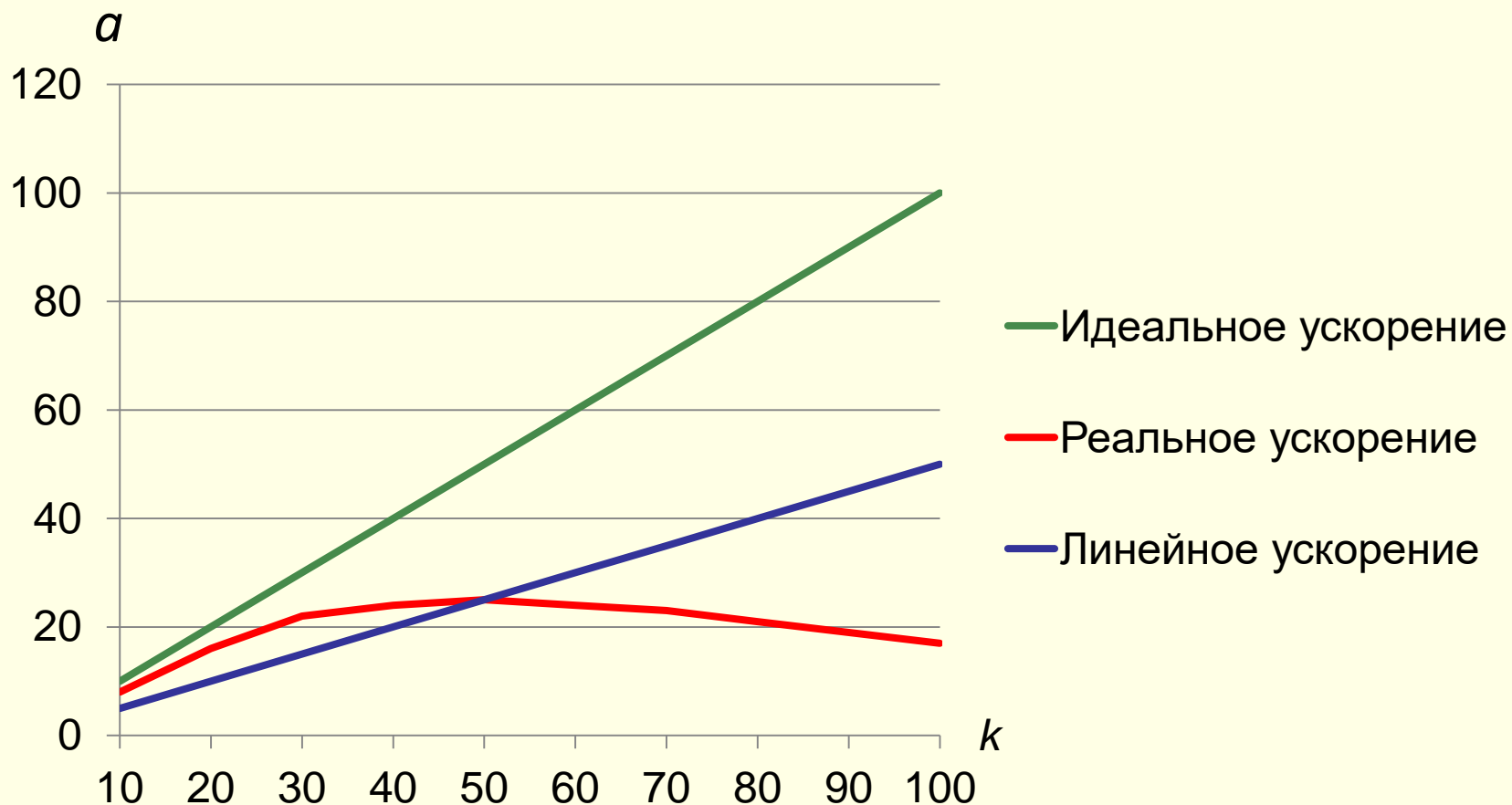
Количество узлов

Время решения задачи на 1 узле

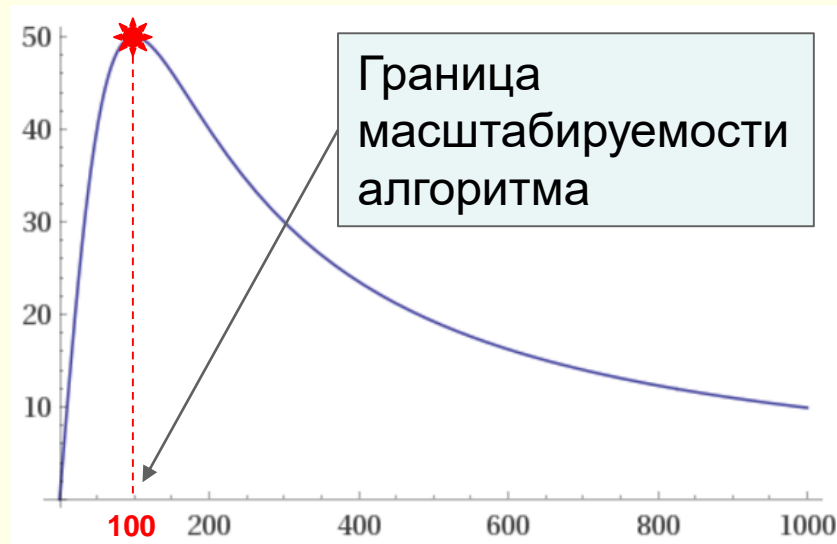
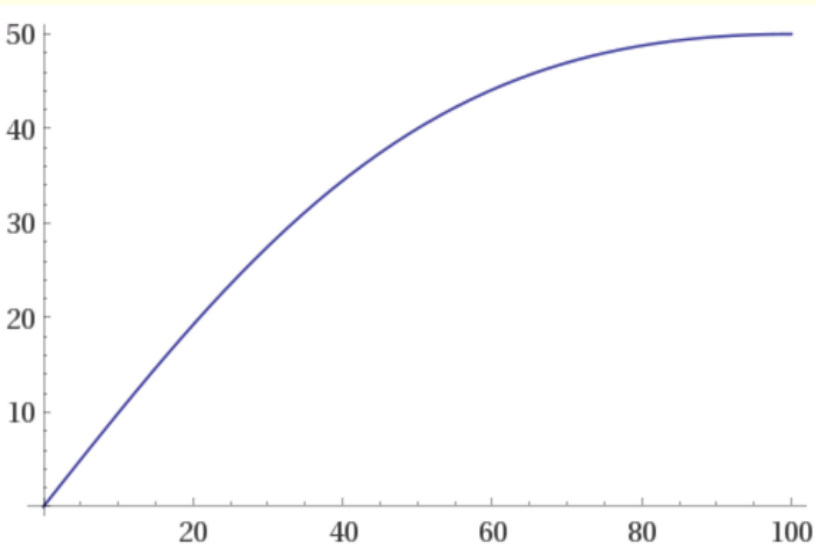
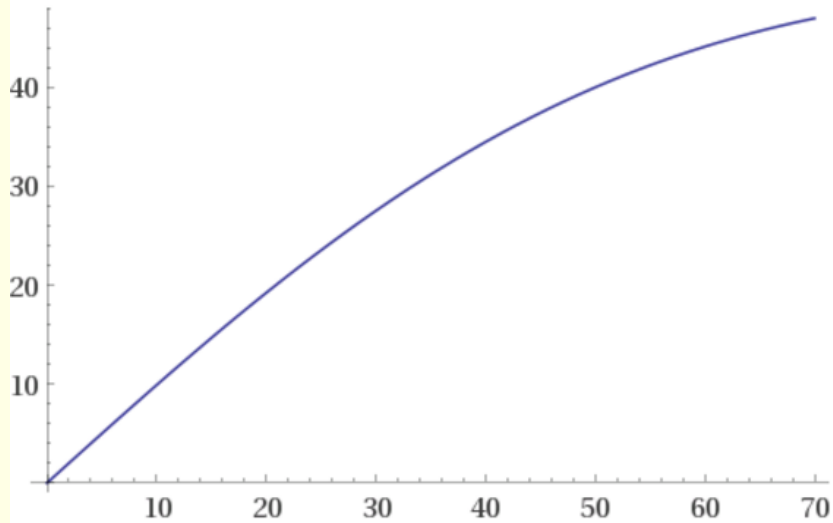
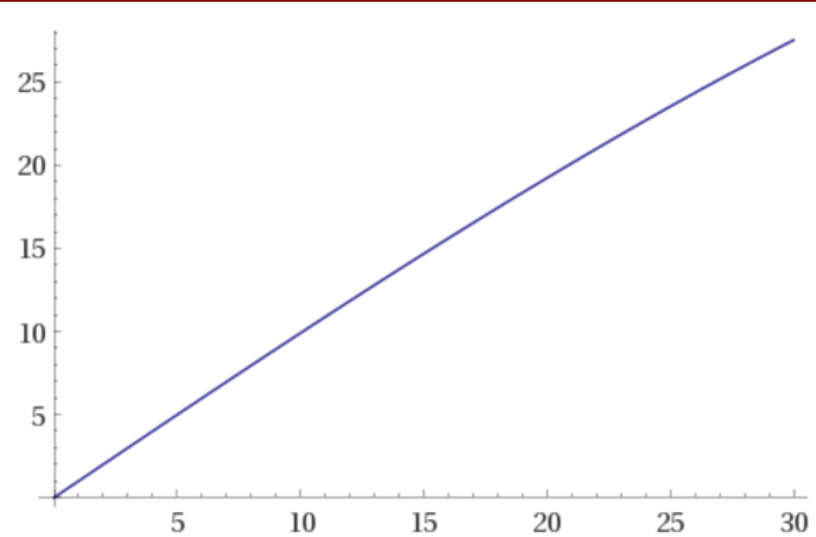
Время решения задачи на  $k$  узлах

# Ускорение: идеальное, линейное, реальное

---



# Ускорение реальной задачи на кластерной вычислительной системе



# Параллельная эффективность

---

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

Ускорение

Количество узлов

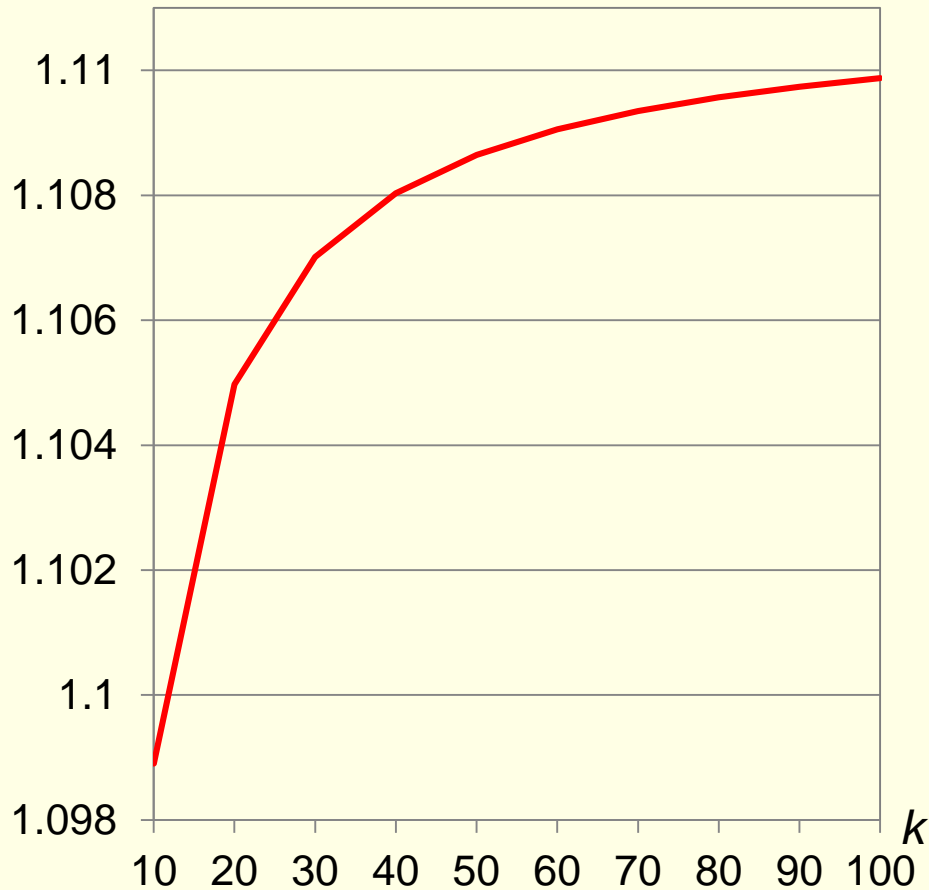
The diagram illustrates the formula for parallel efficiency,  $e(k) = \frac{a(k)}{k}$ . A light blue box labeled "Ускорение" (Acceleration) has an arrow pointing to the numerator  $a(k)$ . Another light blue box labeled "Количество узлов" (Number of nodes) has an arrow pointing to the denominator  $k$ .



# 10% параллельного кода

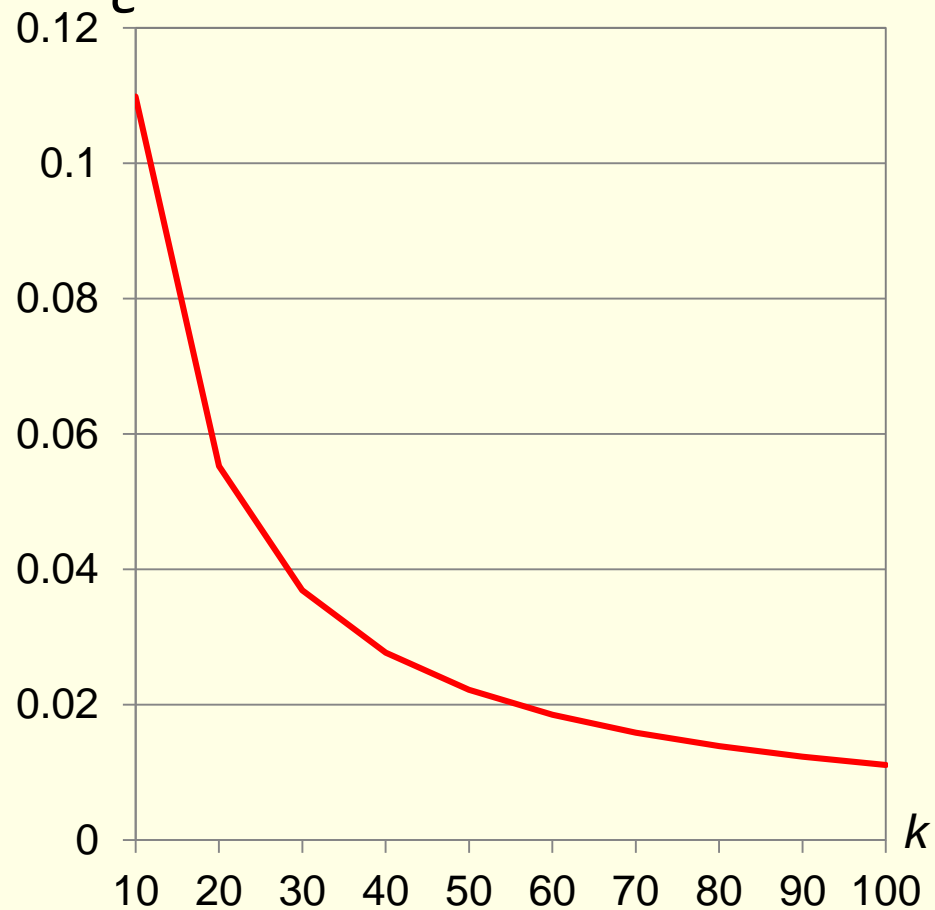
*Ускорение*

*a*

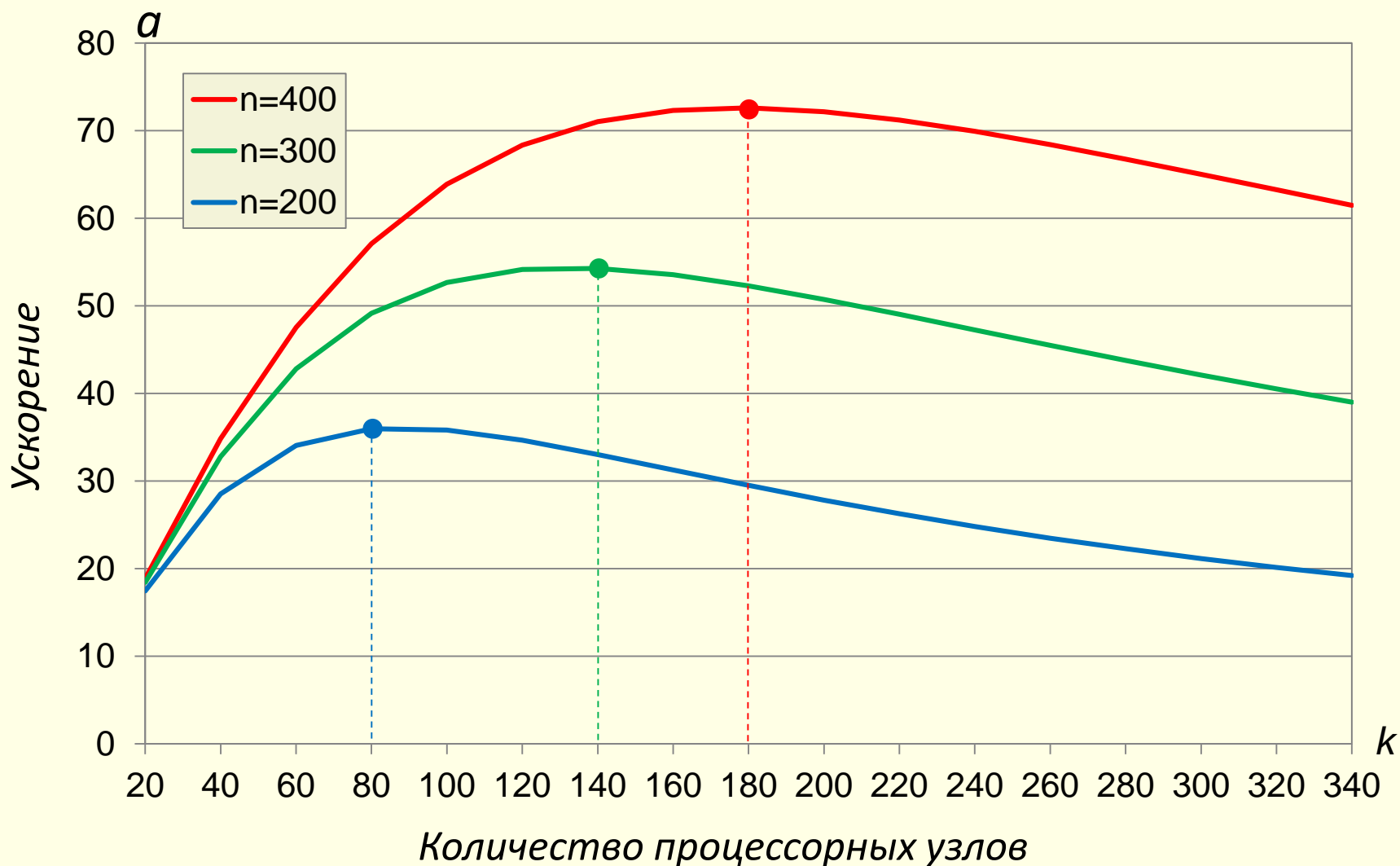


*Параллельная эффективность*

*e*

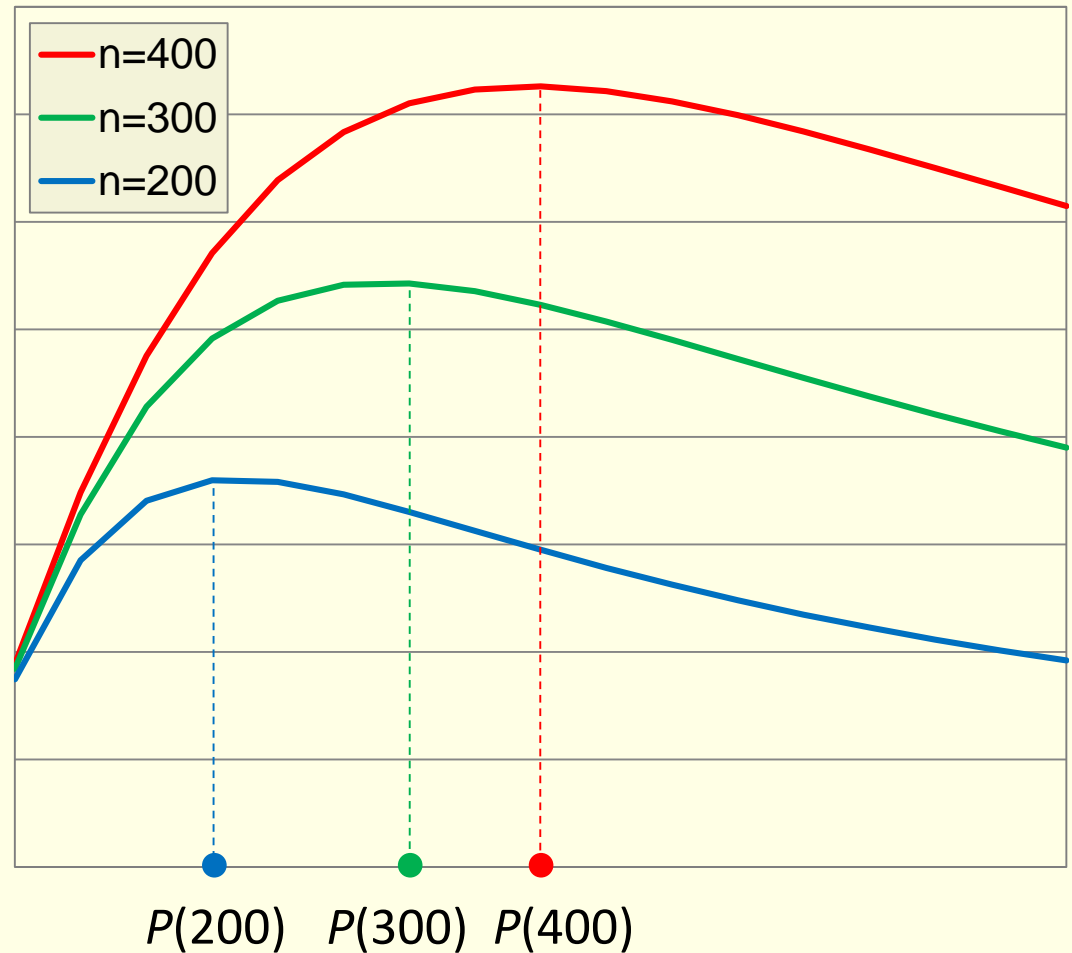


# Размер задачи имеет значение!



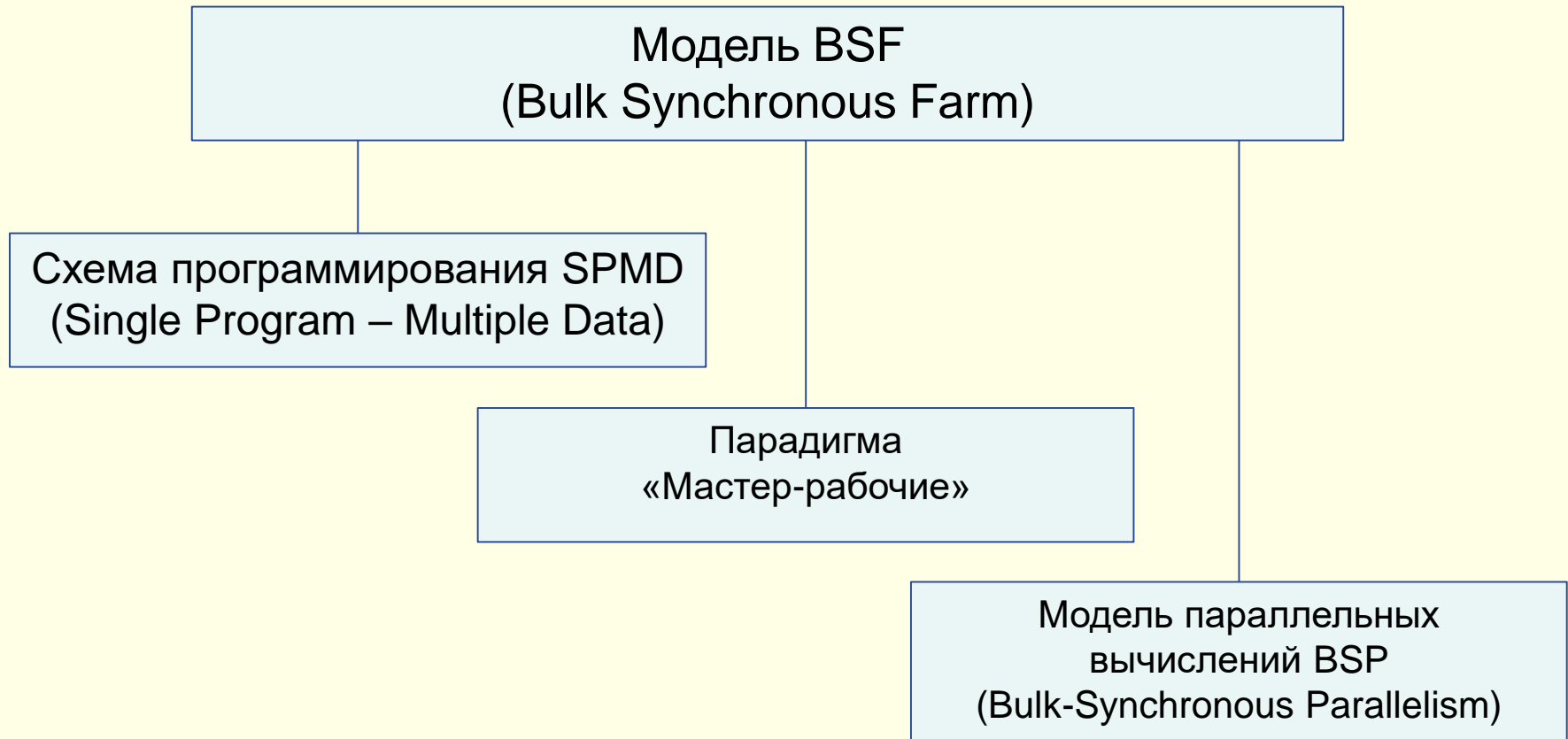
# Необходима формула для оценка масштабируемости алгоритма **ДО** написания программы

$$k = P(n)$$



# Модель параллельных вычислений BSF

---



# Область применения модели BSF

---

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

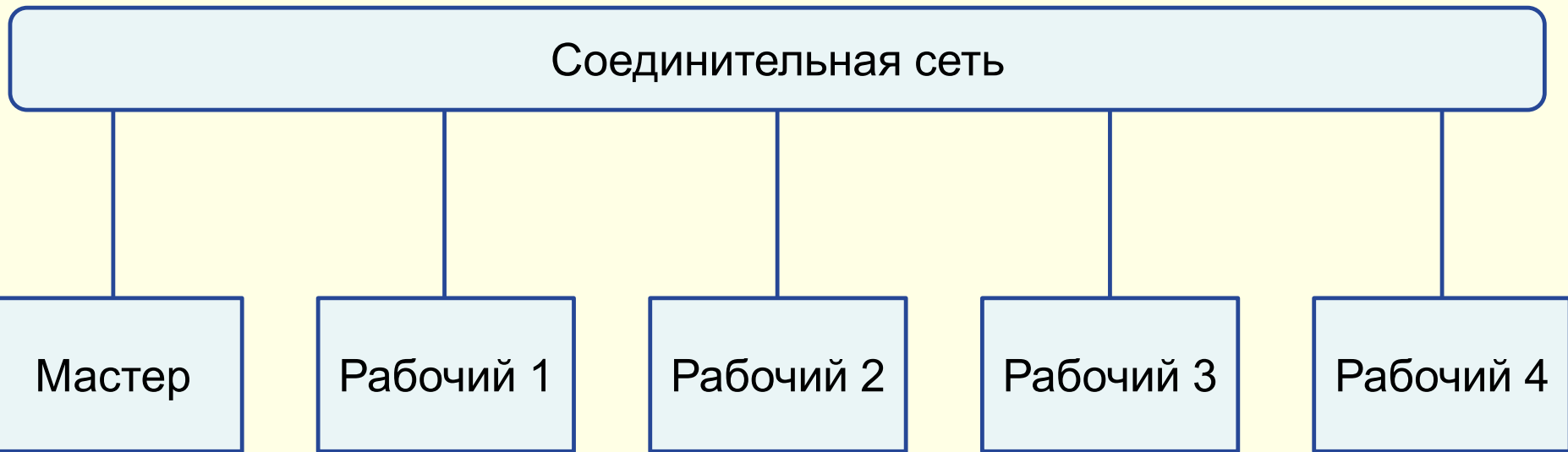
# Модель BSF позволяет предсказать:

---

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

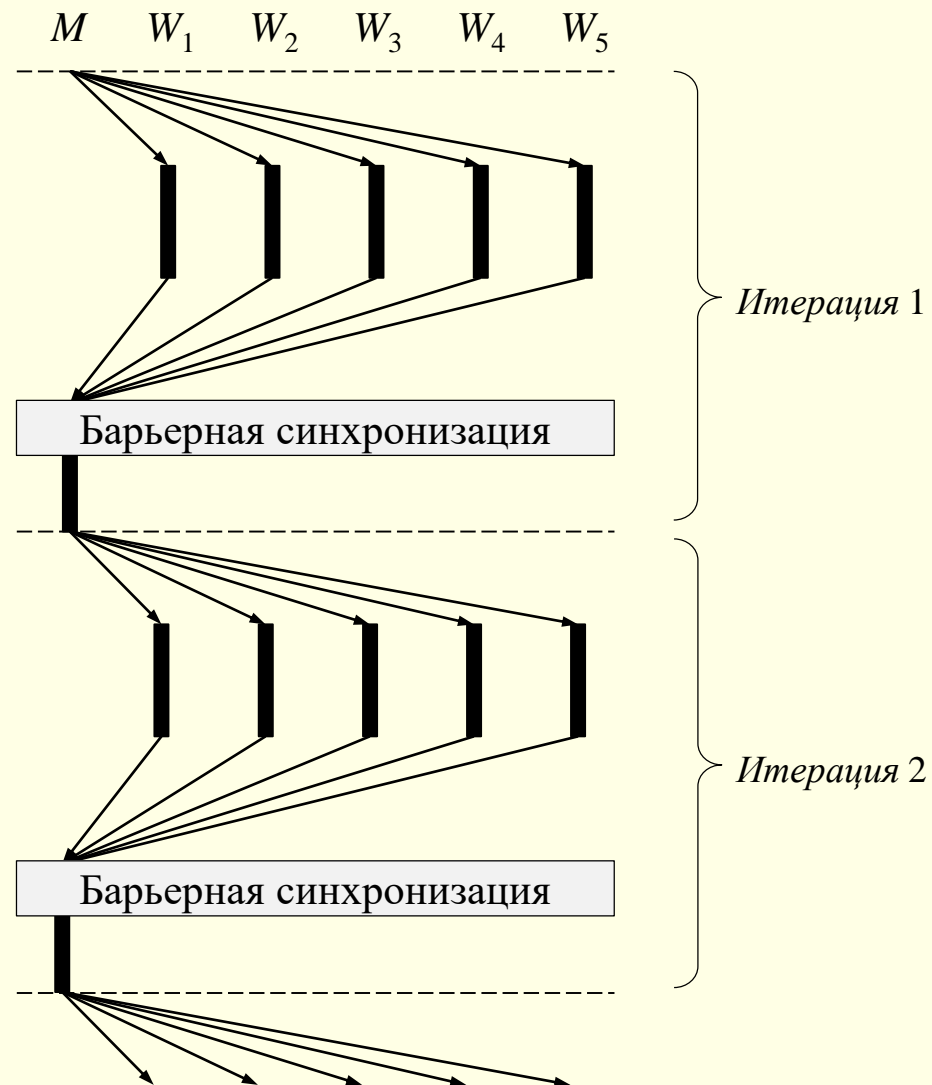
# BSF-компьютер

---



Процессорные узлы

# Схема работы *BSF*-алгоритма



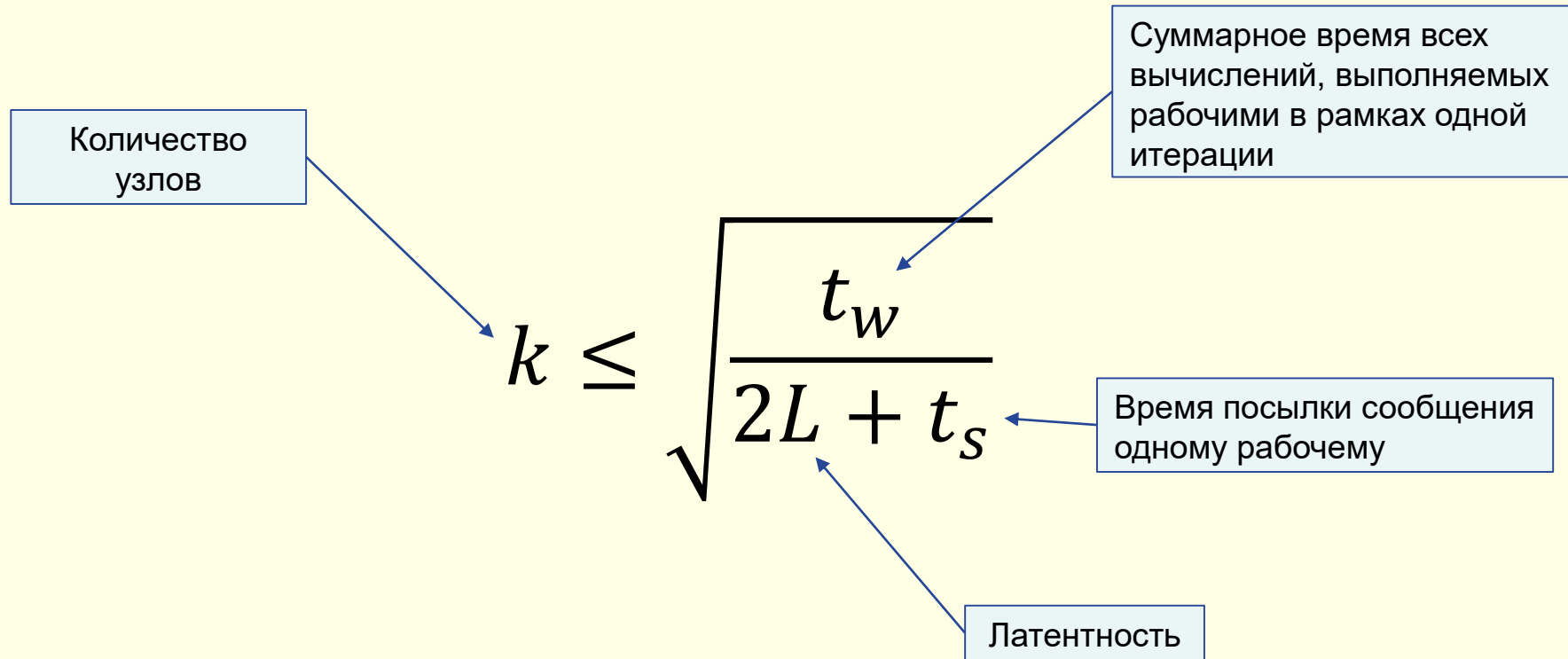


# Параметры BSF-модели

---

- $t_s$  – время, необходимое для посылки сообщения одному рабочему
- $L$  – латентность (затраты на инициализацию операции передачи сообщения)
- $t_w$  – 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

# 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

# Ускорение BSF-алгоритма

---

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

$k$  – количество процессорных узлов

$t_s$  – время посылки сообщения одному рабочему

$L$  – латентность

$t_w$  – время вычислений, выполняемых рабочими

$t_r$  – время передачи результатов мастеру от рабочих

$t_p$  – время обработки результатов, полученных от рабочих

# Параллельная эффективность BSF-алгоритма

---

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

$k$  – количество процессорных узлов

$t_s$  – время для посылки сообщения одному рабочему

$L$  – латентность

$t_w$  – время вычислений, выполняемых рабочими

$t_r$  – время передачи результатов мастеру от рабочих

$t_p$  – время обработки результатов, полученных от рабочих

# Применение модели BSF для исследования алгоритма NSLP

---

Нестационарная задача линейного программирования

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

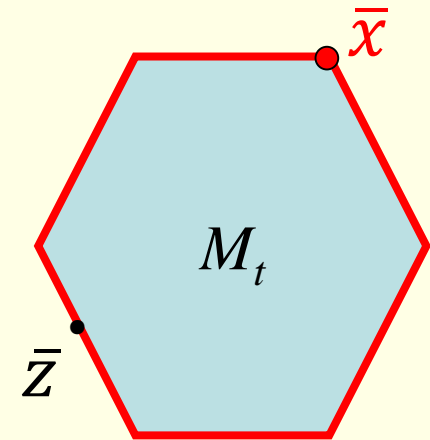
- $x \in \mathbb{R}_n$
- $A_t$  – матрица  $m \times n$
- $c_t, b_t$  – векторы размерности  $n$
- $t \in \mathbb{R}_{\geq 0}$  – время

# Алгоритм NSLP (Non Stationary Linear Programming)

---

## Фазы алгоритма:

- *Quest* – поиск точки  $\bar{z} \in M_t$
- *Targeting* – перемещение точки  $\bar{z}$  таким образом, чтобы точное решение  $\bar{x}$  задачи ЛП находилось в ее  $\varepsilon$ -окрестности



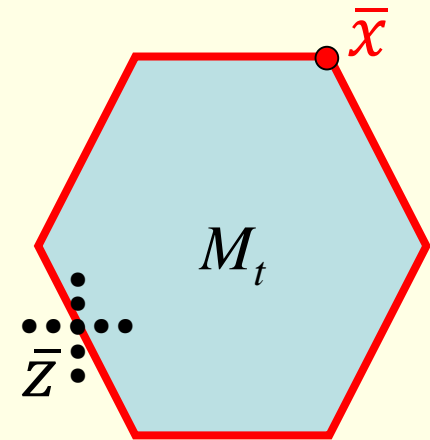
$$A_t x \leq b_t \Leftrightarrow x \in M_t$$

# Алгоритм NSLP (Non Stationary Linear Programming)

---

## Фазы алгоритма:

- *Quest* – поиск точки  $\bar{z} \in M_t$
- *Targeting* – перемещение точки  $\bar{z}$  таким образом, чтобы точное решение  $\bar{x}$  задачи ЛП находилось в ее  $\varepsilon$ -окрестности



$$A_t x \leq b_t \Leftrightarrow x \in M_t$$

# 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 процессорных узлов

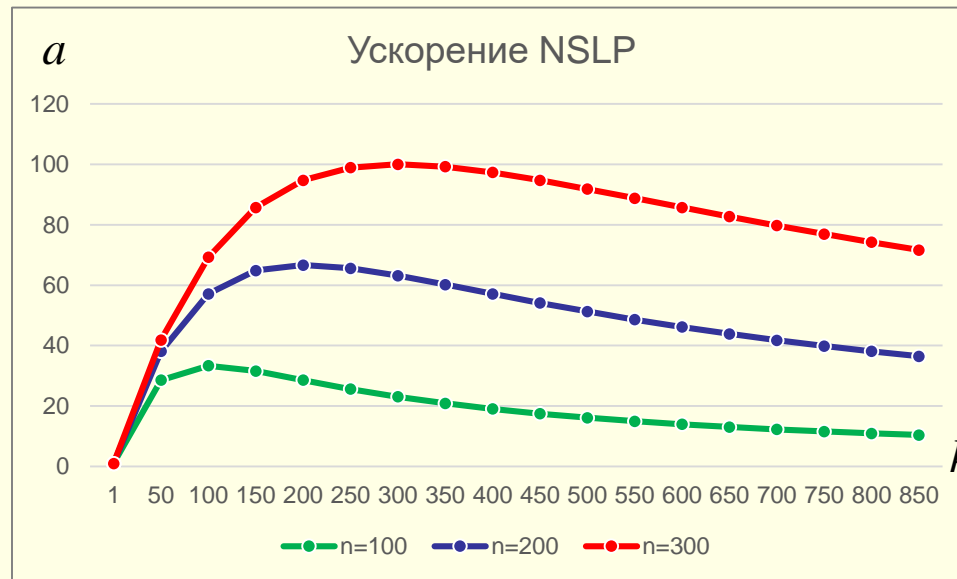
$n$  – размерность задачи



# Ускорение для BSF-реализации алгоритма NSLP

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

(меняется один элемент матрицы A)



$k$  – количество процессорных узлов

$n$  – размерность задачи

# Параллельная эффективность BSF-реализации алгоритма NSLP

---

Меняются все элементы матрицы A:

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

Меняется один элемент матрицы A:

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

$k$  – количество процессорных узлов

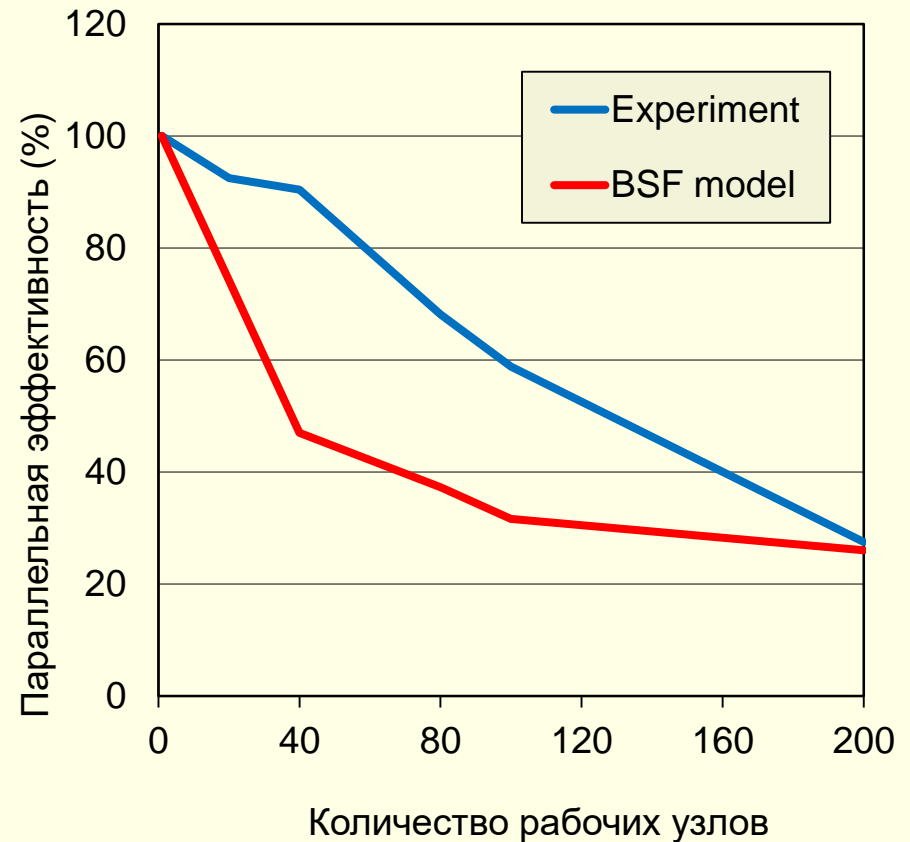
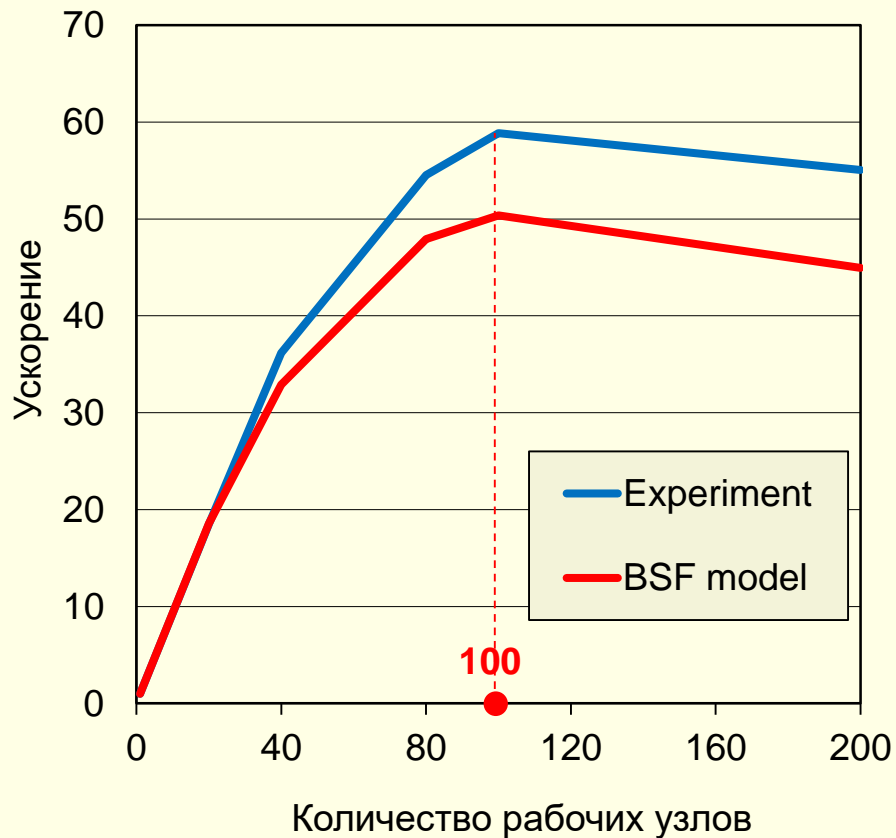
$n$  – размерность задачи

# BSF-реализация алгоритма NSLP на C++ и MPI

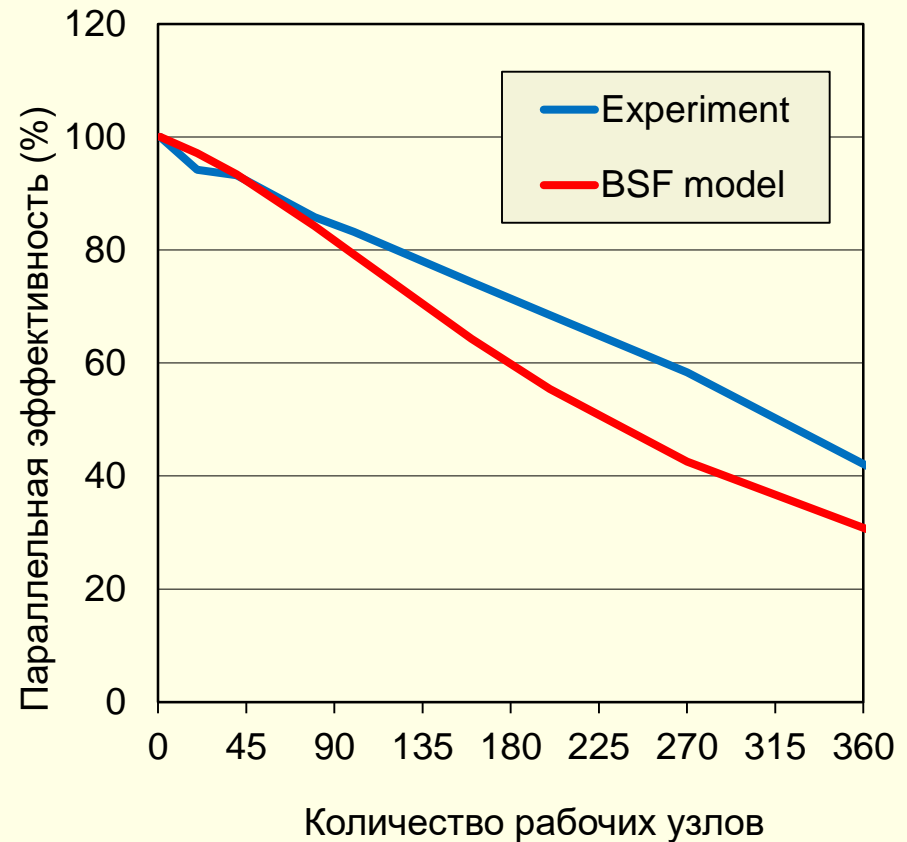
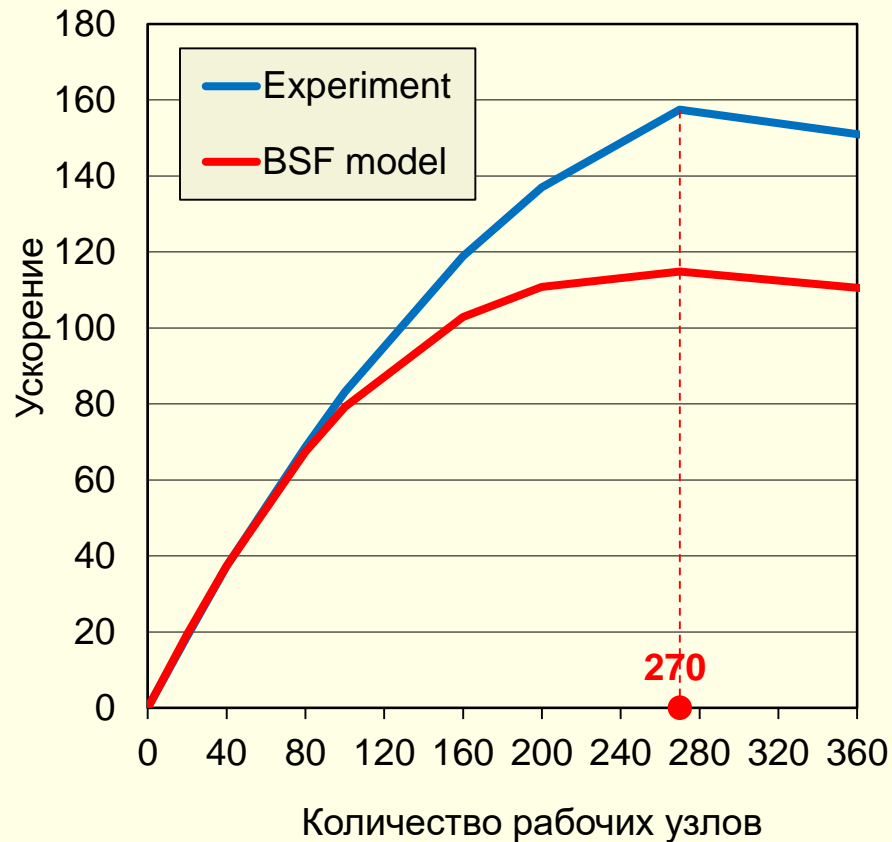
---

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

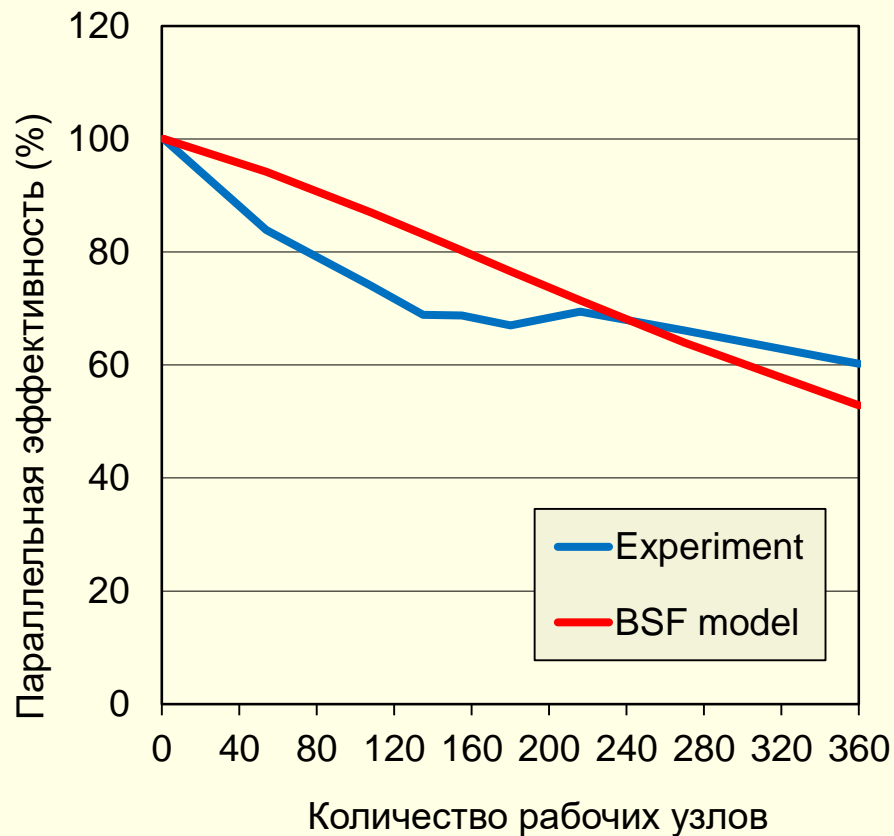
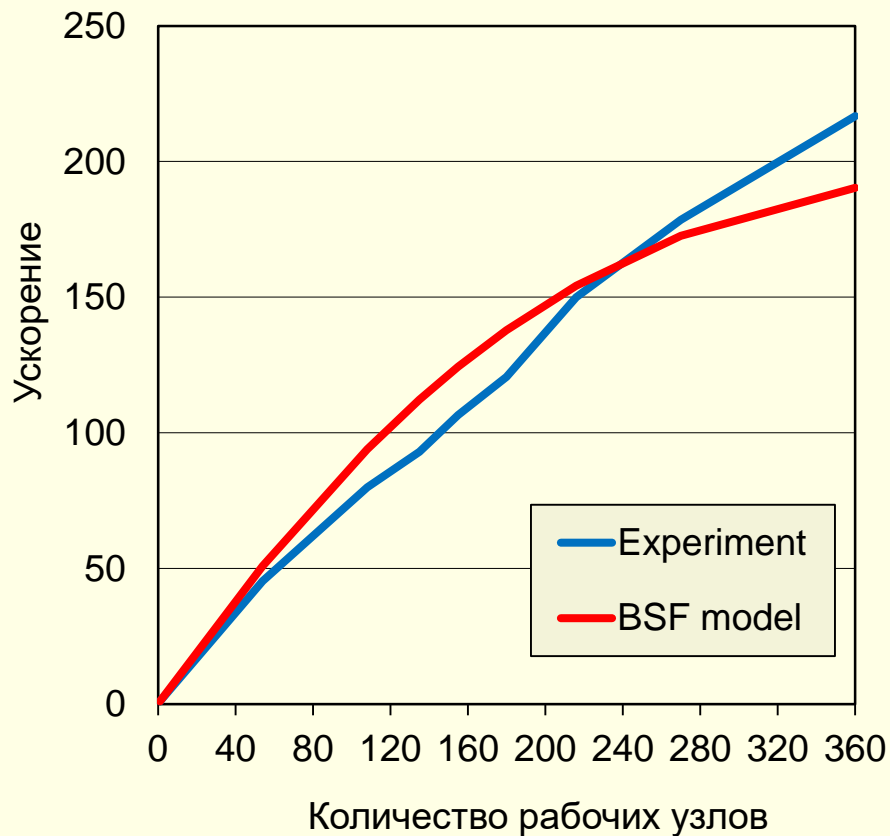
# Сравнение теории с экспериментом ( $n = 400$ )



# Сравнение теории с экспериментом ( $n = 800$ )



# Сравнение теории с экспериментом ( $n = 1080$ )



# Библиография

---

- *Sokolinsky L.B.* Analytical study of the “master-worker” framework scalability on multiprocessors with distributed memory [Electronic resource] // Submitted to “Parallel computational technologies (PCT) 2018”. arXiv:1704.05816 [cs.DC]. 2017. P. 15. (in Russian). <http://arxiv.org/abs/1704.05816>
- *Соколинская И.М., Соколинский Л.Б.* О решении задачи линейного программирования в эпоху больших данных // Параллельные вычислительные технологии – XI международная конференция, ПаВТ’2017, г. Казань, 3–7 апреля 2017 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2017. С. 471-484. <http://omega.sp.susu.ru/pavt2017/short/014.pdf>  
[Презентация доклада в формате PDF](#)
- *Sokolinskaya I., Sokolinsky L.B.* Scalability Evaluation of NSLP Algorithm for Solving Non-Stationary Linear Programming Problems on Cluster Computing Systems [Electronic resource] // Submitted to “Russian Supercomputing Days 2017”. arXiv:1709.04640 [cs.DC]. 2017. <http://arxiv.org/abs/1709.04640>

# Приглашаю к сотрудничеству авторов параллельных итерационных алгоритмов

---

- Помогу
  - сделать BSF-дизайн алгоритма
  - получить аналитические оценки масштабируемости
  - быстро разработать BSF-реализацию на C++ и MPI на основе готового BSF-каркаса
  - получить бесплатное время для проведения вычислительных экспериментов на суперкомпьютере «Торнадо-ЮУрГУ»
- Приглашу в соавторы научной статьи о модели BSF



# Спасибо за внимание!

---

Вопросы?

# №1 в TOP500



## Sunway TaihuLight Supercomputer

National Supercomputing  
Center in Wuxi, China

- Ядер: **10 649 600**
- Память: **1 310 720 GB**
- Linpack: **93 PFlop/s**
- Узлов: **40 960**

# Структура BSF-программы

