

Экспериментальное сравнение производительности и отказоустойчивости программных комплексов «Пирамида», X-COM и BOINC

А.В. Баранов, Е.А. Киселёв, Д.В. Черняев

Межведомственный суперкомпьютерный центр Российской академии наук – филиал Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук»

Статья посвящена экспериментальному сравнению программных комплексов «Пирамида», X-COM и BOINC по показателям производительности и отказоустойчивости. Статья содержит методики проведения и результаты экспериментов. Сравнение по производительности было проведено путем оценки накладных расходов на организацию распараллеливания по данным, для чего авторами были разработаны специальные тестовые примеры, моделирующие типовые задачи распараллеливания по данным. Сравнение по отказоустойчивости было проведено путем моделирования различных аварийных ситуаций, возникающих во время вычислений.

Ключевые слова: распараллеливание по данным, BOINC, X-COM, программный комплекс «Пирамида», накладные расходы на организацию параллельных вычислений, отказоустойчивость параллельных вычислений.

1. Введение

Среди различных прикладных задач, решаемых с помощью суперкомпьютеров, важный класс составляют задачи распараллеливания по данным, при котором одна и та же последовательность вычислений (прикладной алгоритм) выполняется над всеми элементами множества (пула) входных данных. При этом достаточно часто вычислительный алгоритм может быть реализован в виде одной последовательной программы (ОПП), для которой порция входных данных определяется значениями одного или нескольких параметров. Из таких порций вычислительной работы складывается пул входных данных, в итоге определяемый множеством всех возможных значений параметров ОПП во всех их комбинациях.

Современные кластерные вычислительные системы состоят из связанных вычислительных модулей (ВМ), среди которых обычно выделяется отдельная управляющая ЭВМ. При распараллеливании по данным на каждом вычислительном модуле кластера выполняются один или несколько экземпляров ОПП с различными значениями входных параметров. Организация параллельных вычислений заключается в осуществлении запусков экземпляров ОПП на всем множестве доступных ресурсов, при этом необходимо обеспечить максимальные производительность и отказоустойчивость.

При распараллеливании по данным скорость вычислений, как правило, прямо пропорциональна объёму используемых вычислительных ресурсов, прежде всего – процессорных ядер. В современных кластерных вычислительных системах не редкость наличие нескольких тысяч ВМ с суммарным числом ядер 10^5 - 10^6 и более. Длительное решение прикладной задачи в такой системе сопряжено с высокой вероятностью отказа во время счёта одного или нескольких ВМ кластера. Организация отказоустойчивых параллельных вычислений подразумевает как периодическое сохранение контрольных точек, так и исключение из решающего поля отказавших ВМ с автоматическим перераспределением вычислительной нагрузки на исправные модули. Вычисления при этом продолжают с некоторой деградацией производительности.

Для организации параллельных вычислений с распараллеливанием по данным в настоящее время могут быть применены несколько технологий, одну из которых реализует программный комплекс (ПК) «Пирамида» [1]. В работе [2] было осуществлено экспериментальное сравнение ПК «Пирамида» с технологиями MapReduce и MPI. Авторы продолжили начатые в [2] экспе-

рименты по сравнению ПК «Пирамида» с альтернативными аналогами, в качестве которых в настоящей работе рассмотрены программные комплексы X-COM [3] и VOINC [4].

На сегодняшний день экспериментального сравнения указанных ПК для решения задач распараллеливания по данным не проводилось. Настоящая работа призвана восполнить этот недостаток и предоставить потенциальным прикладным пользователям информацию для последующего выбора программного комплекса.

2. Программные комплексы организации параллельных вычислений с распараллеливанием по данным

Программный комплекс (ПК) «Пирамида» [1] предназначен для функционирования на вычислительной установке, имеющей иерархическую структуру. В составе вычислительной установки выделяют центральный сервер, серверы управления кластером и вычислительные модули в составе кластеров. ПК «Пирамида» изначально создавался с целью освобождения прикладного пользователя от решения задачи организации параллельных вычислений. Прикладной вычислительный алгоритм реализуется в виде ОПП, а за запуск множества экземпляров ОПП на вычислительных модулях и распределение вычислительной работы целиком отвечает ПК «Пирамида».

ПК «Пирамида» обеспечивает надёжность и отказоустойчивость вычислений. Выход из строя одного или нескольких вычислительных модулей, а также одного или нескольких кластеров, не останавливает расчёты, а только снижает их скорость. При повторном включении в работу (после устранения отказа) ранее отключившегося вычислительного модуля комплекс автоматически начинает выделять ему вычислительную работу. При этом ПК «Пирамида» с задаваемым периодом сохраняет контрольные точки, что позволяет восстановить вычисления при отказе всей вычислительной установки.

Рассмотрим аналоги-альтернативы ПК «Пирамида»: ПК X-COM и VOINC.

ПК X-COM [3] разработан специалистами НИВЦ МГУ им. М.В.Ломоносова и реализован на языке программирования Perl, что делает его одним из наименее ресурсоёмких средств распараллеливания по данным. Архитектурно в ПК X-COM можно выделить два основных компонента: сервер и вычислительные узлы. Сервер X-COM отвечает за разделение исходной задачи на блоки (задания), распределение заданий по узлам, координацию работ всех узлов, контроль целостности результата и сбор результата в единое целое. В качестве узла может выступить любая вычислительная единица (рабочая станция, VM кластера, виртуальная машина), имеющая возможность выполнить экземпляр прикладной программы. Узлы отвечают за расчёт блоков прикладной задачи (принятых от сервера заданий), запрос заданий от сервера, передачу результатов расчётов на сервер.

ПК VOINC [4] реализует одну из форм распределённых вычислений – добровольные вычисления, специфика которых заключается в использовании ресурсов персональных компьютеров, рабочих станций, вычислительных кластеров во время простоя, когда тот или иной вычислительный ресурс не задействуется под нужды владельца. Во время простоя ПК VOINC может использовать ресурс для выполнения прикладной ОПП, разработанной в рамках того или иного проекта VOINC. В качестве примера проекта VOINC можно привести SETI@HOME [5] – некоммерческий проект по поиску внеземного разума с помощью анализа представленных в цифровой форме радиосигналов с радиотелескопа Аресибо.

Подобно X-COM, ПК VOINC состоит из раздающего задания сервера и из множества выполняющих розданные задания клиентов. В состав сервера VOINC входят, как минимум, один веб-сервер для приёма и обработки клиентских запросов, сервер баз данных, хранящий состояния выданных заданий и соответствующие им результаты, планировщик, назначающий задания клиентам. В последние годы ПК VOINC получил широкое распространение, комплекс постоянно развивается и поддерживается многомиллионным сообществом добровольных пользователей и разработчиков.

Отметим следующие существенные отличия ПК X-COM и VOINC от ПК «Пирамида»:

1. X-COM и BOINC предназначены для организации параллельных вычислений в распределённой среде, хотя могут быть использованы и в рамках отдельной кластерной вычислительной установки. ПК «Пирамида» не функционирует в распределённой среде.

2. Для решения прикладной задачи с помощью ПК X-COM и BOINC необходима разработка серверной и клиентской управляющих программ, в то время как в ПК «Пирамида» достаточно подготовить паспорт задания.

3. Сохранение контрольных точек в ПК X-COM и BOINC возлагается на разработчика управляющих программ, ПК «Пирамида» сохраняет контрольные точки автоматически.

Настоящее исследование призвано определить, насколько эффективно применение ПК X-COM и BOINC для организации распараллеливания по данным в нераспределённой вычислительной среде по сравнению с ПК «Пирамида».

3. Методики экспериментального сравнения программных комплексов распараллеливания по данным

Сравнение программных комплексов «Пирамида», X-COM и BOINC проводилось авторами по показателям производительности и отказоустойчивости, причём для экспериментального определения значений каждого показателя использовалась отдельная методика. Анализ опубликованных работ по тематике исследований показал, что оценка производительности комплексов распараллеливания по данным осуществляется посредством измерения времени выполнения тестовых программ, моделирующих реальные вычислительные задачи. В работах [6-9] для оценки производительности программных комплексов используется программа, вычисляющая частоту встречаемых в текстовом файле слов. В работе [7] авторы используют программу вычисления частоты используемых RGB-цветов в изображении и программу поиска заданной строки (подстроки) в файле. В [9] для оценки производительности используется программа вычисления рейтинга посещаемости Интернет-ресурсов.

Оценка отказоустойчивости программных средств организации параллельных вычислений проводится посредством моделирования ситуации выхода из строя одного или нескольких вычислительных узлов на различных этапах вычислительного процесса. В [8] авторы проводят серию экспериментов, моделирующих потерю соединения с вычислительным узлом в момент загрузки на него обрабатываемой порции работы, в процессе вычислений и при сборе результатов вычислений.

В качестве экспериментального стенда для исследования ПК «Пирамида», X-COM и BOINC авторами был использован вычислительный кластер, решающее поле которого составляют 7 вычислительных модулей, каждый VM оснащён двумя 4-ядерными процессорами Intel Xeon L5408, т.е. общее число процессорных ядер решающего поля – 56. Кластер оснащён комплексной платформой виртуализации, что позволило установить исследуемые ПК в виде набора виртуальных машин под управлением гостевой ОС Linux Debian версии 8.

3.1 Методика сравнения по производительности

Для сравнения производительности программных комплексов авторами определялись накладные расходы на распараллеливание как доля времени, затрачиваемого каждым ПК на организацию вычислительного процесса. Определение накладных расходов осуществлялось следующим образом.

Пусть одна последовательная программа (ОПП) обрабатывает весь объём входных данных на одном процессоре (процессорном ядре) за некоторое время T . При использовании распараллеливания по данным на p процессорных ядрах время T обработки того же объёма входных данных в идеальном случае сократится в p раз. В действительности этого не произойдёт, поскольку имеют место быть накладные расходы на организацию распараллеливания: затраты времени на передачу данных между вычислительными модулями кластера, на обращения к сервисам (СУБД, веб-сервер, планировщик), задержки между получением данных и началом их обработки, между окончанием обработки и началом передачи результатов.

Введём понятие элементарной вычислительной работы, под которой будем понимать выполнение обработки неделимой (элементарной) порции входных данных. Элементарной работой могут являться обработка строки, сгенерированной на основе переданных параметров или считанной из файла, либо перебор вариантов значений в некотором минимальном диапазоне входных данных. Важно, что элементарная работа не может быть разделена на более мелкие части и, следовательно, не может быть распараллелена.

Пусть элементарная вычислительная работа выполняется за время τ , а весь объём входных данных составляют N элементарных порций. Следовательно, на одном процессорном ядре весь объём входных данных будет обработан за время $T_1 = N \cdot \tau$. При использовании p процессорных ядер идеальное время обработки T_p составит

$$T_p = \frac{T_1}{p} = \frac{N \cdot \tau}{p}$$

Пусть исследуемый ПК произвёл обработку входных данных на p процессорных ядрах за время $T_{\text{экс}}(p)$. Тогда доля накладных расходов μ , привносимых ПК, составит

$$\mu = 1 - \frac{T_p}{T_{\text{экс}}(p)} = 1 - \frac{N \cdot \tau}{p \cdot T_{\text{экс}}(p)} \quad (1)$$

Как видим, доля накладных расходов зависит от параметров N , τ и p . Варьируя значения одного из параметров при фиксированных значениях двух других, можно оценить динамику изменения накладных расходов на организацию распараллеливания по данным для каждого из исследуемых ПК.

3.2 Тестовые примеры

Для экспериментального определения накладных расходов должны быть подобраны тестовые примеры ОПП, удовлетворяющие следующим условиям.

1. Тестовая ОПП должна иметь возможность обработки от одной до произвольного числа элементарных порций входных данных, т.е. быть способной выполнить произвольное число N элементарных работ.

2. Для тестовой ОПП должна быть предусмотрена возможность задания времени τ выполнения элементарной работы.

3. Если тестовая ОПП выполняет N элементарных работ, за время τ каждую, то время выполнения ОПП на одном процессорном ядре должно составлять $N \cdot \tau$.

4. Обрабатываемая тестовой ОПП порция данных должна полностью определяться значениями её параметров.

5. Случай, когда обрабатываемая ОПП порция данных определяется значением (диапазоном значений) только одного параметра, встречается на практике довольно часто, поэтому целесообразно при сравнении ПК рассмотреть этот случай отдельно с использованием соответствующего тестового примера.

6. Отдельно следует рассмотреть случай, когда обрабатываемая ОПП порция данных определяется комбинацией значений (диапазонов значений) нескольких параметров, что существенно повышает трудоёмкость организации параллельных вычислений. Заметим, что именно этот случай является типичным при практическом применении ПК «Пирамида», поскольку комплекс обладает встроенным механизмом перебора всех возможных комбинаций значений нескольких параметров ОПП.

7. Отдельно следует рассмотреть случай, когда обрабатываемая ОПП порция данных определяется строкой (диапазоном строк) файла входных данных. Этот случай является типичным при применении ПК X-COM и BOINC.

Для проведения экспериментального сравнения авторами были разработаны три тестовых примера, удовлетворяющих поставленным условиям.

Тестовый пример *Opp_one* моделирует работу ОПП с одним входным параметром, значение которого задаёт диапазон входных данных в виде тройки чисел "a b s", где a – начало диапазона, b – конец диапазона, s – шаг. Например, тройка "10 20 3" задаёт последовательность перебираемых чисел 10, 13, 16, 19, каждое из которых определяет элементарную порцию дан-

ных. Для теста *Opp_one* есть возможность указания времени τ обработки одной элементарной порции данных. Например, если указать $\tau = 1$ с, то время обработки порции данных, заданной значением входного параметра "10 20 3", составит 4 с.

Тестовый пример *Opp_three* моделирует работу ОПП с тремя входными параметрами. Значения первых двух параметров задают диапазоны входных данных в виде троек чисел "a b s", где a – начало диапазона, b – конец диапазона, s – шаг. Значение третьего параметра задаёт список перебираемых строк. Например, если значение первого параметра – "10 15 3", значение второго – "1 2 1", значение третьего – "str1 str2", то порцию входных данных будут определять 8 возможных комбинаций значений трёх параметров:

"10 1 str1" "10 1 str2" "10 2 str1" "10 2 str2" "13 1 str1" "13 1 str2" "13 2 str1" "13 2 str2"

Так же как для теста *Opp_one*, для теста *Opp_three* предусмотрена возможность указания времени τ обработки одной комбинации значений входных параметров, т.е. времени обработки одной элементарной порции данных. Для нашего примера при $\tau = 1$ с время обработки 8 комбинаций составит 8 с.

Тестовый пример *Opp_file* моделирует обработку строк, считываемых из переданного ему в качестве параметра файла. Программа принимает на вход два параметра – имя файла со строками и время τ обработки одной строки из файла. Например, если в файле 20 строк, и время τ обработки одной строки равняется 2 с, то время работы тестового примера *Opp_file* на одном процессорном ядре составит 40 с.

3.3 Методика сравнения по отказоустойчивости

Учитывая состав экспериментального стенда, для проверки отказоустойчивости сравниваемых ПК применялась следующая методика. Все потенциальные отказы можно разделить на три группы: связанные с перезагрузкой вычислительного модуля, связанные с аварийным отключением питания одного или нескольких вычислительных модулей и связанные с разрывом соединения в коммуникационной сети. Для имитации отказов первой группы во время вычислений осуществлялась перезагрузка одного из вычислительных модулей (командой перезапуска виртуальной машины на панели инструментов гипервизора). Отказы второй группы имитировались выключением вычислительного модуля в процессе вычислений (командой выключения виртуальной машины на панели инструментов гипервизора). Помимо имитации отказа одного модуля, была произведена имитация отказа всех ВМ решающего поля и управляющей машины. После выключения ВМ они запускались повторно, и осуществлялась попытка продолжить прерванные вычисления. Для имитации отказов третьей группы осуществлялось отключение виртуального коммутатора.

4. Результаты проведённых экспериментов

4.1 Результаты экспериментов по определению накладных расходов

В ходе проведения экспериментов для каждого из сравниваемых ПК были осуществлены замеры времени выполнения трёх тестовых примеров – *Opp_one*, *Opp_file* и *Opp_three*, после чего по формуле (1) были вычислены накладные расходы на организацию распараллеливания по данным. Измерения времён выполнения для каждого тестового примера были проведены в три этапа:

- при переменном числе процессорных ядер p и постоянных объёме вычислений в N элементарных порций и времени τ обработки элементарной порции данных;
- при переменном τ и постоянных p и N ;
- при переменном N и постоянных p и τ .

На рис. 1 представлена динамика изменения накладных расходов на организацию распараллеливания ПК X-COM, BOINC и «Пирамида» для тестового примера *Opp_one* при переменном числе процессорных ядер, объёме данных в 10^5 элементарных порций, времени обработки элементарной порции 1 с. Отметим рост накладных расходов ПК BOINC с 6% до 36%. У ПК X-COM и «Пирамида» наблюдается медленный рост накладных расходов с 2% до 6-7%.

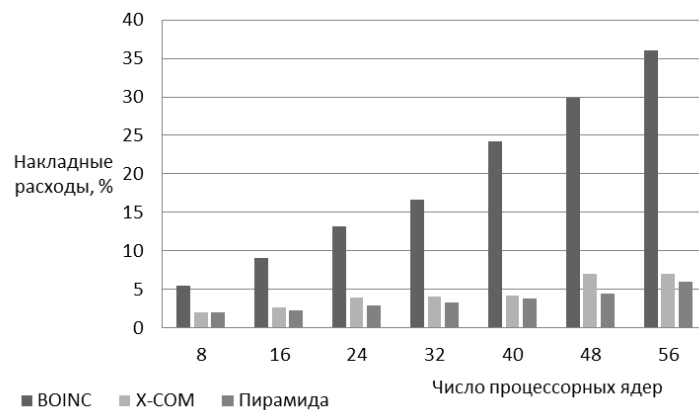


Рис. 1. Накладные расходы для тестового примера *Opp_one* при переменном числе процессорных ядер, $N=10^5$ элем. порций, $\tau = 1$ с

На рис. 2 представлена динамика изменения накладных расходов для тестового примера *Opp_one* при переменном значении времени обработки элементарной порции данных, объёме данных в 10^5 элементарных порций, числе процессорных ядер 56. При увеличении времени обработки элементарной порции у ПК BOINC отмечается снижение накладных расходов с 51% до 20%, после чего накладные расходы устанавливаются у отметки в 18-20%. У ПК X-COM и ПК «Пирамида» доля накладных расходов плавно снижается с 12% до 2%.

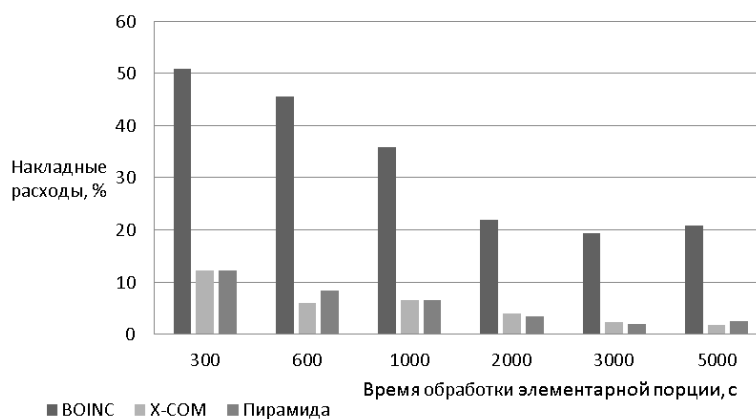


Рис. 2. Накладные расходы для тестового примера *Opp_one* при переменном значении времени обработки элементарной порции данных, $N=10^5$ элем. порций, $p=56$

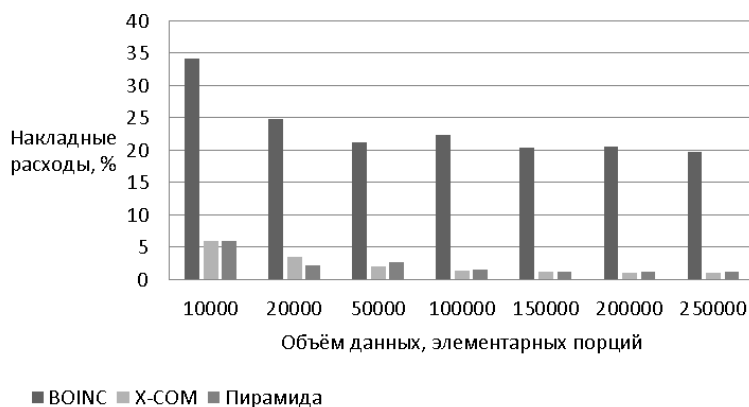


Рис. 3. Накладные расходы для тестового примера *Opp_one* при переменном объёме данных, $p = 56$, $\tau = 1$ с

На рис. 3 представлена динамика изменения накладных расходов для тестового примера *Opp_one* при переменном объёме данных, числе процессорных ядер 56, времени обработки элементарной порции 1 с. Наблюдается уменьшение накладных расходов ПК BOINC с 34% до 20%, после чего снижение останавливается у отметки в 20%. У ПК X-COM и ПК «Пирамида» накладные расходы плавно снижаются с 6% до 1%.

Представленные на рис. 2 и рис. 3 результаты свидетельствуют о повышении эффективности сравниваемых ПК как при укрупнении зерна параллелизма, так и при увеличении объёма входных данных.

На рис. 4 представлена динамика изменения накладных расходов для тестового примера *Opp_file* при переменном числе процессорных ядер. Объём данных во входном файле составил 10^5 строк, время обработки элементарной порции данных (одной строки из файла) – 1 с. Наблюдается рост накладных расходов у всех сравниваемых ПК. У ПК X-COM доля накладных расходов увеличивается с 2% до 6%, у ПК «Пирамида» – с 7% до 14%, у ПК BOINC – с 5% до 36%.

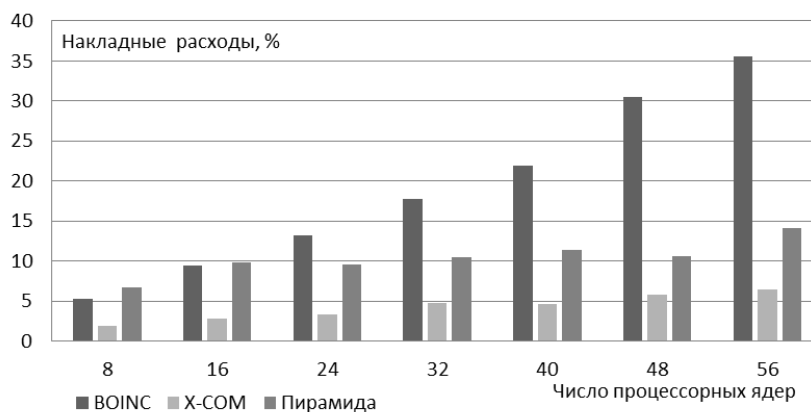


Рис. 4. Накладные расходы для тестового примера *Opp_file* при переменном числе процессорных ядер, $N = 10^5$ строк, $\tau = 1$ с

На рис. 5 представлена динамика изменения накладных расходов для тестового примера *Opp_file* при переменном числе строк во входном файле, времени обработки строки 1 с, числе процессорных ядер 56. Наблюдается уменьшение накладных расходов ПК X-COM с 7% до 2%, при объёме файла 50 тыс. строк и более накладные расходы составляют 2%. Накладные расходы ПК «Пирамида» уменьшаются с 15% до 11% и при объёме файла 20 тыс. строк и более находятся в пределах 10-11%. Накладные расходы ПК BOINC сравнительно быстро снижаются при увеличении объёма файла до 100 тыс. строк, далее наблюдается незначительное изменение накладных расходов в пределах 20-22%.

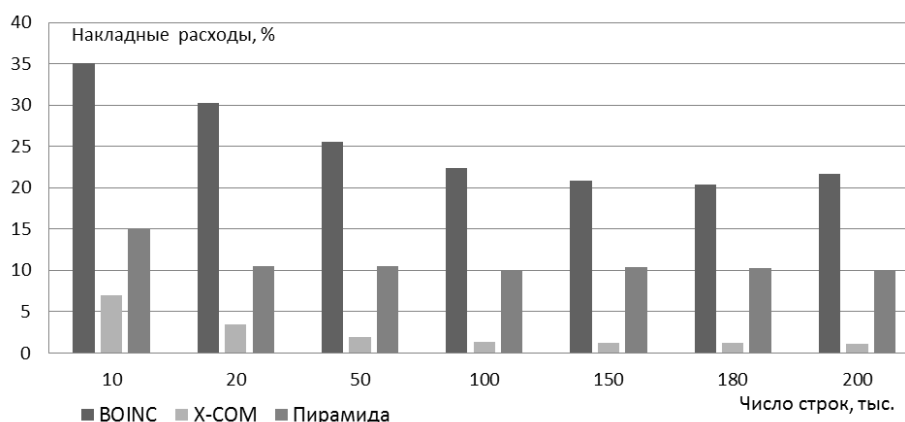


Рис. 5. Накладные расходы для тестового примера *Opp_file* при переменном числе строк в файле, числе процессорных ядер $p = 56$, $\tau = 1$ с

На рис. 6 представлена динамика изменения накладных расходов для тестового примера *Opp_file* при переменном значении времени обработки одной строки, объеме данных 10^5 строк в файле и числе процессорных ядер 56. Наблюдается быстрое снижение накладных расходов ПК X-COM с 15% до 5%, далее они медленно снижаются до значения 2%. Доля накладных расходов ПК «Пирамида» медленно снижается с 16% до 6%. Накладные расходы ПК BOINC быстро снижаются при увеличении времени обработки элементарной порции до 2000 мс, далее наблюдается незначительное изменение накладных расходов в пределах 20-22%.



Рис. 6. Накладные расходы для тестового примера *Opp_file* при переменном времени обработки одной строки, $N = 10^5$ строк, $p = 56$

На рис. 7 представлена динамика изменения накладных расходов для тестового примера *Opp_three* при переменном числе процессорных ядер, объеме данных в 10^5 элементарных порций, времени обработки элементарной порции 1 с. Наблюдается рост накладных расходов у ПК X-COM до значения 23%, затем наблюдается снижение до 13%. У ПК «Пирамида» наблюдается медленный рост величины накладных расходов с 2% до 5%. Накладные расходы ПК BOINC быстро растут с 11% до 33%.

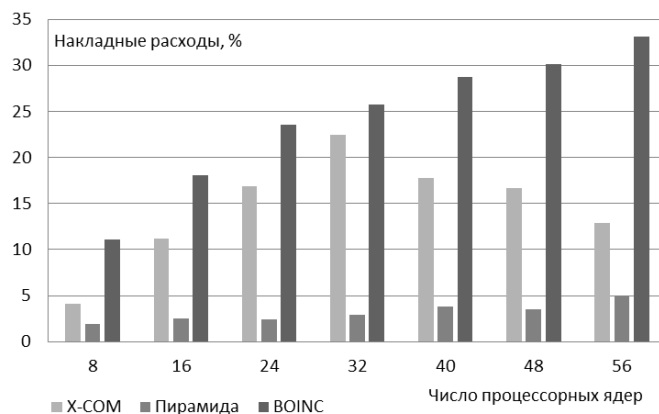


Рис. 7. Накладные расходы для тестового примера *Opp_three* при переменном числе процессорных ядер, $N = 10^5$ элем. порций, $\tau = 1$ с

На рис. 8 представлена динамика изменения накладных расходов для тестового примера *Opp_three* при переменном значении времени обработки элементарной порции данных, объеме данных в 10^5 элементарных порций, числе процессорных ядер 56. Наблюдается снижение доли накладных расходов у ПК X-COM с 14% до 11%, у ПК «Пирамида» – с 9% до 1%, у ПК BOINC – с 54% до 24%.

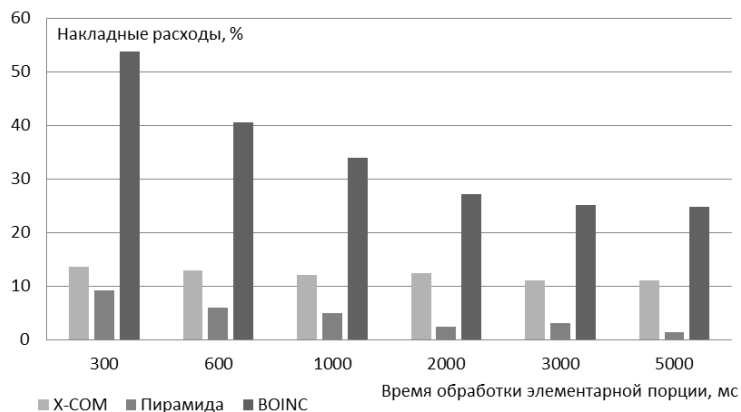


Рис. 8. Накладные расходы для тестового примера *Opp_three* при переменном времени обработки элементарной порции данных, $p = 56$ ядер, $N = 10^5$ элем. порций

На рис. 9 представлена динамика изменения накладных расходов для тестового примера *Opp_three* при переменном объеме данных, числе процессорных ядер 56, времени обработки элементарной порции 1 с. Доля накладных расходов у ПК X-COM находится в диапазоне с 11% до 12%. Накладные расходы ПК «Пирамида» снижаются с 4% до 2%, ПК VOINC – с 34% до 20%.

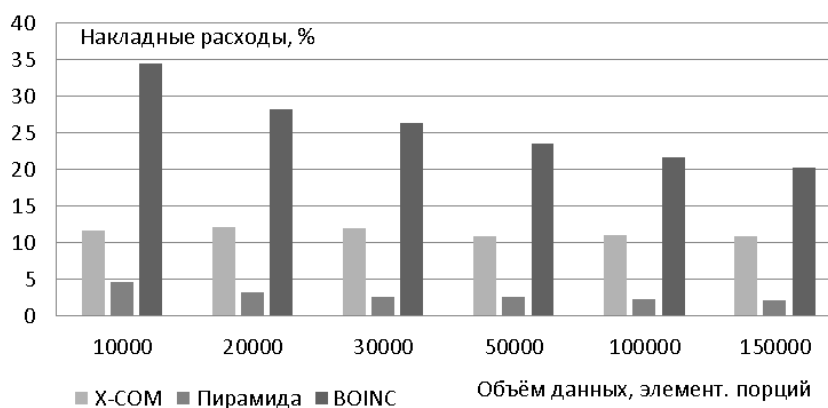


Рис. 9. Накладные расходы для тестового примера *Opp_three* при переменном времени объеме данных, $p = 56$ ядер, $N = 10^5$ элем. порций

4.2 Результаты экспериментов по сравнению отказоустойчивости

В ходе экспериментальных тестов на отказоустойчивость все сравниваемые ПК продемонстрировали одинаковую реакцию на все смоделированные согласно методике аварийные ситуации:

- после перезагрузки вычислительного модуля, ему автоматически начинает выделяться вычислительная работа;
- при выключении ВМ вычислительная работа перераспределяется между рабочими модулями;
- при отказе управляющей машины вычисления прекращаются;
- при отключении сетевого интерфейса у управляющей машины она бесконечно ожидает соединения с вычислительными модулями;
- при отключении сетевого интерфейса у вычислительного модуля вычислительная работа перераспределяется между рабочими модулями;
- при перезапуске всех ВМ, после попытки продолжить прерванные вычисления, вычисления продолжают с момента сохранения последней контрольной точки.

4. Выводы

1. ПК BOINC показал во всех экспериментахкратно большие накладные расходы по сравнению с ПК «Пирамида» и X-COM. Связано это, прежде всего, со сложной структурой комплекса, которую составляют множество компонентов, включая СУБД. ПК BOINC создавался для надёжного функционирования в распределённой среде, и его применение в рамках отдельных кластерных вычислительных систем нецелесообразно.

2. Накладные расходы всех ПК во всех тестовых примерах растут при увеличении числа процессорных ядер и, как правило, снижаются при увеличении объёма входных данных и времени обработки элементарной порции данных.

3. ПК X-COM показал кратно лучшие результаты при обработке строк, считываемых из входного файла, причём наивысшей эффективности (накладные расходы около 2%) комплекс достигает на больших файлах и при крупном зерне параллелизма.

4. ПК «Пирамида» показал кратно лучшие результаты при обработке нескольких параметров ОПП, что логично, поскольку этот ПК – единственный из исследуемых комплексов, обладающий встроенным механизмом перебора комбинаций значений нескольких параметров.

5. Все исследуемые ПК продемонстрировали одинаково высокие показатели отказоустойчивости.

Литература

1. Баранов А.В., Киселёв А.В., Киселёв Е.А., Корнеев В.В, Семёнов Д.В. Программный комплекс «Пирамида» организации параллельных вычислений с распараллеливанием по данным. Научный сервис в сети интернет: суперкомпьютерные центры и задачи: Труды международной суперкомпьютерной конференции (Новороссийск, 20-25 сентября 2010 г.). М: Изд-во МГУ, 2010. С. 299-302. URL: <http://agora.guru.ru/abrau2010/pdf/299.pdf> (дата обращения: 01.06.2016).
2. Алексеев А.В., Баранов А.В., Киселёв А.В., Киселёв Е.А. Экспериментальное сравнение технологий распараллеливания по данным «Пирамида», MapReduce и MPI. Суперкомпьютерные технологии (СКТ-2014): Материалы 3-й Всероссийской научно-технической конференции (Дивноморское, Геленджик, 29 сентября – 04 октября 2014 г.). Ростов-на-Дону: Издательство ЮФУ, 2014. Т. 1. С. 77-80.
3. Филамофитский М.П. Система поддержки метакомпьютерных расчетов X-COM: архитектура и технология работы // Вычислительные методы и программирование. 2004. Т. 5. С. 123–137. URL: http://www.num-meth.srcc.msu.ru/zhurnal/tom_2004/pdf/v5r201.pdf (дата обращения: 01.06.2016).
4. BOINC. Open-source software for volunteer computing. URL: <http://boinc.berkeley.edu> (дата обращения: 01.06.2016).
5. SETI@HOME. URL:<http://setiathome.ssl.berkeley.edu> (дата обращения: 01.06.2016).
6. Volunteer Cloud Computing: MapReduce over the Internet. URL: <http://hpcolsi.dei.uc.pt/hpcolsi/publications/mapreduce-pcgrid2011.pdf> (дата обращения: 29.07.2016).
7. MapReduce Framework Performance Comparison. URL: http://www.cs.ru.nl/bachelorscripties/2013/Thomas_Naegele___4031253___MapReduce_Framework_Performance_Comparison.pdf (дата обращения: 29.07.2016).
8. Towards MapReduce for Desktop Grid Computing. URL: <http://graal.ens-lyon.fr/~gfedak/thesis/xtremmapreduce.3pgcic10.pdf> (дата обращения: 29.07.2016).
9. A Comparison of Approaches to Large-Scale Data Analysis. <http://database.cs.brown.edu/sigmod09/benchmarks-sigmod09.pdf> (дата обращения: 29.07.2016).

The experimental comparison of performance and fault tolerance of software packages «Pyramid», X-COM and BOINC

A.V. Baranov, E.A. Kiselev, D.V. Chernyaev

Joint Supercomputer Center of the Russian Academy of Sciences - Branch of Federal State Institution «Scientific Research Institute for System Analysis of the Russian Academy of Sciences»

The paper is devoted to the experimental comparison of performance and fault tolerance of software packages "Pyramid", X-COM and BOINC. This paper contains technique for carrying out experiments and the results of this experiments. The performance comparison was done by assessing the overhead costs to organize parallelization by data. In this case special tests simulating typical tasks of parallelization by data were designed by the authors. The comparison of fault tolerance was done by simulating various emergency situations that arise during calculations.

Keywords: parallelization by data, BOINC, X-COM, software package «Pyramid», overhead costs to organize parallelization, fault tolerance of parallel calculations.

References

1. Baranov A.V., Kiselev A.V., Kiselev E.A., Korneev V.V., Semenov D.V. Programmnyy kompleks «Piramida» organizatsii parallel'nykh vychisleniy s rasparallelivaniem po dannym [The software package «Pyramid» for parallel calculations arrangement with parallelization by data]. Nauchnyy servis v seti internet: superkomp'yuternye tsentry i zadachi: Trudy mezhdunarodnoy superkomp'yuternoy konferentsii (Novorossiysk, 20-25 sentyabrya 2010 g.) [Scientific Services & Internet: Supercomputing Centers and Applications: Proceedings of the International Supercomputing Conference (Novorossiysk, Russia, September, 20– 25, 2010)]. Moscow, Publishing of Lomonosov Moscow State University, 2010. P. 299–302. URL: <http://agora.guru.ru/abrau2010/pdf/299.pdf> (accessed: 01.06.2016).
2. Alekseev A.V., Baranov A.V., Kiselev A.V., Kiselev E.A. Eksperimental'noe sravnenie tekhnologiy rasparallelivaniya po dannym «Piramida», MapReduce i MPI [The experimental comparison of the technologies of parallelization by data "Pyramid", MapReduce and MPI]. Superkomp'yuternye tekhnologii (SKT-2014): Materialy 3-y Vserossiyskoy nauchno-tekhnicheskoy konferentsii (Divnomorskoe, Gelendzhik, 29 sentyabrya – 04 oktyabrya 2014 g.) [Supercomputing Technologies (SCT'2014): Proceedings of the 3-d All-Russian Scientific and Technical Conference (Divnomorskoe, Gelendzhik, Russia, September, 29 – October, 04, 2014)]. Postov-on-Don, Publishing of the Southern Federal University, 2014. V. 1. P. 77–80.
3. Filamofitsky M.P. Sistema podderzhki metakomp'yuternykh raschetov X-COM: arkhitektura i tekhnologiya raboty [The system X-Com for metacomputing support: architecture and technology]. Vychislitel'nye Metody i Programirovanie [Numerical Methods and Programming]. Moscow, Publishing of the Research Computing Center of Lomonosov Moscow State University, 2004. V. 5. P. 123–137. URL: http://www.num-meth.srcc.msu.ru/zhurnal/tom_2004/pdf/v5r201.pdf (accessed: 01.06.2016).
4. BOINC. Open-source software for volunteer computing. URL: <http://boinc.berkeley.edu> (accessed: 01.06.2016).
5. SETI@HOME. URL:<http://setiathome.ssl.berkeley.edu> (accessed: 01.06.2016).
6. Volunteer Cloud Computing: MapReduce over the Internet. URL: <http://hpcolsi.dei.uc.pt/hpcolsi/publications/mapreduce-pcgrid2011.pdf> (accessed: 29.07.2016).

7. MapReduce Framework Performance Comparison. URL:
http://www.cs.ru.nl/bachelorscripties/2013/Thomas_Naegele___4031253___MapReduce_Framework_Performance_Comparison.pdf (accessed: 29.07.2016).
8. Towards MapReduce for Desktop Grid Computing. URL: <http://graal.ens-lyon.fr/~gfedak/thesis/xtremmapreduce.3pgcic10.pdf> (accessed: 29.07.2016).
9. A Comparison of Approaches to Large-Scale Data Analysis.
<http://database.cs.brown.edu/sigmod09/benchmarks-sigmod09.pdf> (accessed: 29.07.2016).