О ПОНЯТИИ УСКОРЕНИЯ И ЭФФЕКТИВНОСТИ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ[1]

М.А. Посыпкин, А.С. Хританков

ВВЕДЕНИЕ

Разработчиков программ для многопроцессорных и распределенных вычислительных систем всегда интересовали вопросы, насколько эффективно используются вычислительные ресурсы, в какой степени удалось ускорить программу по отношению к последовательному варианту, а также другие характеристики производительности. В частности для однородных параллельных программ были введены[1,2] такие понятия как *ускорение* – отношение времени работы алгоритма в последовательном и параллельном вариантах и *эффективность* – отношение ускорения к числу процессоров, участвующих в вычислениях. Впоследствии эти понятия были обобщены для параллельных систем, содержащих устройства различной производительности[2]. Были также установлены фундаментальные соотношения, связывающие долю последовательных вычислений в программе с ускорением и эффективностью – законы Амдала и Густавсона-Барсиса [1,2]. Вероятно в силу своей простоты и универсальности, данная теория широко применяется для анализа параллельных приложений и многопроцессорных систем.

В последнее десятилетие появились и получили развитие технологии распределенных вычислений или Грид[3], применяющиеся для проведения длительных расчетов, в которых участвуют разнородные вычислительные ресурсы различных организаций. Ключевой особенностью таких систем является неоднородная и динамически изменяющаяся структура ресурсов: вычислительные узлы могут подключаться к расчетам или выходить из них на протяжении всего времени расчетов. Традиционные модели, применяемые для параллельных систем, предполагают постоянный состав вычислительного пространства, и не применимы для описания распределенных систем. Поэтому, производительность распределенных приложений описывается, как правило, на качественном уровне.

В данной работе рассматривается распределенная вычислительная среда, в состав которой входят узлы различной производительности. При этом узлы участвуют в вычислениях в соответствии с некоторым расписанием – функцией, принимающей значение 1 в моменты времени, когда ресурс выделяется для вычислений, и 0 в остальные моменты. Такие системы далее именуются *системами с расписанием*. Для систем с расписанием обобщаются понятия ускорения, эффективности, приводятся соотношения, связывающие эти величины. Также для частного случая системы с расписанием – неоднородной системы выводится аналог закона Амдала. Приводится также описание реализации данной модели в среде распределенных вычислений BNB-Grid[4] и опыта ее применения для анализа эффективности решения задач глобальной оптимизации.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С РАСПИСАНИЕМ

Рассмотрим традиционную модель производительности параллельной системы, состоящую из n одинаковых функциональных устройств, объединенных вычислительной сетью []. Пусть дана вычислительной задача A, время решения которой на этой системе составляет T. Допустим, что задача может быть решена тем же алгоритмом на каждом устройстве системы последовательно. Алгоритм последовательного решения задачи будем называть *эталонным алгоритмом*. Пусть время решения задачи A с помощью эталонного алгоритма одним устройством составляет T_0. Тогда *ускорение* S определяется как отношение $S = T_0 / T$. Таким образом, ускорение показывает, во сколько раз можно уменьшить время решения задачи с помощью применения параллельной системы. Кроме понятия ускорения определяется понятие *эффективности* E параллельной системы как $E = S / n$.

В данной модели предполагается, что все узлы системы имеют одинаковые производительности и доступны на протяжении всего времени расчетов. Обобщим данную модель производительности на случай вычислительных систем, состоящих из узлов разной производительности, которые могут принимать участие в расчетах на части временного интервала решения задачи. Такие системы будем называть *системами с расписанием*. Для каждого устройства задано *расписание* – функция времени $h_i(t)$ такая, что $h_i(t) = 1$, если устройство в момент времени t выделено для решения, и $h_i(t) = 0$ в противном случае. *Доступностью* $\rho_i(t)$ устройства будем называть долю интервала времени $[0,t]$, в течение которого устройство было выделено для решения задачи: $\rho_i(T) = \frac{1}{T}\int_0^T h_i(t)d\tau$.

[1] Работа выполнена при финансовой поддержке программы № 15П фундаментальных исследований Президиума РАН «Разработка фундаментальных основ создания научной распределенной информационно – вычислительной среды на основе технологий GRID» и проектов РФФИ №06-07-89079-а, №08-07-00072-а

Обозначим через $\overline{T_i} > 0$ *эталонное время решения задачи* A i*-м устройством*, полученное с использованием эталонного алгоритма. Будем называть *эталонной производительностью* π_i устройства i при решении задачи A величину $\pi_i = \frac{L}{\overline{T_i}}$, где L – *трудоемкость* задачи, которая выражает наше априорное знание о сложности решения задачи, т.е. о вычислительных ресурсах, которые необходимо затратить на ее решение. Трудоемкость не имеет самостоятельного значения, и ее абсолютная величина может быть выбрана произвольно. В некоторых случаях трудоемкость задачи удобно принять равной единице.

Вычислительной системой с расписанием $\mathbf{R}$ будем называть совокупность

$\mathbf{R} = < \overline{\pi}, \overline{h}(t) >$, $\overline{\pi} = (\pi_1,..,\pi_n)$, $h(t) = \left(h_1(t),\ldots,h_n(t)\right)$.

Определим *эталонную производительность системы с расписанием* как сумму эталонных производительностей устройств, выделенных для решения задачи в данный момент времени *t*:

$$\pi(t) = \sum_{i=1}^{n} \pi_i h_i(t) = \overline{\pi}^T \cdot \overline{h}(t). \qquad (1)$$

При полной доступности устройств $h_i(t) \equiv 1$ в процессе решения задачи и при одинаковой эталонной производительности узлов $\pi_i = \pi_0$ эталонная производительность системы будет совпадать с $n \cdot \pi_0$, то есть с эталонной производительностью параллельной системы.

Эталонным временем решения $\overline{T}$ задачи системой $\mathbf{R}$ будем называть время ее решения при условии, что все узлы работают с эталонной производительностью. Эту величина определяется следующим соотношением:

$$\overline{T} = \min t : \int_{\tau=0}^{t} \pi(\tau) d\tau = L. \qquad (2)$$

Тогда естественно определить э*ффективность* E системы $\mathbf{R}$ как отношение времени решения эталонного и реального времен решения задачи: $E = \frac{\overline{T}}{T}$, где – T реальное время решения задачи A.

Ускорение для параллельной системы определяется как отношение времени решения задачи на одном устройстве ко времени решения задачи на всей системе. В системе с расписанием устройства могут быть различными, поэтому вводить понятие ускорения таким же образом некорректно. Определим более общее понятие относительного ускорения. *Ускорением* S *системы* $\mathbf{R}_1$ *относительно системы* $\mathbf{R}_2$ будем называть отношение времен решения задачи этими системами: $S(\mathbf{R}_1, \mathbf{R}_2) = T_2 / T_1$. По аналогии, определим *ускорение* S_i как отношение эталонного времени решения задачи на устройстве i ко времени решения задачи A на всей системе: $S_i = \overline{T}_i / T$. Под относительным ускорением системы с расписанием будем понимать вектор $\vec{S} = (S_1, S_2, .., S_n)$. Несложно показать [7] справедливость следующего утверждения, связывающего эффективность с ускорениями и функцией доступности.

Утверждение 1

Для распределенной системы с расписанием $\mathbf{R}$ справедливо соотношение:

$$E = \left(\sum_{i=1}^{n} \frac{\overline{\rho}_i}{S_i} \right)^{-1}, \quad \text{где } \overline{\rho}_i = \rho_i(\overline{T}). \qquad (3)$$

НЕОДНОРОДНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Рассмотрим важный частный случай систем с расписанием, в которых устройства доступны в течение всего времени решения, то есть $h_i(t) \equiv 1$ для любого устройства i. Системы такого рода назовем *неоднородными вычислительными системами*. От параллельных систем их отличает различная производительность узлов. Доступность каждого устройства неоднородной системы тождественно равна единице $\rho_i(t) \equiv 1$, поэтому выражение (3) для эффективности E примет вид: $E = \left(\sum_{i=1}^{n} \frac{1}{S_i} \right)^{-1}$. Если все устройства системы решают задачу A за равное вре-

мя, то неоднородная система будет эквивалентна параллельной и выражение для эффективности совпадает с выражением эффективности для параллельных систем: $E = \left(\sum_{i=1}^{n} \frac{1}{S_i} \right)^{-1} = \frac{S}{n}$.

Рассмотрим выражение для закона Амдала для параллельных систем [,]. Пусть β выражает долю последовательных вычислений. Закон Амдала задает максимально возможное ускорение для параллельных систем S_p^*: $S \le S_p^* = \frac{1}{\beta + (1-\beta)/n}$. Максимально возможная эффективность для параллельных систем при этом составляет $E_p^* = \frac{S_p^*}{n} = \frac{1}{\beta n + (1-\beta)}$. Данные соотношения очевидно не применимы для неоднородных систем.

Для неоднородных систем закон Амдала обобщается следующим образом.

Утверждение 2

Пусть устройства неоднородной вычислительной системы **R** упорядочены по убыванию эталонной производительности $\pi_1 \ge \pi_2 \ge .. \ge \pi_n$, тогда справедливы соотношения:

$$E \le \frac{1}{\beta \frac{\pi}{\pi_1} + (1-\beta)}, \qquad (4)$$

$$S_i \le \frac{1}{\beta \frac{\pi_i}{\pi_1} + (1-\beta)\frac{\pi_i}{\pi}}, \qquad (5)$$

где β - доля времени, затраченная системой на решение последовательной части задачи, а $\pi = \sum_{i=1}^{n} \pi_i$.

Доказательство

Время T, затраченное на решение задачи, не меньше чем $\beta L/\pi_1 + (1-\beta)L/\pi$. Тогда $E = \frac{\overline{T}}{T} \le \frac{L/\pi}{\beta L/\pi_1 + (1-\beta)L/\pi} = \frac{\beta \pi_1}{\beta \pi + (1-\beta)\pi_1} = \frac{\beta}{\beta \frac{\pi}{\pi_1} + (1-\beta)}$. Аналогично для ускорения имеем: $S_i = \frac{\overline{T_i}}{T} \le \frac{L/\pi_i}{\beta L/\pi_1 + (1-\beta)L/\pi} = \frac{1}{\beta \frac{\pi_i}{\pi_1} + (1-\beta)\frac{\pi_i}{\pi}}$. Утверждение доказано.

Соотношения (4) и (5) сводятся к классическому закону Амдала при $\pi_1 = \ldots = \pi_n$. Из этих соотношений следует, что неоднородная система может обладать большей эффективностью, чем однородная, при заданной доли последовательных вычислений. Действительно, если удается распределить последовательную часть вычислений на самый производительный процессор (первый), то простой остальных процессоров при этом меньше, чем в случае однородной системы.

РЕАЛИЗАЦИЯ В СИСТЕМЕ BNB-GRID

Построенная модель была реализована в трехуровневой иерархической распределенной системой, предназначенной для решения задач глобальной оптимизации BNB-Grid[3]. Исходная задача разбивается на множество подзадач, которые затем распределяются между вычислительными узлами системы. Компонент CS-Manager работает на корневом узле первого уровня, формирует вычислительное пространство (запускает приложения на узлах) и распределяет подзадачи между дочерними узлами второго уровня. На узлах второго уровня функционирует библиотека BNB-Solver[4], решающая задачи глобальной оптимизации на кластере или рабочей станции. На третьем уровне системы находятся процессоры, решающие подзадачи под управлением BNB-Solver.

Каждое приложение BNB-Solver, участвующее в решении подзадачи представлено в системе экземпляром агента CA-Manager (Computing Agent Manager). На одном физическом вычислительном ресурсе может работать

несколько агентов. Взаимодействие между CS-Manager и CA-Manager организовано с помощью промежуточного программного обеспечения ICE[5], основанного на тех же принципах, что и CORBA.

Состав вычислительного пространства может изменяться в процессе решения задачи по нескольким причинам. Во-первых, вычислительные узлы могут быть доступны на протяжении только части все времени вычислений из-за загруженности другими задачами или сбоев. Во-вторых, если подзадач на управляющем узле не осталось, то ресурсы, выполнившие предназначенную им работу, целесообразно освободить, т.е. удалить из вычислительного пространства. В-третьих, на кластерах общего доступа обычно устанавливается система пакетной обработки, лимитирующая время работы приложения на нем. Время ожидания приложения на кластере зависит от его текущей загруженности и запрошенной продолжительности счета. По окончании выделенного периода приложение принудительно завершается.

Алгоритм балансировки нагрузки в системе BNB-Grid работает по известной схеме «управляющий-рабочие». Перед началом решения исходная задача разбивается на подзадачи и из них формируется исходный пул подзадач. После начала решения задачи, по запросу CS-Manager передает приложению на узле число подзадач, равное количеству процессоров, используемых приложением. После того, как переданные подзадачи обработаны, приложение направляет новый запрос и получает новые подзадачи от CS-Manager. После того как все агенты завершили вычисления и пул подзадач пуст, CS-Manager прекращает решение задачи и останавливает все приложения.

В управляющем компоненте CS-Manager был реализован механизм сбора данных о вычислительном процессе. При этом устройства в модели были сопоставлены вычислительным агентам. Для предложенной модели необходимы следующие входные данные:

- общее время решения задачи оптимизации системой BNB-Grid,
- интервалы работы агентов за время решения задачи,
- эталонная производительность каждого агента.

Общее время решения задачи измеряется от момента начала решения до момента завершения решения последней подзадачи. Интервал выделения ресурса измеряется от момента старта вычислительного агента и системы BNB-Solver до остановки агента. Для каждого агента функция расписания задается выражением:

$$h_i(t) = \begin{cases} 1, & t \in [T_i^s, T_i^e) \\ 0, & t \in [0, T_i^s) \cup [T_i^e, T) \end{cases},$$

где T_i^s - время начала интервала, T_i^e - время завершения интервала, T - общее время решения задачи.

Для измерения эталонной производительности на каждом из вычислительных ресурсов были проведены эксперименты по решению задачи на одном процессоре. При этом эталонная производительность одного процессора j-го ресурса вычислялась по формуле $\pi_j^0 = L / T_j^0$, где T_j^0 - экспериментально измеренное время решения задачи на данном процессоре. Трудоемкость L играет роль коэффициента масштабирования и может быть выбрана произвольным образом. После этого, полагая процессоры одного ресурса одинаковыми, эталонная производительность для i-го агента, использующего n_i процессоров j-го ресурса, может быть рассчитана по формуле:

$$\pi_i = n_i \pi_j^0 = \frac{n_i L}{T_j^0}.$$

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для экспериментов была выбрана известная задача нахождения расположения атомов в молекуле с минимальной потенциальной энергией []. Постановка задачи следующая. Пусть заданы координаты $x = \{x^{(1)}, .., x^{(n)}\}$ расположения n атомов в молекуле. Тогда потенциальная энергия молекулы задается функцией

$$F(x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} v(\left\| x^{(i)} - x^{(j)} \right\|),$$

где $\left\| x^{(i)} - x^{(j)} \right\|$ - евклидово расстояние между атомами i и j, а функция $v(r)$ задает потенциал парных взаимодействий атомов. В нашем случае используется потенциал Морзе $v(r, \rho) = e^{\rho(1-r)}(e^{\rho(1-r)} - 2)$. Требуется найти расположение атомов в пространстве, при котором функция $F(x)$ принимает минимальное значение. Для решения задачи применялся предложенный в работе [8] алгоритм монотонного спуска из произвольной точки окрестности (Monotonic Basin-Hopping, MBH). В начале вычислений на управляющем узле формировался пул, содержащий несколько

случайных стартовых точек для МВН. Они играли роль подзадач. Исследовалась постановка $n = 20$, $\rho = 14$ и 24 исходные подзадачи.

Для проведения экспериментов использовались ресурсы, представленные в табл. 1. Результаты измерения эталонной производительности одного вычислительного ядра для каждого из вычислительных ресурсов приведены в табл. 2, трудоемкость полагалась равной 1000.

Таблица 1

Название	Модель процессора	Число процессоров	Расположение	Система пакетной обработки
МВС-100К	Intel Xeon Clovertown (4 ядра), 3 ГГц	940	МСЦ РАН, Москва	+
МВС-6000IM	Intel Itanium II, 2.2 ГГц	256	ВЦ РАН, Москва	+
Кластер ТГТУ	Intel Pentium IV, 3.2 ГГц	8	ТГТУ, Тамбов	-
Рабочая станция DCS	Intel Pentium IV, 3.2 ГГц	1	ИСА РАН, Москва	-

Таблица 2

Параметр	МВС-100К	МВС-6000IM	Кластер ТГТУ	Рабочая станция DCS
L	1000	1000	1000	1000
T_j^0, с	2225,9	4275,3	6993,0	7342,1
π_j^0	0,4493	0,2339	0,1430	0,1362

Для расчета эффективности работы системы с расписанием необходимо найти эталонное время $\overline{T}$ решения системой задачи A из интегрального уравнения (2). Подынтегральная функция производительности $\pi(t)$, определяемая формулой (1), является кусочно-линейной и, поэтому численное решение уравнения (2) не представляет труда. После завершения решения задачи системой, полученное значение использует для расчета эффективности по формуле: $E = \overline{T}/T$.

Рассмотрим в качестве примера расчет эффективности для системы, состоящей из узлов DCS, ТГТУ, МВС-100К и МВС-6000IM. Размещение приложений BNB-Solver и количество процессоров в каждом, приведено в табл. 3.

Таблица 3

№	Название	Количество агентов	Число процессоров каждого из приложений, n_i	Эталонная производительность, π_i
1	МВС-100К	2	3	1,330
2			3	1,330
3	МВС-6000IM	2	3	0,701
4			3	0,701
5	Кластер ТГТУ	1	4	0,572
6	Рабочая станция DCS	1	1	0,136

При решении задачи на приведенной выше конфигурации, было получено расписание, представленное на рис. 1. По оси ординат отложено суммарное число процессоров, выделенных на узле для решения задачи, а по оси абсцисс – время от начала решения задачи в секундах.

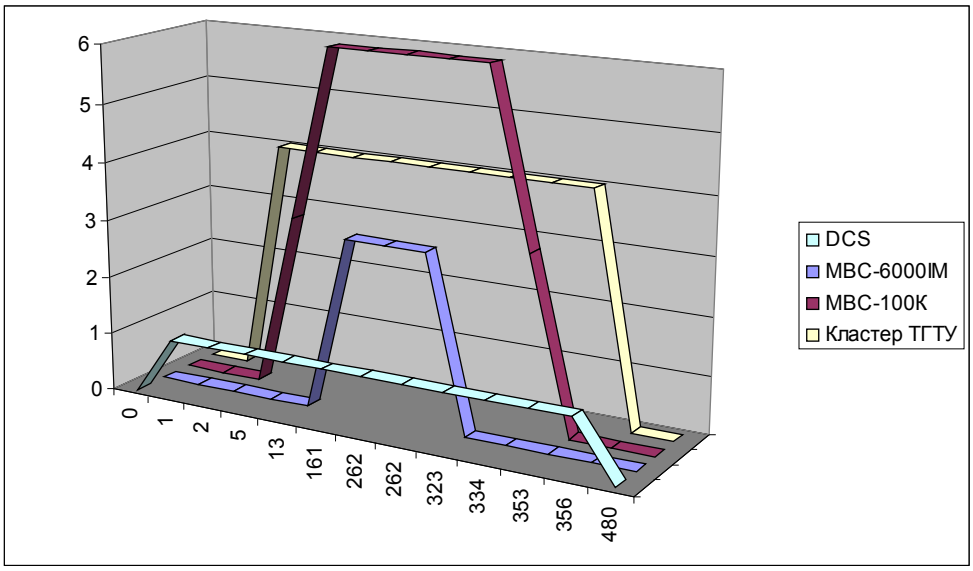

Рис. 1.

На графике видно, что рабочая станция DCS и кластер ТГТУ начали решение практически сразу, в то время как системы пакетной обработки, установленные на кластерах МВС-100К и МВС-6000IM, поставили задание в очередь. В результате, на кластере МВС-6000IM для решения задачи было выделено три процессора и запущен одно приложение, вместо двух, на МВС-100К были запущены оба приложения с некоторой задержкой от начала расчетов. Приложение, завершившие свою работу, останавливалось, если пул на управляющем узле был пуст.

Значения времени расчетов и эффективности для различных наборов вычислительных узлов приведены в табл. 4. Из результатов эксперимента видно, что с увеличением числа узлов в системе, эффективность E, вообще говоря, снижается, и уменьшается общее время расчета T. Значение эффективности указывает, насколько реальная система решает задачу медленнее гипотетической системы, использующей эталонный алгоритм и работающей по тому же расписанию, что и реальная система.

Таблица 4

№	Конфигурация	Количество агентов	Всего процессоров	E	T, с
1	DCS, ТГТУ, МВС-100К, МВС-6000IM	6	17	0,58	479
2	МВС-100К, МВС-6000IM	4	12	0,72	358
3	МВС-100К	2	6	0,71	636
4	МВС-6000IM	2	6	0,81	2218
5	ТГТУ	1	4	0,75	2319
6	DCS	1	1	1	7342

Эффективность решения задачи зависит не только от производительности узлов, но и от расписания. Расписание выделения узлов для системы 2 приведено на рис. 2. По оси ординат отложено число процессоров, выделенных на ресурсе для решения задачи, а по оси абсцисс – время от начала решения задачи в секундах. Вследствие того, что все узлы начали работу сразу после начала решения задачи, полное время решения T меньше полного времени решения для более производительной системы 1. Задержки с выделением узлов для расчетов также увеличили полное время решения для системы под номером 4.

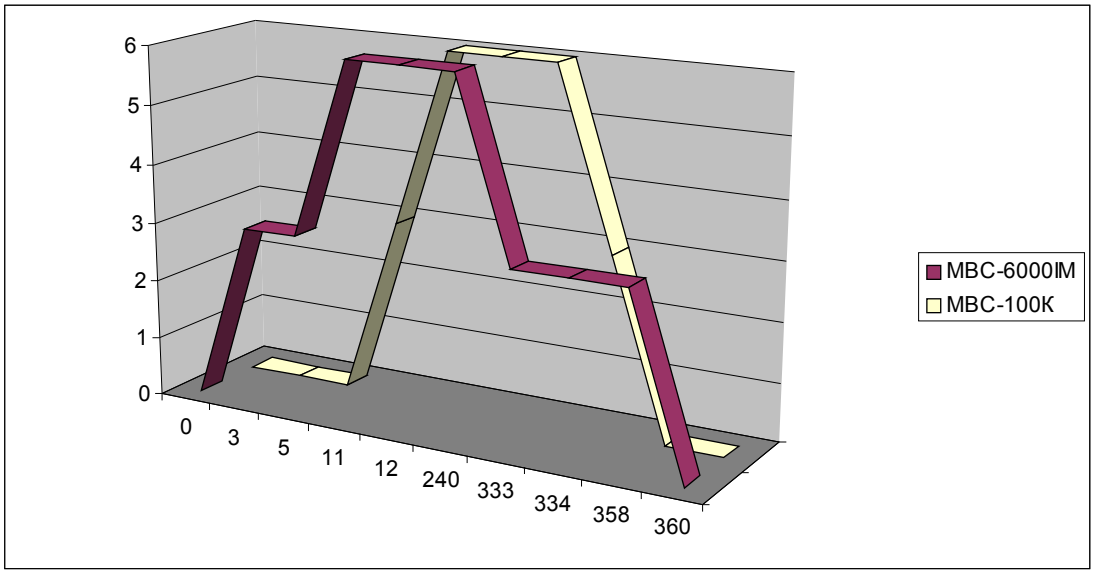

Рис. 2.

ЗАКЛЮЧЕНИЕ

В работе представлена модель производительности для неоднородных распределенных систем с расписанием. Введены основные характеристики производительности: эталонная производительность, эффективность и ускорение. В работе получено соотношение, связывающее эффективность и ускорение для распределенной системы с расписанием, аналогичное соотношению для параллельных систем. Также был обобщен закон Амдала на случай неоднородных систем.

Разработанная модель реализована в системе BNB-Grid. Проведенные эксперименты показали, что модель дает адекватное представление об эффективности использования ресурсов в процессе расчета. Также в результате анализа были выявлены потенциальные источники потерь эффективности и сформулированы рекомендации по формированию эффективного расписания запуска приложений на узлах.

ЛИТЕРАТУРА:

1. A. Grama, A. Gupta, G. Karypis, V. Kumar "Introduction to Parallel Computing, Second Edition"//USA: Addison Wesley, 2003.
2. В.В. Воеводин, Вл. В. Воеводин "Параллельные вычисления" // С.П.: БХВ-Петербург, 2002.
3. А.П.Афанасьев, В.В. Волошинов, С.В. Рогов, О.В. Сухорослов "Развитие концепции распределенных вычислительных сред" // Проблемы вычислений в распределенной среде: организация вычислений в глобальных сетях. Труды ИСА РАН. .: РОХОС, 2004, С.6-105.
4. А.П. Афанасьев, В.В. Волошинов, М.А. Посыпкин, И.Х. Сигал, Д.А. Хуторной "Программный комплекс для решения задач оптимизации методом ветвей и границ на распределенных вычислительных системах" // Труды ИСА РАН, 2006, Т. 25, С. 5-17.
5. М.А. Посыпкин "Архитектура и программная организация библиотеки для решения задач дискретной оптимизации методом ветвей и границ на многопроцессорных вычислительных комплексах" // Труды ИСА РАН, 2006, Т. 25, С. 18-25.
6. M. Henning "A New Approach to Object-Oriented Middleware" // IEEE Internet Computing, Jan 2004.
7. А. Хританков "Математическая модель характеристик производительности распределенных вычислительных систем" // Избранные труды 50й научной конференции МФТИ, 2007.
8. D. J. Wales, J. P. K. Doye "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms" // Journal of Physical Chemistry, №101, 1997, С. 5111-5116.
9. R. H. Leary "Global Optimization on Funneling Landscapes" // Journal of Global Optimization. №18-4, 2000, С. 367-383.