

Техническая документация «DistComp»

Автор: Агеев Данил Евгеньевич

Содержание:

1. Предисловие, дополнительная информация.
2. Распределение вычислений ($3n+1$)
3. Распределение вычислений ($an+b$)
4. Поток, работающий на стороне сервера
5. Поток, работающий на стороне клиента
6. Отправка отчетов по e-mail.
7. Запуск и настройка программы

Предисловие, доп. информация.

DistComp – это программа, позволяющая распределять вычисления по алгоритму гипотезы Коллатца между несколькими компьютерами.

Далее в тексте будут использованы наименования сторон программы – *сервер* и *клиент*. Сервер распределяет вычисления между клиентами, а клиенты выполняют эти вычисления. При первом запуске Клиент генерирует UUID (идентификатор) и сохраняет его в файл `uuid.txt`, идентификатор клиента служит для определения и запоминания конкретного клиента сервером. Если клиент прекратил работу, то в любой момент сможет продолжить ее, подключившись к серверу с таким же идентификатором, который был в прошлый раз.

Ведется логирование как в файл, находящийся в папке «log», так и в стандартный вывод (STDOUT). Для каждой сессии создается новый файл, названный по времени его создания в формате «год-месяц-день-час-минута-секунда.txt».

Распределение вычислений ($3n+1$).

В этом параграфе описано распределение вычислений для DistComp версии 2.1 и ниже.

Распределение вычислений происходит по системе выделения интервалов для каждого из клиентов. Всего в программе предусмотрено два типа интервала – *выделенный* и *свободный*. Выделенный интервал – это интервал $[a...b]$, содержащий целые числа, над которым работает клиент. В нем задаются два параметра – «от числа N» (включительно) и до «до числа N» (исключительно). Так например, интервал, обозначенный $[5...8]$ будет покрывать числа 5, 6 и 7. Свободный же интервал содержит все целые числа, которые не были выделены, и обозначается он как $[a...inf]$.

Давайте рассмотрим пошаговую обработку интервалов сервером. Зеленым цветом обозначим свободный интервал.



Допустим точкой отсчета¹ у сервера будет 3, а буфер клиента¹ – 4 (сколько чисел выделяется за раз) Значит свободный интервал здесь $[3...inf]$ Сейчас сервер инициализирован и готов к распределению работы.



Через некоторое время к серверу подключились два клиента (в хронологическом порядке: красный, синий). Сервер выделяет им по интервалу. Красному – $[3...7]$, а синему – $[7...11]$ (т.к. буфер клиента равен 4, каждый получает по 4 числа) А свободный интервал стал $[11...inf]$



Клиенты начали работать, и спустя некоторое время красный клиент закончил свою работу, в то время как синий сделал всего лишь часть от своей. Красный клиент сообщает серверу о том, что он закончил работу на своем выделенном интервале, а сервер выделяет ему еще – $[11 \dots 15]$, при этом свободный интервал передвигается до $[15 \dots \text{inf}]$



При потере соединения с клиентом, сервер запоминает его интервал и место, на котором он остановил вычисление, и сохраняет его до того момента, когда этот же клиент подключится.

Примечание: Интервал, однажды выделенный одному клиенту, никогда не станет частью свободного.

¹ Значение точки отсчета и буфера клиента задаются при инициализации сервера и являются положительными целыми числами

Распределение вычислений ($an+b$).

В этом параграфе описано распределение вычислений для DistComp версии 2.2.

Распределение вычислений с коэффициентами, отличными от $a=3$; $b=1$, производится по похожей системе. Единственным исключением является то, что каждому клиенту выделяется один и тот же интервал чисел, заданный при инициализации программы, только с разными коэффициентами.

В файле `COEF.txt` должны быть указаны коэффициенты, для которых необходимо проводить расчеты. Указываются они парами в каждой строке. Одна пара выглядит как "a b" (первый и второй коэффициенты отделены пробелом). Пример:

```
3 -1
5 -1
7 -1
5 1
5 3
```

Результат расчетов по каждому числу будет записан в файл `COMPLOG_an±b_IDклиента.txt` (пример: `COMPLOG_5n+1_a5cf9d17-19f7-3bdc-95e7-b5a501bcef14.txt`).

- Если число закончилось циклом, то в файле появится запись типа «CY A B», где **A** – число, для которого пишется результат, а **B** – первое число, которое закончилось таким же циклом. Если это новый цикл, который программа еще не встречала, то **A** и **B** будут равны.
- Если лимит на кол-во рекурсивных шагов для одного числа (см. «Запуск и настройка программы») превышен, то делается создается запись типа «MSR A». В данном случае, число либо никогда не приведет к 1, либо потребует больше рекурсивных действий.

Потоки, работающие на стороне сервера.

1. «SocketAcceptor» – принимает, клиентов, пытающихся подключиться к серверу. `ServerSocket.accept()` блокирует поток, поэтому для принятия клиентов необходимо выделить отдельный поток.
2. «Console» – читает и обрабатывает запросы из потока стандартного ввода (`STDIN`). Подробнее о доступных командах в разделе «Запуск настройка программы»
3. «ClientHandler» – для каждого подключенного клиента запущен свой `ClientHandler` поток (различаются они по `SocketAddress` и `UUID`, приписанному в конец названия потока). Здесь происходит чтение и обработка информации из потока ввода сокета клиента (`java.lang.Socket`)
4. «Heartbeat» – запускается JVM каждые 20 секунд и проверяет, если от клиентов было получено сообщение за последние 20 секунд, производит отключение неактивных клиентов.

Потоки, работающие на стороне клиента.

1. «ServerCom» – читает и обрабатывает запросы из потока ввода сокета сервера (`java.lang.Socket`)
2. «Solver» – выполняет вычисления на заданном интервал
3. «Heartbeat» – вызывается JVM раз в 5 секунд и отправляет пустое сообщение на сервер, чтобы поддерживать соединение.

Отправка отчетов по электронной почте.

Сервер будет отправлять отчеты на указанный в конфигурации адрес электронной почты. Периодичность задается при инициализации (см. «Запуск и настройка программы»).

Настройки отчетов пишутся в файл `email.txt`. Этот файл необходимо создать вручную и написать три строки:

1. Адрес e-mail, с которого нужно отправлять отчеты (должен быть аккаунт Gmail)
2. Пароль от аккаунта Google для этой почты.
3. Адрес e-mail, на который нужно отправлять отчеты.

Например:

```
distcompemail@gmail.com  
distcompemailpassword123  
myemail@gmail.com
```

Отчет содержит список всех работающих клиентов и их текущий интервал. Также к письму прилагается zip-архив со всеми COMPLOG файлами. Если размер архива выходит за лимит (25МБ), то письмо отправляется без вложения.

Запуск и настройка программы.

Серверная сторона запускается с единственным обязательным аргументом – портом, на котором будет работать сервер.

Пример команды запуска программы как сервер в Windows:

```
java -jar distcomp.jar server 12345
```

У стороны клиента же два аргумента – порт и адрес сервера.

Пример:

```
java -jar distcomp.jar client 12345 127.0.0.1
```

Дополнительно, добавив `-enableDebugLogging` в конец аргументов запуска, можно включить логирование на уровне отладки. Это значительно замедлит вычисление, т.к. каждое обработанное число нужно будет записать в файл.

(v2.1 и ранее) Для инициализации вычисления необходимо написать следующую команду в стандартный ввод сервера:

```
init [1] [2] [3] [4] [5] [6]
```

Разберем аргументы:

1. Число, с которого начнется вычисление. Если значение этого аргумента равно a , то свободный интервал будет установлен на $[a...inf]$
2. Буфер клиента, определяющий сколько чисел сервер будет выделять каждому клиенту за раз.
3. Время в миллисекундах (N). Если одно число занимает больше N миллисекунд на обработку, то каждые N миллисекунд серверу будет отправлен отчет об этом. Отчеты записываются в файл журнал вычислений² как строки в формате « X ms >> Y », где X – время в миллисекундах, сколько заняла обработка одного числа, кратно N ; Y – само число.
4. Время в миллисекундах (L). Каждые L миллисекунд будет производиться запись в журнал вычислений² соответствующая интервалу, который был вычислен конкретным клиентом за последние L миллисекунд. Каждый раз, когда клиент заканчивает работу на предварительно выделенном ему интервале, этот интервал записывается в журнал, и таймер авто-записи сбрасывается, т.е. если клиент успевает обработать выделенный интервал быстрее, чем за L миллисекунд, авто-запись не произойдет.
5. Время в минутах (L). Раз в L минут сервер будет отправлять электронное письмо на почту с отчетом.
6. Лимит на кол-во рекурсивных действий для одного числа.

(v2.2) Для инициализации вычисления необходимо написать следующую команду в стандартный ввод сервера:

```
init [1] [2] [3] [4]
```

Разберем аргументы:

1. Число, с которого начнется вычисление.
2. Кол-во чисел для счета. Конечное число интервала равно сумме первого и второго аргументов.
3. Время в минутах (L). Раз в L минут сервер будет отправлять электронное письмо на почту с отчетом.
4. Лимит на кол-во рекурсивных действий для одного числа.

Для корректного выключения сервера используйте команду:

x

Примечание: При аварийном выключении сервера информация о прогрессе вычисления клиентов не сохранится и единственным способом восстановления будет –вручную переписать файлы сохранений через журналы вычисления².

² Журнал вычислений – файл с именем COMPLOG_%.txt, где % – UUID клиента. Создается отдельный журнал для каждого клиента. В него записываются интервалы, которые клиент успешно вычислил, а также отчеты о том, что какое-то число занимает слишком много времени для обработки.

Эта версия документации была написана 11 сентября 2016 г. для версий ПО 2.1-a7 и 2.2-a12 и не является последней.