УДК 519.687.1 DOI: 10.17587/it.27.242-248

П. Е. Голосов, канд. техн. наук, декан факультета информационных технологий и анализа данных, e-mail: pgolosov@gmail.com,
Институт ЭМИТ, Российская академия народного хозяйства и государственной службы при Президенте РФ (РАНХиГС), Москва,
И. М. Гостев, д-р техн. наук, вед. науч. сотр., e-mail: igostev@gmail.com,
Институт проблем передачи информации им. А. А. Харкевича Российской академии наук (ИППИ РАН), Москва

Оптимизация распределения потока задач поиска хеш-решений при априорно заданной сложности решений

Рост числа вычислительно трудоемких задач в условиях развития цифровой экономики (в рамках внедрения блокчейн-решений, распределенных реестров и пр.) требует все больше вычислительных ресурсов. При этом пользователи для минимизации расходов стремятся перенести вычислительный процесс в облако, а владельцы облачных сервисов вынуждены искать решения для повышения эффективности их использования. В работе рассматриваются подходы, позволяющие рассмотреть возможности оптимизации использования параллельных вычислительных ресурсов для поступающих наборов ресурсоемких задач, анализируются различные подходы к стратегии назначения задач на вычислительные ресурсы. Представлены результаты модельных экспериментов, учитывающих распределение заданий, параметризованных предельным временем выполнения в рамках моделирования исполнения соглашения с пользователем об уровне сервиса.

Ключевые слова: *оптимизация, параллельные вычисления, планирование вычислений, оценка эффективности*

Введение

В последнее время активно развивается тенденция переноса сложных и трудоемких вычислений с аппаратуры пользователя на облачные сервисы [1—3]. Это обусловлено тем, что пользователи стремятся сократить расходы на содержание сложной вычислительной системы и переложить их на специализированные сервисы облачных операторов. При этом некоторые виды вычислительных процессов требуют выделения существенных аппаратно-программных ресурсов. Одной из наиболее известных задач такого типа являются задача майнинга [4] при генерации различных криптовалют. При ее решении основные ресурсы отводятся на вычисление хеш-функций методами, основанными на простом переборе. Трудоемкость и время вычислений возрастает с увеличением длины хеша. Кроме задач майнинга в последнее время технологии блокчейна стали активно применять в распределенных базах данных, реестрах, кадастрах и т. п. Все эти задачи фактически решаются методами перебора, а характерным признаком таких задач является непредсказуемое время получения решения. Для обслуживания таких задач владельцы облачных сервисов стремятся минимизировать свои расходы посредством повышения эффективности использования своих вычислительных ресурсов. Однако в связи с необходимостью применения процедуры подтверждения работы при внедрении блокчейн-решений, в которых основное время занимает поиск нужного хеш-решения [5], эффективность функционирования сложных, гетерогенных вычислительных облачных систем постоянно снижается [6, 7]. Такое явление обусловлено тем фактом, что время, за которое необходимо получить решение, является случайной величиной, распределенной по равномерному закону (время, затрачиваемое на простой перебор решений). При этом специализированные планировщики различного уровня, позволяющие учитывать задачи такого типа, отсутствуют.

Все это приводит к необходимости рассмотрения ситуаций, при которых некоторый поток задач, который в дальнейшем будем называть *пакетом*, поступает в распределенную систему. Далее задачи, как правило, разделяются на несколько одинаковых частей, каждая из которых будет выполняться на своем вычислительном устройстве (процессоре). При окончании вычислений на одном процессоре (получении значения хеша требуемой сложности) все части задачи прекращают работу, а занятые ресурсы освобождаются.

Итак, основная цель настоящей работы заключается в моделировании функционирования

распределенной вычислительной системы для отыскания значений хеш-функций (решений) с заданными свойствами сложности (далее — операция), адресуемого пользователем специализированному облачному сервису. Результатом моделирования будут верхние оценки директивных сроков выполнения. Основным требуемым условием здесь является возможность представления задачи в виде большого числа малых вычислительных подзадач, называемых элементарными заданиями, каждое из которых оперирует с подмножеством исходного множества.

1. Постановка задачи

Рассмотрим задачу поиска хеш-функций как задачу перебора чисел с поиском заданных свойств. Примером такой операции служит подбор одноразовых чисел, например, для сети Биткоин [8], где число вариантов для поля ***nonce*** составляет 2^{32}. В силу свойства ее аддитивности разбиение пространства подстановок значений в функцию таково, что каждое элементарное задание независимо от остальных и все элементарные задания могут выполняться одновременно (параллельно) на различных вычислительных устройствах параллельной вычислительной системы (сети). Будем считать, что доступ к таким вычислителям предоставляется облачным сервисом[1]. В качестве искомого решения будем принимать первый отвечающий входным ограничениям результат (например, отыскание хеша заданной сложности, такого как SHA-256 [9, 10]), полученный на любой из параллельных ветвей исполнения операции для элементарных заданий. Будем понимать, что каждая из таких задач при выделении на ее решение всего ресурса системы решается за случайное время с равномерном законом распределения [11]. В работах [12, 13] показано, что процесс решения представляет собой процесс восстановления. Далее будем рассматривать различные варианты планирования процесса вычислений с учетом распараллеливания пакетов на вычислительной системе с распределенными ресурсами. При этом будем считать, что процесс решения будет происходить в распределенной, параллельной, гомогенной среде.

[1]В данной работе не учитываются времена: пересылки заданий от пользователя к вычислителю, загрузки и освобождения процессоров и ресурсов, необходимых для выполнения заданий.

2. Определения и процесс моделирования

Пусть в массово-параллельную вычислительную систему [14-16], обладающую единственным типом вычислительных устройств (например, GPU [17], далее процессор), на обработку поступает пакет из $m \geqslant 2$ задач.

Предположим, что система может одновременно обрабатывать как одно, так и несколько заданий на всех работоспособных в данный момент процессорах. В свою очередь отдельное задание может быть разделено на подзадания, каждое из которых может выполняться независимо от других на отдельном процессоре. При этом на каждом процессоре в единицу времени может обрабатываться только одно задание или его часть. Также будем считать, что задания поступают от пользователей одновременно в виде некоторого пакета, без каких-либо признаков или отношений предшествования между ними. Каждое задание из пакета может быть поставлено на обработку в любое определенное некоторым диспетчером время.

При полном владении ресурсом системы i-я задача решается за случайное время X_i, $i = \overline{1,m}$. Пусть функции распределения случайных величин X_i принадлежат множеству $\Im = \{F_1, \ldots, F_m\}$. Не ограничивая общности, полагаем, что X_i принимает значения на отрезке [0,1], т. е. $F(0) = 0$ и $F_i(x) = 1$ при $x \geqslant 1$, а для всех X_i, $i = \overline{1,m}$, существует конечный первый момент. ***Планом*** выполнения пакета задач Z будем называть набор из m векторов, $r_i = (r_1^{(i)}, \ldots, r_m^{(i)})$, $i = \overline{1,m}$, в котором $r_\upsilon^{(1)}$, $\upsilon = \overline{1,m}$, — это доля ресурса, выделяемая υ-й задаче при запуске пакета Z на обработку, а $r_\upsilon^{(i+1)}$, $\upsilon = \overline{1,m}$, — доля ресурса, выделяемая υ-й задаче после того как завершено решение i задач, $i = \overline{1,m-1}$. Для любого плана $z_i \in Z$, $i = \overline{1,m}$, и $x > 0$ обозначим $t_z(x)$ — условное математическое ожидание времени пребывания в системе k-й задачи при условии, что ее длина есть x: $t_Z(x) = M(t_{x_i}^Z | X_i = x)$, а момент времени пребывания i-й задачи в системе обозначим T_i.

Регулярные планы выполнения пакета задач определим следующим образом. Зафиксируем натуральное число $k : 1 \leqslant k \leqslant m$. Из имеющегося пакета выберем случайным образом k задач, разделим ресурс на k частей и начнем обрабатывать отобранные задачи одновременно. После окончания решения одной или нескольких задач освободившийся ресурс разделим поровну на все незавершенные задачи и продолжим их обработку. Будем продолжать процесс до тех пор, пока все k задач не будут

решены. После этого описанная процедура повторяется до тех пор, пока общий список имеющихся задач не будет исчерпан [18].

Как отмечено ранее, ресурс вычислительной системы и любая его часть кратны m. Аналогично предполагается делимость рассматриваемых задач на произвольное число независимых вычислительных ветвей.

Введем в рассмотрение два граничных плана при $k = 1$ и $k = m$, обозначив их как **1-*регулярный*** и ***m*-регулярный**, соответственно. При этом 1-*регулярный* план будем называть последовательным, а *m*-*регулярный* — параллельным. Очевидно, что общее время решения всех задач пакета не зависит от плана и равно $X_1 + X_2 + ... + X_m$.

Обозначим X вектор $(X_1, ..., X_m)$, где X_i — время решения i-го задания — случайная величина, имеющая функцию распределения $F(x)$. Упорядочим его компоненты по возрастанию и перенумеруем. Полученный вариационный ряд обозначим $X^* = (X_1^*, ..., X_m^*)$, для него: $X_1^* + X_2^* + ... + X_m^* = X_1 + X_2 + ... + X_m$.

Введем следующие обозначения:

$\overline{S}_m = \frac{1}{m}(X_1 + ... + X_m)$ — среднее время решения задач из множества Z; T_{seq} — среднее время пребывания задач в системе при последовательном плане; T_{par} — среднее время пребывания задач в системе при параллельном плане.

Утверждение 1. Для параллельного плана среднее время пребывания задач в системе определяется следующим равенством:

$$T_{par} = (2m-1)\overline{S}_m - \frac{2}{m}\sum_{k=1}^{m}(k-1)X_k^*.$$

Доказательство. Поскольку при делении ресурса на k частей время решения задач увеличивается в k раз, и освободившийся ресурс делится поровну, справедливы равенства:

$$T_1 = mX_1^*;\ T_2 = mX_1^* + (m-1)(X_2^* - X_1^*);$$
$$T_k - T_{k-1} = (m-k+1)(X_k^* - X_{k-1}^*);$$
$$T_k = X_1^* + ... + X_k^* + (m-k)X_k^*,\ k = \overline{2,m}.$$

После выполнения преобразований получаем:

$$T_{par} = \frac{1}{m}\sum_{k=1}^{m}[(X_1^* + ... + X_k^*) + (m-k)X_k^*] =$$
$$= \frac{1}{m}\left(\sum_{k=1}^{m}(m-k+1)X_k^* + (m-k)X_k^*\right) =$$
$$= \frac{2m-1}{m}\sum_{k=1}^{m}X_k^* - \frac{2}{m}\sum_{k=1}^{m}(k-1)X_k^* =$$
$$= \frac{(2m-1)}{m}\sum_{k=1}^{m}X_k - \frac{2}{m}\sum_{k=1}^{m}(k-1)X_k^*.$$

Утверждение доказано.

Замечание. Пусть время выполнения каждого задания — случайная величина, равномерно распределенная на отрезке [0,1]. Поскольку X_k^* — случайная величина, значение которой ограничено конечным интервалом, поэтому X_k^* имеет бета-распределение [19] с параметрами $(k,\ m - k + 1)$. Тогда для математического ожидания X_k^* справедливо равенство

$$MX_k^* = \frac{k}{m+1}.$$

Отсюда следует, что при использовании параллельного плана математическое ожидание времени пребывания k-го задания в системе равно

$$MT_{par}^w = \frac{k(2m-k+1)}{2(m+1)}.$$

При этом математическое ожидание времени задержки T_{del} между выполнением k-го и $(k-1)$-го заданий линейно убывает с ростом k и равно

$$MT_{del} = \frac{m-k+1}{m+1}.$$

Лемма 1. Пусть случайные величины X_i, $i = \overline{1,m}$, независимы, имеют функцию распределения $F(x)$ и имеют конечный первый момент. Тогда справедливо равенство

$$S(F) = M\sum_{k=1}^{m}(k-1)X_k^* = \binom{m}{2}M\max(X_1, X_2).$$

Доказательство. Поскольку функция распределения наибольшей порядковой статистики X_n имеет вид $F_n(x) = P\{X_n \leqslant x\} = P^n(x)$ [20], откуда

$$F_1(x) = P\{X_1 \leqslant x\} = 1 - P(X_1 > x) = 1 - [1 - P(x)]^n$$

и в более общем виде:

$$P\{X_k^* \leqslant x\} = \sum_{r=k}^{m}\binom{m}{r}F^r(x)(1-F(x))^{m-r},$$

то отсюда следует, что

$$\sum_{k=1}^{m}(k-1)MX_k^* =$$
$$= \int_0^1 x d\left(\sum_{k=1}^{m}(k-1)\sum_{r=k}^{m}\binom{m}{r}F^r(x)\left(1-F^{m-r}(x)\right)\right) =$$
$$= \int_0^1 x d\left(\sum_{r=1}^{m}\binom{m}{r}F^r(x)(1-F(x))^{m-r}\sum_{k=1}^{r}(k-1)\right) =$$
$$= \binom{m}{2}\int_0^1 x d\left(F^2(x)\sum_{r=0}^{m-2}\binom{m-2}{r}F^r(x)(1-F(x))^{m-r}\right) =$$
$$= \binom{m}{2}\int_0^1 x d(F^2(x)) = \binom{m}{2}M\max(X_1, X_2).$$

Последнее равенство верно, поскольку $F^2(x)$ является функцией распределения случайной величины $\max(X_1, X_2)$.

Утверждение 2. При применении последовательного плана условное математическое ожидание среднего времени пребывания задачи в системе относительно σ-алгебры, порожденной случайными величинами X_1, ..., X_m, равно

$$M(T_{seq}\,|X_1,\ldots,X_m) = \frac{m+1}{2}\overline{S}_m.$$

Доказательство. Если задачи из пакета Z выполняются последовательно в порядке i_1, ..., i_m, то время пребывания k-й задачи в системе равно $T_k = X_{i_1} + \ldots + X_{i_k}$, а среднее время прохождения задачи в системе равно

$$T_{seq} = \frac{1}{m}\sum_{k=1}^{m} T_k = \frac{1}{m}\sum_{k=1}^{m}(m-k+1)X_{i_k}.$$

Вероятность выбора каждой из последовательностей i_1, ..., i_m равна $(m!)^{-1}$. Следовательно,

$$M(T_{seq}\,|X_1,\ldots,X_m) =$$

$$= \frac{1}{m}\sum_{k=1}^{m}(m-k+1)\frac{(m-1)!}{m!}(X_1+\ldots+X_m) =$$

$$= \frac{1}{m^2}(X_1+\ldots+X_m)\sum_{k=1}^{m}(m-k+1) = \frac{m+1}{2}\overline{S}_m.$$

Доказательство завершено.

Теорема 1. Пусть случайные величины X_1, ..., X_m независимы, и множество F состоит из двух функций: $F_{m-l+1}(x) = \ldots = F_m(x) = G(x) = 0$; для $x < 1$; $l = \overline{0,m}$, и

$$F_1 = \ldots = F_{m-l}(x) = F(x).$$

Тогда для математических ожиданий среднего времени пребывания задачи в системе справедливы равенства

$$MT_{seq} = \frac{m+1}{2m}(l + (m-l)\mathrm{M}X_1);$$

$$MT_{par} = \frac{l^2}{m} + \frac{(m-l)}{m}\times$$

$$\times\{(2m-1)MX_1 - (m-l-1)M\max(X_1,X_2)\};$$

$$\Delta = M(T_{par} - T_{seq}) =$$

$$= \frac{l}{m}\left(l - \frac{m+1}{2}\right) + \frac{(m-l)}{m}\times$$

$$\times\left\{\frac{3}{2}(m-1)MX_1 - (m-l-1)M\max(X_1,X_2)\right\}.$$

Доказательство следует из утверждений 1 и 2 и теоремы. Кроме того, поскольку $X^*_{m-l+1} = \ldots = X^*_m = 1$ и $\overline{S}_m = \frac{1}{m}(l + (m-l)MX_1)$, то

$$MT_{par} = \frac{2m-1}{m}(l + (m-l)MX_1) -$$

$$-\frac{2}{m}\left\{\sum_{k=m-l+1}^{m}(k-1) + \sum_{k=1}^{m-l}(k-1)MX^*_k\right\}.$$

Замечание 1. Случай $l = 0$ в теореме 1 характерен для естественного протекания процесса обработки. Случай $l > 0$ связан с различными нарушениями в процессе постановки задач и надежностью вычислительных средств.

Замечание 2. Пусть случайные величины X_1, ..., X_m независимы в совокупности. Какие-либо l, $l = \overline{0,m}$, из них принимают значение "1" с вероятностью $p = 1$, а остальные равномерно распределены на отрезке [0, 1].

Тогда:

$$MX_1 = \frac{1}{2},\ M\max(X_1,X_2) = \frac{2}{3},$$

$$\Delta = \frac{(m-1)(m+l)}{12m} + \frac{l(l-1)}{3m} \geqslant 0.$$

Это означает, что последовательный план совпадает с параллельным только при $m = 1$. В частности, при $l = 0$, $\Delta = (m-1)/12$, а при $l = m$: $\Delta = (m-1)/2$.

Замечание 3. Для произвольной функции распределения $F(x)$ и $l = 0$ можно записать:

$$T_{seq} = \frac{m+1}{2}MX_1;$$

$$T_{par} = (2m-1)MX_1 - (m-1)M\max(X_1,X_2);$$

$$\Delta = (m-1)\left(\frac{3}{2}MX_1 - M\max(X_1,X_2)\right).$$

Последнее означает, что качество плана существенным образом зависит от такого параметра распределения $F(x)$, как $\left(\frac{3}{2}MX_1 - M\max(X_1,X_2)\right)$.

Следует обратить внимание, что полученные аналитические результаты существенным образом используют гипотезу о симметричной неопределенности, которая в реальности выполняется далеко не всегда, если вообще справедлива.

3. Обсуждение и анализ результатов моделирования

Для исследования предложенной модели были проведены следующие вычислительные эксперименты. Набор заданий, удовлетворяющий определенным свойствам, генерировался случайным образом. Далее в первом случае все задания пакета выполнялись одно за другим, при этом каждой из задач последовательно от-

давался весь имеющийся в наличии вычислительный ресурс. Во втором случае на выполнение был поставлен весь пакет одновременно, т. е. имеющиеся задания выполнялись параллельно. При этом мощность всех процессоров первоначально была разделена на все задачи, а освободившиеся ресурсы (процессоры) делились между оставшимися в зависимости от выбранной стратегии. Пакеты формировались как из задач со случайным временем окончания решения, равномерно распределенным на определенном временном интервале, так и из задач с фиксированным временем счета. Целью эксперимента являлось сравнение результатов обработки системой пакетов задач, обладающих различными свойствами, при параллельной и последовательной дисциплинах обслуживания.

На основе анализа результатов рассмотренных моделей относительно выбора планов решения необходимо обратить внимание на некоторые практически значимые аспекты.

Среднее время пребывания заданий в системе является лишь интегральной характеристикой случайного процесса

$$\mu(t) = \sum_{i=1}^{m} I(T_i \leqslant t),$$

где $I(A)$ — индикаторная функция [21] события A, т. е. $I(a) = 1$, если $a \in A$ и $I(a) = 0$ в противном случае.

В случае последовательного плана решения задач поведение процесса $\mu(t)$ совпадает с поведением процесса $\nu(t)$, определенного следующим образом. Если при $n \to \infty$ функции

$$F_n(x) = 0, \quad 0 \leqslant x < \frac{1}{n},$$

$$F_n(x) = \sum_{\nu \leqslant nx} p_\nu^{(n)}, \quad \frac{1}{n} \leqslant x \leqslant 1,$$

сходятся на отрезке [0, 1] равномерно к непрерывной функции $F(x)$, тогда конечномерные распределения процесса $\nu_n(t)$ слабо сходятся к конечномерным распределениям процесса восстановления

$$\nu(t) = \max\{k : X_0 + \ldots + X_k \leqslant t\}, \; t \geqslant 0.$$

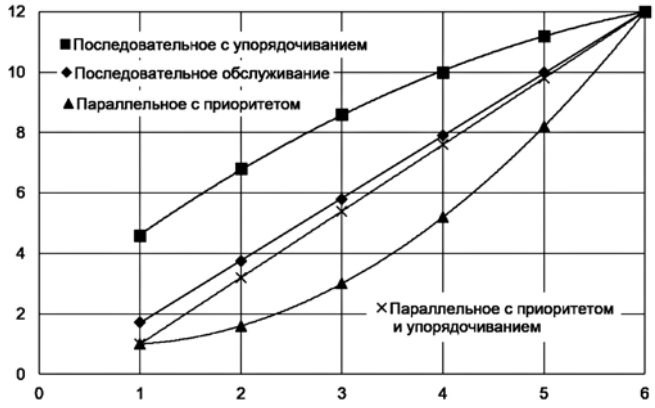
Для случая параллельного плана при $m > 1$ требуется исследование поведения уже не процесса восстановления, а считающего процесса [22], что связано со значительными теоретическими и вычислительными сложностями. Тем не менее, в целях исследования поведения системы при решении пакетов задач было проведено статистическое моделирование, условия и основные результаты которого приведены ниже [23].

Считалось, что в систему одновременно поступает пакет из m задач, каждая из которых определяется случайной выборкой без возвращений. Предполагалось, что каждая такая задача имеет решение. Максимально возможное время решения такой задачи $\overline{\tau_i}$, $i = \overline{1,m}$, для каждой группы экспериментов определялось отдельно, при этом длительность решения каждой задачи τ_i полагалась либо случайной величиной, равномерно распределенной на отрезке $[0; \overline{\tau_i}]$, либо была зафиксирована и равна $\overline{\tau_i}$. Кроме того, для каждой задачи назначался директивный срок окончания ее выполнения d_i, $i = \overline{1,m}$.

Были рассмотрены четыре стратегии решения пакета задач.

- ***Последовательное выполнение***. В каждый момент времени системой решается ровно одна задача, при этом весь доступный ресурс отдается этой задаче. По окончанию работы предыдущего процесса начинается выполнение последующего. Задачи решаются в случайном порядке.
- ***Последовательное выполнение с упорядочиванием*** по максимально возможному времени решения задачи $\overline{\tau_i}$. Имеющийся пакет задач упорядочивается по возрастанию $\overline{\tau_i}$ и перенумеровывается, на первое место помещается априорно самая "короткая" задача, на последнее — самая "объемная".
- ***Параллельное выполнение*** с выделением приоритетных задач. Это означает, что все задачи начинают выполняться одновременно, при этом ресурс между ними распределяется пропорционально отношению $\overline{\tau_i}/d_i$. После завершения какой-либо задачи освободившийся ресурс распределяется между оставшимися также пропорционально отношению $\overline{\tau_i}/d_i$.
- ***Параллельное выполнение с предпочтением коротких задач*** и упорядочиванием по максимально возможному времени решения задачи $\overline{\tau_i}$. Это означает, что все задачи начинают решаться одновременно, при этом первоначально ресурс между ними распределяется пропорционально величине $\overline{\tau_i}/d_i$. После завершения очередной задачи весь освободившийся ресурс отдается той задаче, у которой меньше максимально возможное время счета (наименьшая сложность).

Для сравнения параллельного и последовательного планов решения задач был проведен ряд модельных экспериментов в целях выявления особенностей процедуры их прохождения. Для каждой модели генерировалось 10^6 случайных реализаций наборов данных, вы-



Число решенных задач при различных планах обслуживания

числялись выборочные средние и среднеквадратичные отклонения исследуемых величин. Результаты счета сравнивались по следующим показателям: среднее время решения i-й задачи, средняя разность моментов завершения i-й и $(i-1)$-й задач, их среднеквадратичное отклонение и другие параметры.

На рисунке приведены результаты моделирования процесса обработки пакетов объемом 12 задач, максимальное время решения каждой из которых $\overline{\tau_i}$ являлось равномерно распределенной случайной величиной, нормированной следующим образом:

$$\overline{\tau_i} = k\alpha_i,$$

где α_i — случайная величина, равномерно распределенная на отрезке [0,1]; $k = \dfrac{12}{\sum\limits_{i=1}^{12} \alpha_i}$ — нормирующий коэффициент.

Директивный срок окончания решения для всех задач одинаков и составляет 12 единиц времени. Время решения каждой задачи — случайная величина, равномерно распределенная на интервале $[0, \overline{\tau_i}]$.

Результаты экспериментов полностью подтвердили теоретические положения, сформулированные выше. В зависимости от выбранной общей стратегии решения пакетов ресурсоемких задач путем группирования и упорядочивания их по некоторым априорным или статистическим характеристикам имеется реальная возможность согласовывать график решения с директивными сроками выполнения, обеспечив условия соглашения "потребитель—поставщик".

Заключение

Полученные результаты позволяют построить некоторую стратегию управления ресурсами распределенной параллельной вычислительной системы, на основе которой в дальнейшем будут разрабатываться планировщики различного уровня, обеспечивающие максимальную эффективность таких систем. Вопросы экономических оценок эффективности вычислений также будут рассмотрены в отдельной статье.

Список литературы

1. **Armbrust M., Fox A., Griffith R.** et al. Above the Clouds: A Berkeley View of Cloud Computing. UCBerkeley Reliable Adaptive Distributed Systems Laboratory, 2009.
2. **Toby Velte, Anthony Velte, Robert C. Elsenpeter.** Cloud Computing, A Practical Approach. McGraw Hill Professional, 2009. 400 p.
3. **Мирин С., Башилов Г., Патрикеев Д.** Cloud providing 2018—2022: economy, strategies, business-models. iKS-Consulting, 2018. 180 с.
4. **Rosenfeld M.** Analysis of Bitcoin Pooled Mining Reward Systems // CoRR: Computing Research Repository abs/1112.4980, arXiv:1112.4980 (2011).
5. **Carter J. L., Wegman M. N.** Universal classes of hash functions // J. Comp. Sys. Sci. 1979. Vol. 18. P. 143—154.
6. **Narayanan A., Bonneau J.** et al. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction Hardcover Princeton University Press, 2016. 336 p.
7. **Cohen R.** Global Bitcoin Computing Power Now 256 Times Faster Than Top 500 Supercomputers Combined // Forbes. 2013.
8. **Nakamoto S.** Bitcoin: //A Peer-to-Peer Electronic Cash System. 2008. URL: https://bitcoin.org/bitcoin.pdf (дата обращения: 08.07.2020).
9. **Kraft D.** Difficulty control for blockchain-based consensus systems // Peer-to-Peer Networking and Applications. 2015. P. 1—17.
10. **Frankenfield J.** Cryptocurrency Difficulty. 2020 [Электронный ресурс]. URL: https://www.investopedia.com/terms/d/difficulty-cryptocurrencies.asp. Электронный ресурс (дата обращения: 08.07.2020).
11. **Корн Г., Корн Т.** Справочник по математике для научных работников и инженеров. М.: Наука, 1970.
12. **Кокс Д. Р., Смит В. Л.** Теория восстановления. Пер. с англ. М.: Сов. радио, 1967.
13. **Математическая** энциклопедия. Т. 1 (А — Г). М.: Советская Энциклопедия, 1977. 1152 с.
14. **Таненбаум Э., Стен М.** Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
15. **Almasi G. S., Gottlieb A.** Highly Parallel Computing. Benjamin-Cummings publishers, Redwood City, CA, 1989.
16. **Корнеев В. В.** Архитектура вычислительных систем с программируемой структурой. Новосибирск: Наука, 1985.
17. **Орлов С.** Гетерогенные вычисления и новые серверные платформы // ИнформКурьер-Связь. 2019. № 3. С. 44—50.
18. **Воеводин В. В., Воеводин Вл. В.** Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
19. **Королюк В. С., Портенко Н. И.,Скороход А. В., Турбин А. Ф.** Справочник по теории вероятностей и математической статистике. М.: Наука, 1985. 640 с.
20. **Дэйвид Г.** Порядковые статистики. М.: Наука, 1979.
21. **Кормен Т., Лейзерсон Ч., Ривест Р.** Алгоритмы: Построение и анализ М.: МЦНМО, 2001.
22. **Кабанов Ю. М., Липцер Р. Ш., Ширяев А. Н.** Слабая и сильная сходимость распределений считающих процессов // Теория вероятности и ее применения. 1983. Т. XXVIII, № 2. С. 288—319.
23. **Голосов П. Е., Козлов М. В., Малашенко Ю. Е.** и др. Модель системы управления специализированным вычислительным комплексом. М.: ВЦ РАН, 2010. 48 с.

P. E. Golosov, Cand. of Tech. Sc. (Ph.D.), Dean of the Faculty of Information Technologies and Data Analysis, e-mail: golosov-pe@ranepa.ru,
Institute of EMIT of the Russian Academy of National Economy and Public Administration under the Russian President (RANEPA),
Moscow, Russian Federation, 119571, Moscow,
I. M. Gostev, D. Sc., e-mail: igostev@gmail.com,
Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute), Moscow, 127051, Russian Federation

Optimization of the Distribution of Hash Calculation Tasks Flow at a Priori Given Complexity

The increasing number of computationally intensive tasks induced by digital economy development (within the framework of implementing block-chain solutions, distributed ledgers, etc.) requires more and more computational resources. At the same time users tend to move the computational process to the cloud in order to minimize costs, and the owners of cloud services are forced to look for solutions to improve their efficiency. In this paper we consider approaches that allow optimize the use of parallel computing resources for incoming sets of resource-intensive tasks, analyze different approaches to the strategy of assigning tasks to computing resources. The results of modelling experiments are provided taking into account task distribution, parameterized by execution time limit within modeling the service level agreement with the user.

Keywords: *optimization, parallel computations, computation planning, efficiency estimation*

DOI: 10.17587/it.27.242-248

References

1. **Armbrust M., Fox A., Griffith R. et al.** Above the Clouds: A Berkeley View of Cloud Computing, UCBerkeley Reliable Adaptive Distributed Systems Laboratory, 2009.
2. **Velte T., Velte A., Elsenpeter R. C.** Cloud Computing, A Practical Approach by McGraw Hill Professional, 2009, 400 p.
3. **Mirin S., Bashilov G., Patrykeev D.** Cloud providing 2018—2022: economy, strategies, business-models, *iKS-Consulting*, 2018, 180 p. (in Russian).
4. **Rosenfeld M.** Analysis of Bitcoin Pooled Mining Reward Systems. CoRR: Computing Research Repository abs/1112.4980, arXiv:1112.4980 (2011).
5. **Carter J. L., Wegman M. N.** Universal classes of hash functions, *J. Comp. Sys. Sci.,* 1979, vol. 18, pp. 143—154.
6. **Narayanan A., Bonneau J., Felten E., Miller A., Goldfeder S.** Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction Hardcover, Princeton University Press, 2016, 336 p.
7. **Cohen R.** Global Bitcoin Computing Power Now 256 Times Faster Than Top 500 Supercomputers Combined, *In Forbes, 2013.*
8. **Nakamoto S.** Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, available at: https://bitcoin.org/bitcoin.pdf (circulation date: 08.07.2020).
9. **Kraft D.** Difficulty control for blockchain-based systems, *Peer-to-Peer Networking and Applications*, 2015, pp. 1—17.
10. **Frankenfield J.** Cryptocurrency Difficulty. 2020, available at: https://www.investopedia.com/terms/d/difficulty-cryptocurrencies.asp. Electronic resource (date of access: 08.07.2020).
11. **Korn G., Korn T.** Handbook of Mathematics for Researchers and Engineers, Moscow, Science, 1970 (in Russian).
12. **Cox D. R., Smith V. L.** Theory of Restoration, Per. English, Moscow, 1967 (in Russian).
13. **Vinogradov I. M.** et al. Mathematical Encyclopedia. P. 1 (A—G), Moscow, The Soviet Encyclopedia, 1977, 1152 p. (in Russian).
14. **Tanenbaum E., Stan M.** Distributed systems. Principles and paradigms, SPb, 2003, 877 p. (in Russian).
15. **Almasi G. S., Gottlieb A.** Highly Parallel Computing. Benjamin-Cummings publishers, Redwood City, CA, 1989.
16. **Korneev V. V.** Architecture of Computing Systems with Programmable Structure, Novosibirsk. Nauka, 1985 (in Russian).
17. **Orlov S.** Heterogeneous computations and new server platforms, *InformCourier Communication,* 2019, no. 3, pp. 44—50 (in Russian).
18. **Voevodin V. V., Voevodin V. V.** Parallel computing, St. Petersburg, BHV Petersburg, 2002 (in Russian).
19. **Korolyuk V. S., Portenko N. I., Skorokhod A. V., Turbin A. F.** Handbook on Probability Theory and Mathematical Statistics, Moscow, Nauka, 1985, 640 p. (in Russian).
20. **David G.** Ordinary statistics, Moscow, Nauka, 1979 (in Russian).
21. **Korman T., Layserson C., Rivest R.** Algorithms: Construction and Analysis, Moscow, ICSEMO, 2001.
22. **Kabanov Yu. M., Liptser R. S., Shiryaev A. N.** Weak and strong convergence of distributions of counting processes, Theory of Probability and its Application, Moscow, 1983, vol. XXVIII, ch. 2, pp. 288—319 (in Russian).
23. **Golosov P. E., Kozlov M. V., Malashenko Yu.** Model of control system for specialized computing complex, Moscow, VZ RAN, 2010, 48 p. (in Russian).

Уважаемые коллеги!

С марта по ноябрь 2021 г. проходит X Юбилейная Всероссийская с международным участием научно-техническая конференция "Проблемы разработки перспективных микро- и наноэлектронных систем" (МЭС-2021).

В этом году конференция проходит в виде Интернет-форума с проведением заседаний научных секций разной тематической направленности в online-режиме. Конференция завершится проведением очного Пленарного заседания, на котором будут подведены ее итоги, а также секции "Презентации новых микроэлектронных проектов, САПР и готовых продуктов", на которой партнеры и спонсоры конференции представят доклады о своих разработках.

Прием докладов осуществляется с 01 марта по 01 августа 2021 г. Принятые доклады будут опубликованы на WEB-сайте конференции, а также в четырех выпусках трудов конференции, которые будут издаваться по мере поступления, рецензирования и редакционной подготовки текстов статей. Как и в 2020 году, будут проведены конкурсы на лучшие доклады с призами для победителей.

Более подробную информацию можно найти на WEB-сайте конференции МЭС-2021: http://www.mes-conference.ru.
Участие в конференции МЭС-2021 бесплатное. Ждем ваших докладов!

Оргкомитет МЭС-2021