Параллельные Вычислительные Технологии (ПаВТ'2019)
(Калининград, 2-4 апреля 2019 г.)

Верификация модели параллельных вычислений BSF (на примере гравитационной задачи)

Ежова Надежда Александровна,
Соколинский Леонид Борисович

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).

Модель параллельных вычислений

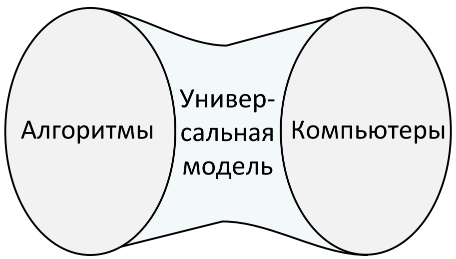
- Модель параллельных вычислений – это фреймворк (система правил и ограничений) для описания и анализа параллельных алгоритмов и программ
- Применение
  - Прогноз времени выполнения
  - Оценка масштабируемость алгоритма

Дерево моделей параллельных вычислений

Модели параллельных вычислений

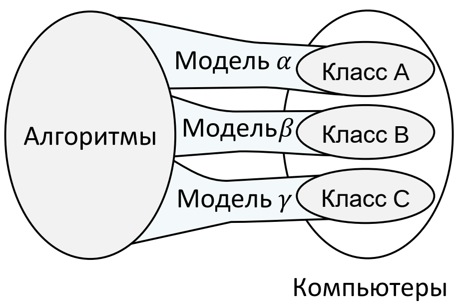
- PRAM 1979
  - QRQW PRAM 1998
  - BSPRAM 1998
  - PRAM-NUMA 2011
- BSP 1990
  - E-BSP 1996
  - BSP* 1998
  - HBSP 2000
  - HCGM 2000
  - D-BSP 2001
  - OBSP* 2004
  - Multi-BSP 2011
  - MBSP 2015
- LogP 1993
  - LogQ 1997
  - LogGP 1997
  - LogPQ 1997
  - PlogP 2000
  - LogGPS 2001
  - LoGPG 2001
  - LoGPC 2001
  - HLogGP 2006
    - LogfP 2006
      - $\log_n P$ 2007
      - $m\log_n P$ 2009
      - LogGPH 2010
      - mHLogGP 2012
      - $\mathrm{HLog}_n\mathrm{GP}$ 2015
- CGM 1996
- PRO 2002
- BSF 2018

Модель, соединяющая алгоритмы и компьютеры



\+ Универсальность

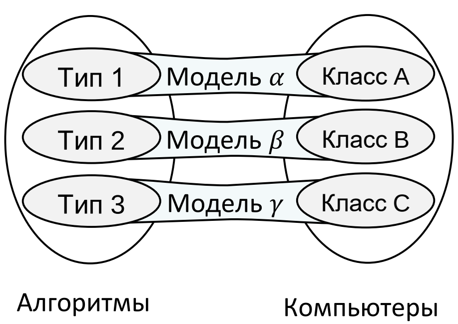
\- Невозможно создать ***адекватную*** универсальную модель для современных компьютеров



\+ Адекватность

± Ограниченная универсальность

\- Сложность практического применения



\+ Адекватность

\+ Простота использования

\- Универсальность

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

Ускорение – главная характеристика масштабируемости алгоритма

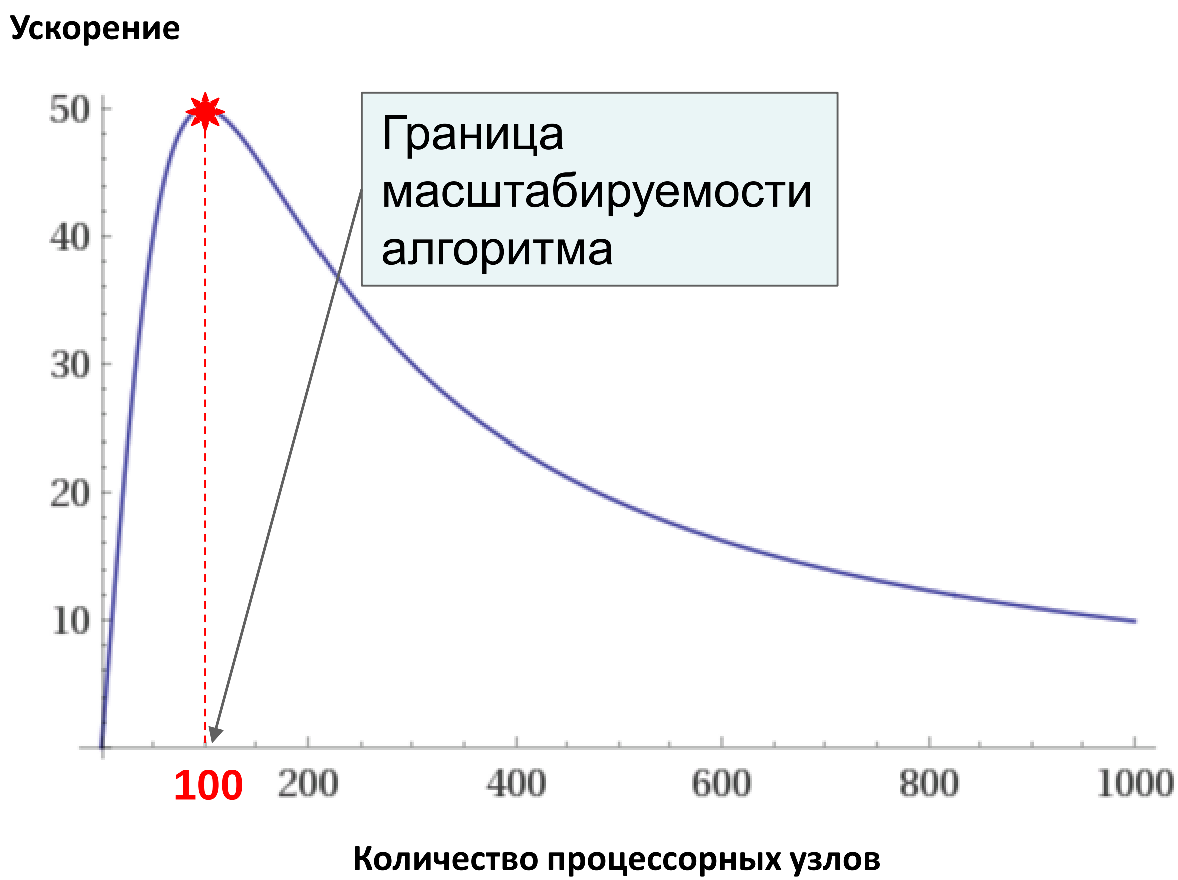
Время решения задачи на 1 узле

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

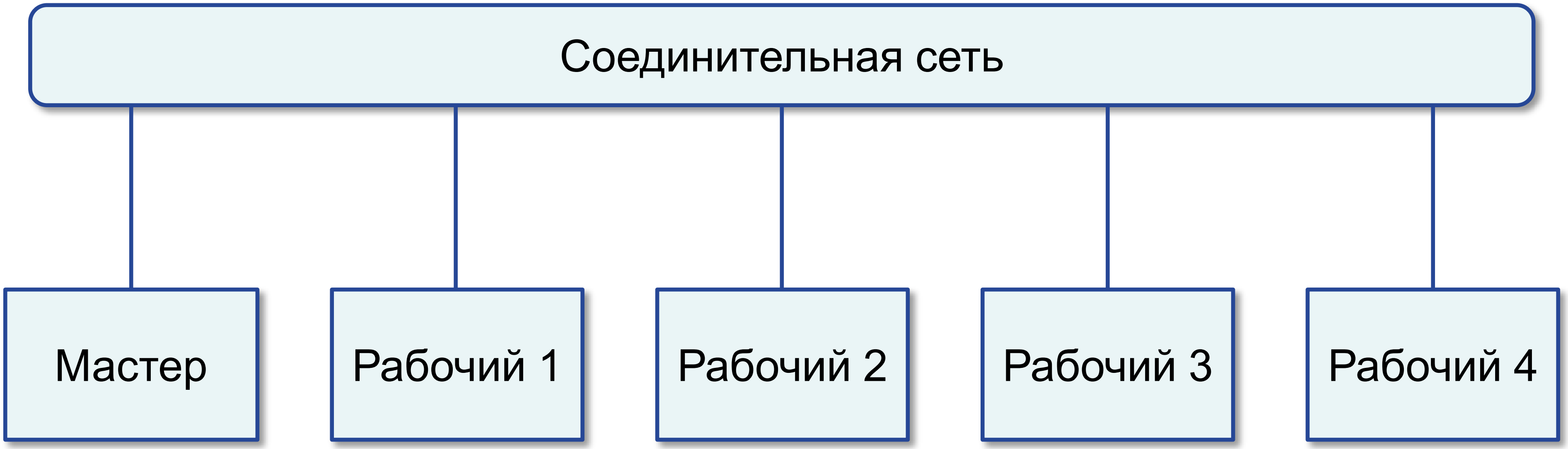
Время решения задачи на *K* узлах

Количество процессорных узлов

Граница масштабируемости алгоритма



BSF-компьютер

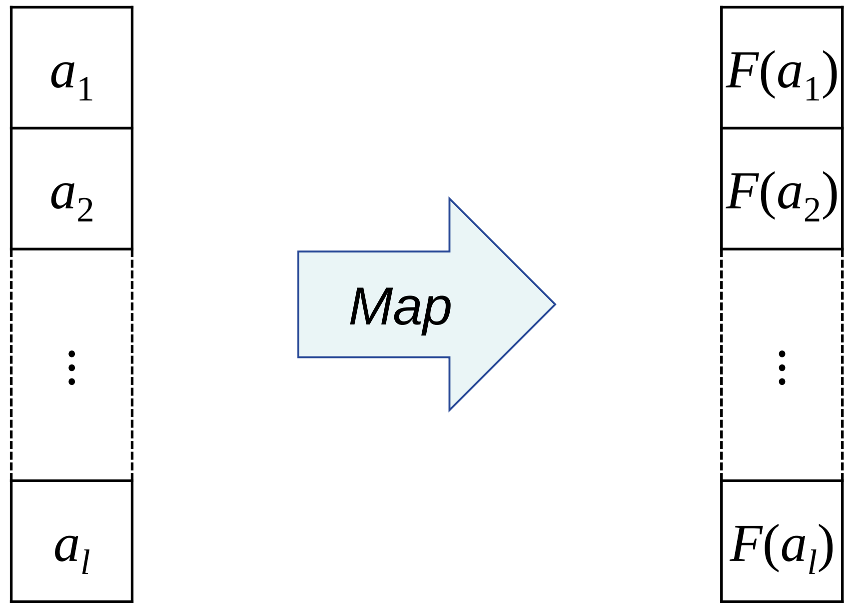


Процессорные узлы

Представление алгоритма в виде операций над списками

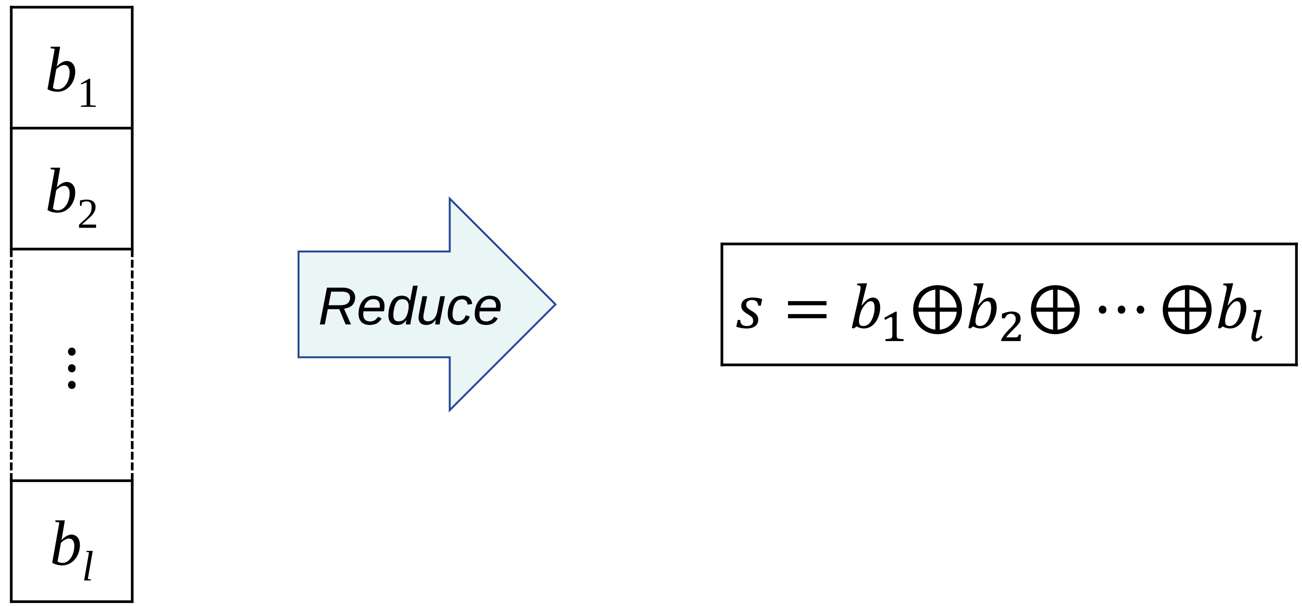
- Функции высшего порядка:
  – Map
  – Reduce

Функция высшего порядка *Map*



$$Map(F,[a_1,\dots,a_l]) = [F(a_1),\dots,F(a_l)]$$

Функция высшего порядка *Reduce*



$$Reduce(\oplus, [b_1, \ldots, b_l]) = b_1 \oplus \cdots \oplus b_l$$

Шаблон итерационного алгоритма в модели BSF

1. $i \coloneqq 0; Input(A, x_0)$
2. $B \coloneqq Map\left(F_{x_i}, A\right)$
3. $s \coloneqq Reduce(\oplus, B)$
4. $x_{i+1} \coloneqq Compute(x_i, s)$; $i \coloneqq i + 1$
5. **if** $StopCond(x_{i-1}, x_i)$ **go to** 7
6. **go to** 2
7. $Output(x_i)$; **stop**

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

Шаблон параллельного алгоритма в модели BSF

Шаг	Мастер	Рабочий j (j=1,…,K)
1.	$i \coloneqq 0; Input(A, x_0)$	$Input(A, x_0)$
2.	$SendToWorkers(x_i)$	$RecvFromMaster(x_i)$
		$B^{[j]} \coloneqq Map\left(F_{x_i}, A^{[j]}\right)$
3.		$s^{(j)} \coloneqq Reduce\left(\oplus, B^{[j]}\right)$
	for j = 1 **to** K **do**	
	$RecvFromWorkers([s^{(1)}, \dots, s^{(K)}])$	$SendToMaster(s^{(j)})$
	end for	
	$s \coloneqq Reduce\left(\oplus, [s^{(1)}, \dots, s^{(K)}]\right)$	
4.	$x_{i+1} \coloneqq Compute(x_i, s);\ i \coloneqq i + 1$	
5.	**if** $StopCond(x_{i-1}, x_i)$ **go to** 7	
6.	**go to** 2	**go to** 2
7.	$Output(x_i)$; **stop**	

Параметры модели BSF

K – количество рабочих

l – длина обрабатываемого списка

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

t_s – время передачи сообщения от мастера рабочему (без учета латентности)

t_r – время передачи сообщения от рабочего мастеру (без учета латентности)

t_{Map} – время выполнения функции Map для всего списка исходных данных

t_{Reduce} – время выполнения операции $\oplus$

t_p – время на обработку результатов итерации

Ускорение в модели BSF

Время решения задачи 1 мастером и 1 рабочим

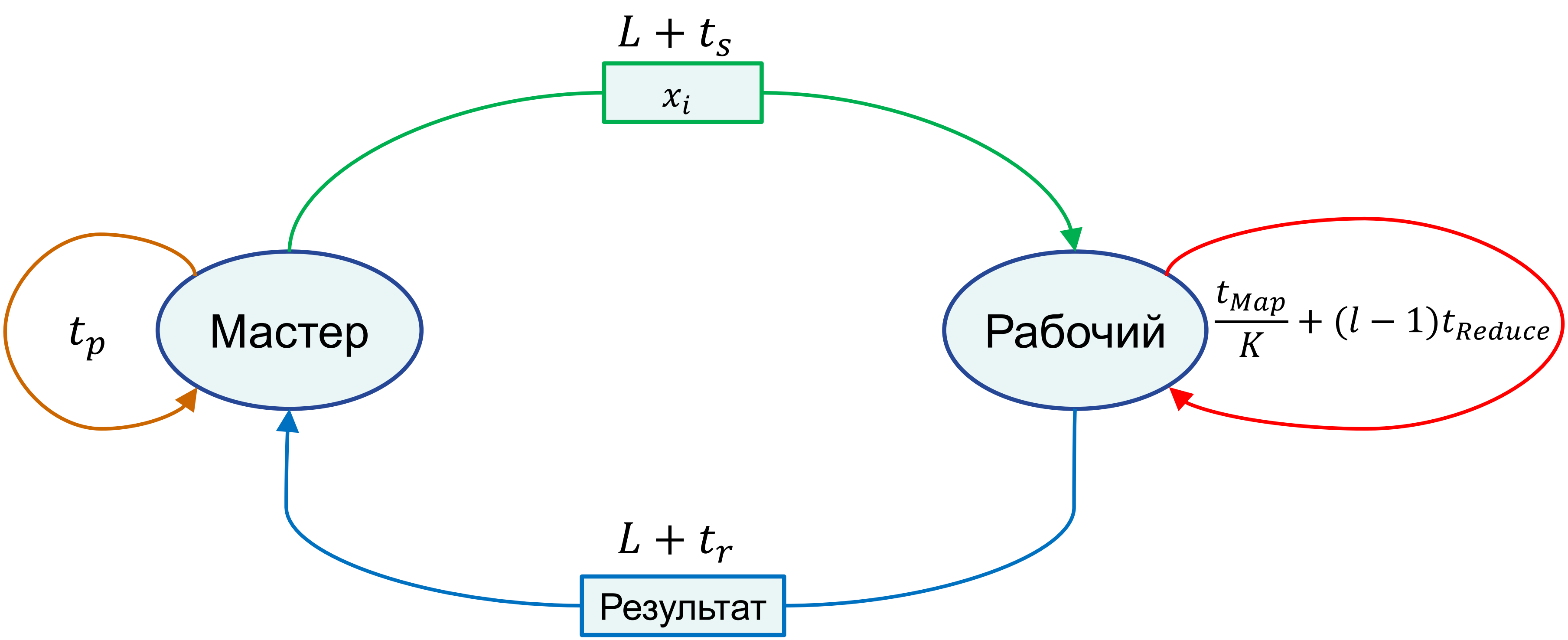
$$a(K) = \frac{T_1}{T_K}$$

Количество рабочих

Время решения задачи 1 мастером и K рабочими

Оценка времени T_1

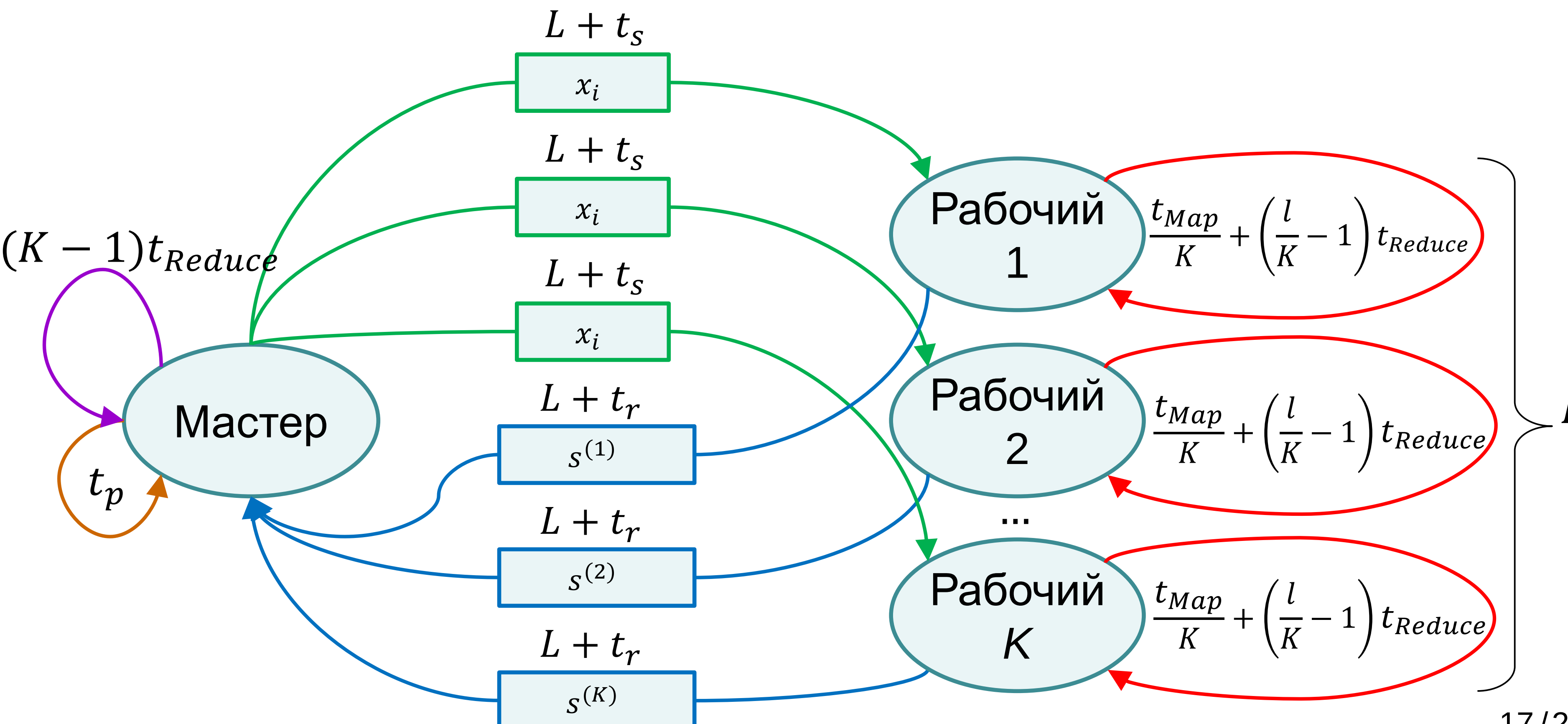
$$T_1 = L + t_s + t_{Map} + (l-1)t_{Reduce} + L + t_r + t_p$$



Оценка времени T_K

$T_K =$

$$K(L + t_s) + \frac{t_{Map}}{K} + \left(\frac{l}{K} - 1\right) t_{Reduce} + K(L + t_r) + (K - 1)t_{Reduce} + t_p$$



Ускорение для модели BSF

$$a_{BSF}(K) = \frac{2L + t_s + t_r + t_p + t_{Map} + lt_{Reduce}}{K(2L + t_s + t_r + t_{Reduce}) + \frac{(t_{Map} + lt_{Reduce})}{K} - t_{Reduce} + t_p}$$

Граница масштабируемости для модели BSF

Время выполнения операции *Map*

Время выполнения операции Reduce

Максимальное количество рабочих

Время выполнения ⊕

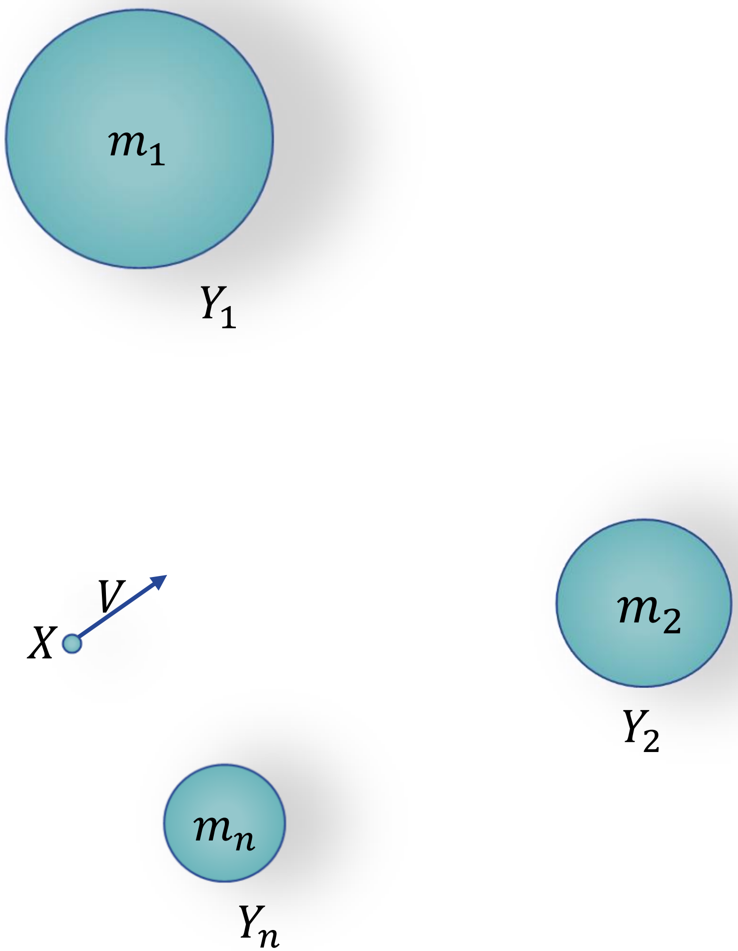
$$K_{Max} = \sqrt{\frac{t_{Map} + l t_{Reduce}}{2L + t_s + t_r + t_{Reduce}}}$$

Латентность

Время посылки сообщения одному рабочему

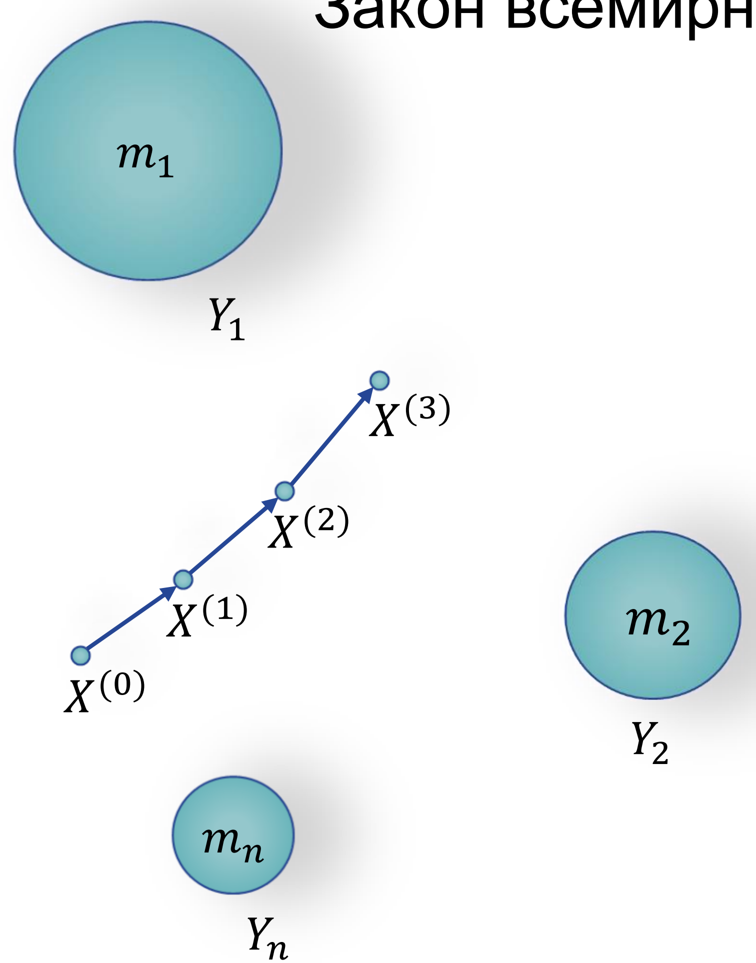
Время получения результата от одного рабочего

Гравитационная задача



Материальная точка X с малой массой m_x движется с начальной скоростью V среди n неподвижных тел $Y_1, Y_2, \dots, Y_n$ с большими массами $m_1, m_2, \dots, m_n$.

Необходимо рассчитать траекторию точки X с шагом Δt



Закон всемирного тяготения: $F_i = G \frac{m_i m_x}{\|Y_i - X\|^3}(Y_i - X)$

Второй закону Ньютона: $A_i = \frac{F_i}{m_x}$

$$\alpha_X(Y_i) = G \frac{m_i}{\|Y_i - X\|^3}(Y_i - X)$$

$$A_i = \alpha_X(Y_i)$$

$$A = \sum_{i=1}^{n} A_i$$

$$V^{(k+1)} = V^{(k)} + A \cdot \Delta t$$

$$X^{(k+1)} = X^{(k)} + V^{(k+1)} \cdot \Delta t$$

Алгоритм решения гравитационной задачи по шаблону BSF

(*T* – конечный момент времени, Δt - шаг по времени)

1. $t \coloneqq t_0; X^{(0)} \coloneqq 0; V^{(0)} \coloneqq 0;\ Input([(Y_1, m_1), \dots, (Y_n, m_n)])$
2. $[A_1, \dots, A_n] \coloneqq Map(\alpha_X, [(Y_1, m_1), \dots, (Y_n, m_n)])$
3. $A \coloneqq Reduce(+, [A_1, \dots, A_n])$
4. $V^{(t+\Delta t)} \coloneqq V^{(t)} + A \cdot \Delta t$

   $X^{(t+\Delta t)} \coloneqq X^{(t)} + V^{(t+\Delta t)} \Delta t$
5. Если $t \geq T$, перейти на шаг 7
6. $t \coloneqq t + \Delta t$; перейти на шаг 2
7. Стоп

Параллельный алгоритм решения гравитационной задачи по шаблону BSF

Шаг	Мастер	Рабочий j (j=1,…,K)
1.	$Input\left(T, t_0, \Delta t, X^{(t_0)}, V^{(t_0)}, [Y_1, \dots, Y_n]\right);\ t \coloneqq t_0$	$Input([Y_1, \dots, Y_n])$
2.	$SendToWorkers\left(X^{(t)}\right)$	$RecvFromMaster\left(X^{(t)}\right)$
		$B^{[j]} \coloneqq Map(\alpha_X, [Y_1, \dots, Y_n])$
3.		$A^{[j]} \coloneqq Reduce\left(\oplus, B^{[j]}\right)$
	for j = 1 **to** K **do**	
	$RecvFromWorkers([A^{(1)}, \dots, A^{(K)}])$	$SendToMaster(A^{[j]})$
	end for	
	$A \coloneqq Reduce\left(\oplus, [A^{(1)}, \dots, A^{(K)}]\right)$	
4.	$V^{(t+\Delta t)} \coloneqq V^{(t)} + A \cdot \Delta t;\ X^{(t+\Delta t)} \coloneqq X^{(t)} + V^{(t+\Delta t)} \Delta t$	
5.	**if** $t \geq T$ **go to** 7	
6.	$t \coloneqq t + \Delta t$**; go to** 2	**go to** 2
7.	$Output\left(X^{(t)}\right)$; **stop**	**stop**

Стоимостные параметры

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

$$t_s = 3\tau_{tr}$$

$$t_r = 3\tau_{tr}$$

$$t_{Map} = 20n\tau_{op}$$

$$t_{Reduce} = 3\tau_{op}$$

$$t_p = 14\tau_{op}$$

BSF-оценки для алгоритма Gravitation для многопроцессорных систем с распределенной памятью

$$a_{BSF}(K) = \frac{2L + 6\tau_{tr} + 14\tau_{op} + 20n\tau_{op} + 3l\tau_{op}}{K\left(2L + 96 + 14\tau_{op}\right) + \dfrac{20n\tau_{op} + 3l\tau_{op}}{K} + 11\tau_{op}}$$

$$K_{Max} = \sqrt{\frac{20n\tau_{op} + 3l\tau_{op}}{2L + 6\tau_{tr} + 14\tau_{op}}}$$

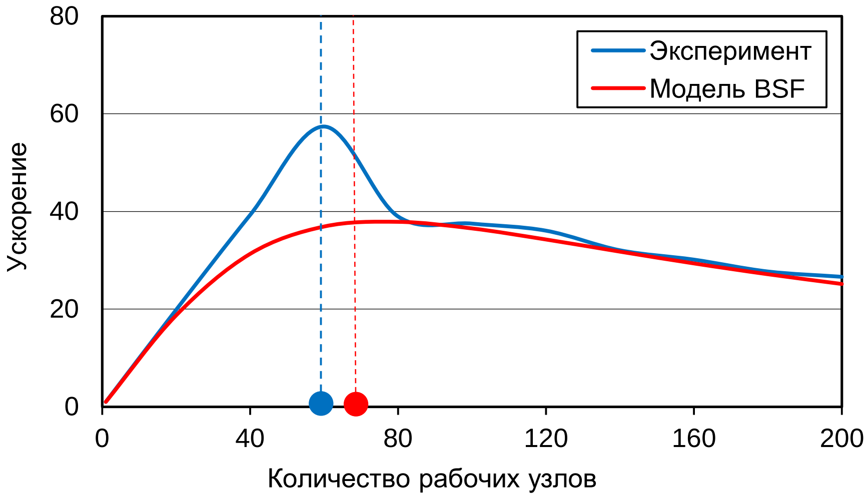
$$\boldsymbol{K_{max} = O\left(\sqrt{n}\right)}$$

Исходный код

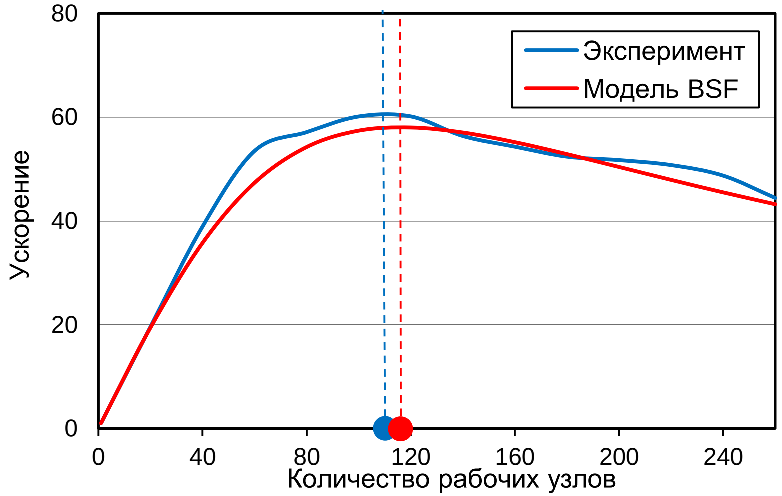
https://github.com/nadezhda-ezhova/Gravitation-MR

Ускорение гравитационного алгоритма: теория и практика

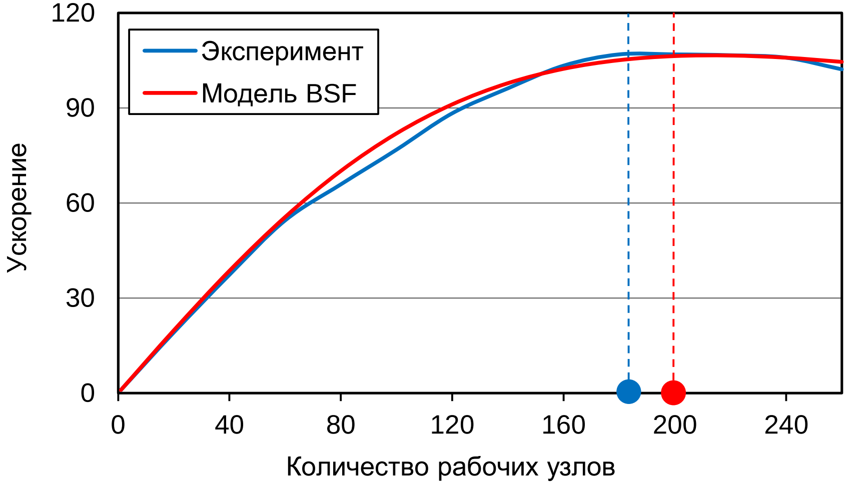
$n = 450$



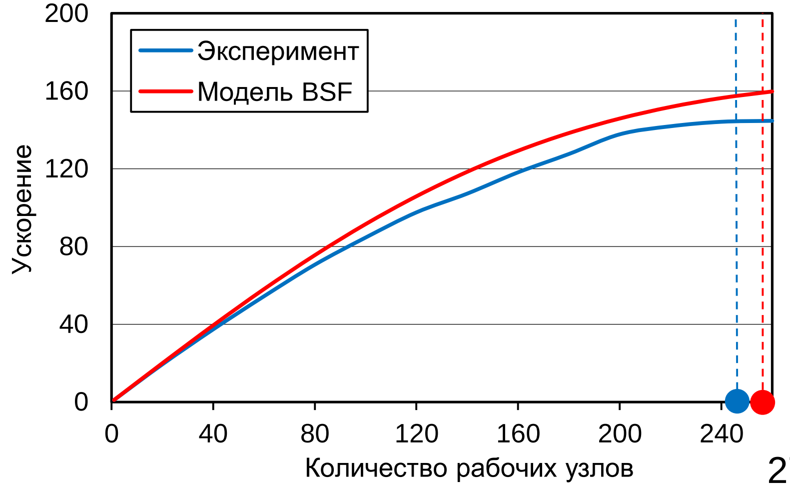
$n = 600$



$n = 900$



$n = 1200$



Другие верификации модели BSF

- *Ezhova N.A., Sokolinsky L.B.* Scalability Evaluation of Iterative Algorithms Used for Supercomputer Simulation of Physical processes // Proceedings - 2018 Global Smart Industry Conference, GloSIC 2018. Article number 8570107. IEEE, 2018. 10 p. DOI: 10.1109/GloSIC.2018.8570131.
- *Sokolinskaya I., Sokolinsky L.B.* Scalability Evaluation of NSLP Algorithm for Solving Non-Stationary Linear Programming Problems on Cluster Computing Systems // Supercomputing. RuSCDays 2017. Communications in Computer and Information Science. 2017. Vol. 793. P. 40-53. DOI: 10.1007/978-3-319-71255-0_4.
- *Sokolinskaya I.M., Sokolinsky L.B.* Scalability Evaluation of Cimmino Algorithm for Solving Linear Inequality Systems on Multiprocessors with Distributed Memory // Supercomputing Frontiers and Innovations. 2018. Vol. 5, No. 2. P. 11-22. DOI: 10.14529/jsfi180202.

Спасибо за внимание!

Параметры экспериментов

- $L = 1.5 \cdot 10^{-5}$ сек
(посылка «точка-точка» одного байта)
- $\tau_{op} = 2.9 \cdot 10^{-8}$ сек
(в цикле выполнялось 10^6 умножений; затраченное время делилось на 10^6)
- $\tau_{tr} = 1.9 \cdot 10^{-7}$ сек
(пересылалось сообщение длиной 10^6 байт; затраченное время делилось на 10^6)

Результаты экспериментов

Кол-во рабочих	**Время выполнения 10 итераций (сек.)**			
	n = 450	n = 600	n = 900	n = 1200
1	9.24	21.933	73.6	174
20	0.466	1.12	3.843	9.016
40	0.235	0.564	1.97	4.663
80	0.237	0.384	1.116	2.463
120	0.256	0.364	0.833	1.783
160	0.306	0.404	0.712	1.473
200	0.347	0.424	0.688	1.263
240		0.45	0.695	1.206
260		0.493	0.72	1.203

Требования к модели параллельных вычислений

1. Юзабилити
2. Переносимость
3. Предиктивность
   - Реальное (астрономическое) время выполнения программы
   - Абстрактное время выполнения программы, позволяющее предсказать, какая из двух программ будет выполняться быстрее
   - Масштабируемость программы, определяющая максимальное количество вычислительных модулей в многопроцессорной системе, после которого прирост ускорения становится нулевым или отрицательным

Фреймворк MapReduce

- Модель распределённых вычислений, представленная компанией Google
- Ориентирована на работу с Big Data
- Время пересылок много больше времени вычислений
- Структура данных «ключ-значение»

Список (list)

Операция конкатенации (объединения)списков является *ассоциативной*:

$$[a] +\!\!+ [b] +\!\!+ [c] = [a, b] +\!\!+ [c] = [a] +\!\!+ [b, c] = [a, b, c]$$

Мультимножество (bag)

Операция конкатенации (объединения) является ассоциативной и *коммутативной*:

$$\{a\} \cup \{b\} = \{b\} \cup \{a\}$$

Множество (set)

Операция конкатенации (объединения) множеств является ассоциативной, коммутативной и *идемпотентной*:

$$\{a\} \cup \{b\} = \{a\} \cup \{b\} \cup \{b\} = \{a, b\}$$

$$f(n) = O\big(g(n)\big)$$

$$\exists N : |f(n)| \leq C|g(n)|, \forall n > N$$

где $C \in \mathbb{R}_{>0}$ - некоторая константа