Предицирующие каналы в параллельном программировании: возможное применение в математическом моделировании процессов в сплошных средах

**Человек:** В данной работе рассматривается проблема применения предикции в классическом и параллельном программировании задач математического моделирования, основанных на численном интегрировании дифференциальных уравнений в частных производных. Предикция может использоваться для замены фрагментов математической модели более простыми приближенными соотношениями, для опережающего прогнозирования значений принимаемых в ходе параллельного счета данных, для прогнозирования времени исполнения фрагментов параллельной программы при решении вопроса о применении последовательного или параллельного алгоритма и при балансировке загрузки отдельных процессоров вычислительной системы. Для формализации видов предикторов и определения математических соотношений, позволяющих их вычислить, используются подходы и методы вычислительной математики (теории интерполяции и экстраполяции). Для определения программных реализаций предикторов используются подходы, характерные для инженерии параллельного программирования. Полученные результаты проверяются экспериментально. В работе впервые предложен новый вид технологических средств предикции для использовании в параллельном программировании – предикционно-решающие каналы. Предложены два вида каналов – авторегрессионные точечные и линейные (явные или неявные) коллективного решения. Изложены математические аспекты предикции в таких каналах, кратко описаны базовые средства программирования. Показано, что совмещение каналов со средствами предсказания данных упрощает программирование ряда алгоритмов, связанных с численным моделированием, и позволяет, в частности, осуществлять скрытые переходы от явных разностных схем к неявным, отличающихся большей устойчивостью счета, а также от последовательных алгоритмов счета к параллельным. На примере задачи численного интегрирования нестационарного уравнения теплопроводности показано, что адекватное применение каналов в ряде случаев позволяет ускорить расчет на многоядерных вычислительных системах.

**Key words:** ускорение счета, разностные схемы, дифференциальные уравнения, математическое моделирование, линейная экстраполяция, предсказание данных, каналы, параллельное программирование, аппроксимация разностных схем, персептроны

=================================

**FastText\_KMeans\_Clean:** Далее организуется цикл отправки "пустых" данных в этот канал с приемом из него данных в строгом предикционном режиме – это, фактически, будут результаты дальнейшего эквивалентного счета по неявной разностной схеме , отличающейся заметно большей вычислительной устойчивостью. Авторегрессионные точечные каналы. vector<\_funnel \*> refs;. funneled\_perceptron\_out \* outpi = new funneled\_perceptron\_out(true, false, refs, "iPFUNNEL", N, 1E-5);. cout << "NO PREDICTION" << endl;. Время счета на 16 потоках (вариант без предицирующих каналов), ts16, с. 0,742. 2. Для апробации линейных каналов коллективного решения рассмотрим задачу перехода от явной разностной схемы к распараллеленной неявной при численном интегрировании одномерного уравнения теплопроводности с источниковым членом. 2. Выполнение "разгоночного" интегрирования уравнения теплопроводности по явной схеме с отправкой/получением T и S, соответственно, в каналы/из каналов в предикционном режиме с последующей коррекцией предиктора – таким путем обучается, формируя неявную разностную схему, предиктор неявного канала.

**Key words part:** 0.6428571428571429

=================================

**FastText\_KMeans\_Raw/:** Далее организуется цикл отправки "пустых" данных в этот канал с приемом из него данных в строгом предикционном режиме – это, фактически, будут результаты дальнейшего эквивалентного счета по неявной разностной схеме , отличающейся заметно большей вычислительной устойчивостью. Обучение неявного экстраполятора (персептрона) выполняется последовательно для каждого из значений i, с помощью метода наименьших квадратов, путем минимизации функционала. Далее заметим, что независимо от явности/неявности канала, количество ненулевых членов p в Ai обычно выбирается из условия минимума F(Ai,Bi,Ci), однако если для всех p не выполняется условие F(Ai,Bi,Ci) ≤ e, где e -- допустимая погрешность экстраполяции, то выбирается p = 0, в этом случае в расчете участвуют только Bi и Ci. На передающей стороне:. funneled\_perceptron\_in \* inpi = new funneled\_perceptron\_in(true, false, refs, "iPFUNNEL", N, 1E-5);. Предложены два вида каналов – авторегрессионные точечные и линейные (явные или неявные) коллективного решения.

**Key words part:** 0.6785714285714286

=================================

**FastText\_PageRank\_Clean/:** funneled\_predictor\_out \* out = new funneled\_predictor\_out("FUNNEL", 1, 4, 0.01);. funneled\_predictor\_in \* in = new funneled\_predictor\_in("FUNNEL", 1, 4, 0.01);. На передающей стороне:. vector<\_funnel \*> refs;. refs.push\_back(inb->getRef());. На принимающей стороне:. vector<\_funnel \*> refs;. cout << "NO PREDICTION" << endl;.

**Key words part:** 0.32142857142857145

=================================

**FastText\_PageRank\_Raw/:** out->put((void \*) &seq[i], sizeof(double));. cout << buf << "[Предсказано = " << seq1[i] << "] ";. vector<\_funnel \*> refs;. На принимающей стороне:. vector<\_funnel \*> refs;. outpi->put((void \*) seq, N\*sizeof(double));. cout << "NO PREDICTION" << endl;. Ускорение счета, t1/t16.

**Key words part:** 0.3928571428571429

=================================

**Mixed\_ML\_TR/:** а) для решения (более корректного с точки зрения численных методов по сравнению с точечными каналами) тех же задач предварительного расчета значений в узлах расчетной сетки на стыках блоков расчетной области, которые в общем случае, несомненно, зависят и от "соседних" узлов и от значений других переменных в текущем и "соседних" узлах;. Далее организуется цикл отправки "пустых" данных в этот канал с приемом из него данных в строгом предикционном режиме – это, фактически, будут результаты дальнейшего эквивалентного счета по неявной разностной схеме , отличающейся заметно большей вычислительной устойчивостью. Линейный канал коллективного решения выполняет или обычный блокирующий прием или экстраполяцию очередных значений Vi некоторой переменной в точках i = 1…N, где N – число точек. д) сброс основным каналом принятых фиктивных значений Q и их замена результатом предикции T. Как показал эксперимент, данная программа генерирует результат, который с высокой степенью точности (относительная погрешность составляет около 0,001% и объясняется погрешностями решения системы линейных уравнений неявным каналом при поиске предиктора) совпадает с результатами счета контрольной программой, которая реализовывала разгонный расчет по явной схеме и основной расчет по запрограммированной неявной схеме. Время счета программой с разгонным расчетом по явной схеме и основным расчетом на предицирующих линейных каналах (с распараллеленной неявной схемой, выведенной каналом), t16, с. 0,698. Из данных, приведенных в таблице 2, очевидно, что применение предицирующих каналов позволило перейти от явной разностной схемы к неявной (со стандартным распараллеленным решателем методом Гаусса-Зейделя с красно-черной раскраской) и получить определенное ускорение счета (в сравнении с однопоточной программой), в 1,1÷1,69 раза. Совмещение каналов со средствами предсказания данных упрощает программирование ряда алгоритмов, связанных с численным моделированием (прогнозирования времени счета для балансировки загрузки, расчета прогнозных значений переменных, например, на стыках блоков расчетной области при параллельной обработке, и других алгоритмов), и позволяет, в ряде случаев, осуществлять скрытые переходы от явных разностных схем к неявным, отличающихся большей устойчивостью счета, а также от последовательных алгоритмов счета к параллельным. Показано, что применение неявных каналов коллективного решения позволило (при использовании минимальных средств программирования) перейти от последовательной явной схемы счета уравнения теплопроводности к распараллеленной неявной схеме, отличающейся более высокой вычислительной устойчивостью, и получить заметное ускорение счета (в 1,1÷1,69 раза на восьмиядерной 16-потоковой вычислительной системе проекта Google’s Compute Engine).

**Key words part:** 0.8571428571428571

=================================

**MultiLingual\_KMeans/:** 2. Линейные каналы коллективного решения (явные и неявные) , выполняющие экстраполяцию массива значений некоторой переменной, основываясь на предыдущих значениях этой же переменной, а также, возможно, актуальных значений переменных из иных каналов (зависимость канала от иных каналов определяется при его создании). Далее организуется цикл отправки "пустых" данных в этот канал с приемом из него данных в строгом предикционном режиме – это, фактически, будут результаты дальнейшего эквивалентного счета по неявной разностной схеме , отличающейся заметно большей вычислительной устойчивостью. Авторегрессионный точечный канал выполняет или обычный блокирующий прием или экстраполяцию очередных значений Vi некоторой переменной в точках i = 1…N, где N – число точек. которая сводится к решению системы линейных алгебраических уравнений, возникающих из условий равенства первых частных производных по неизвестным коэффициентам нулю. В нашем случае, для получения прогнозных результатов применены авторегрессионные точечные каналы , что сразу же упростило расчетную программу. Ускорение на 16 потоках (вариант без предицирующих каналов), t1/ts16. а) вычисление и отправка во вспомогательный канал величин S;. Такой переход может иметь определенную ценность при разработке программ с нестандартными разностными схемами, для которых не существует стандартных, хорошо распараллеленных математических библиотек.

**Key words part:** 0.75

=================================

**Multilingual\_PageRank/:** Математическое моделирование процессов в сплошных средах – важная технология, применяемая при решении разнообразных научно-технических задач в сферах аэро- и гидродинамики, акустики, электротехники, при исследовании и прогнозировании процессов в атмосфере, океанах и магме, а также в ряде иных областей науки. в) наличие неравномерностей загрузки процессоров. Эти факторы в наибольшей степени негативно влияют на время расчета, соответственно, актуальна задача разработки подходов к их устранению или минимизации их влияния. Это приемлемое, но не вполне удобное решение для перечисленных выше задач, возникающих в ходе численного моделирования (сам вспомогательный характер таких задач говорит о необходимости по возможности "скрыть" их от программирующего специалиста, сделав создаваемый код более простым и логичным). Кроме того, он может (по желанию исследователя-программиста) иметь или не иметь свободный член Bi. cout << "NO PREDICTION" << endl;. Таким образом, потери точности не произошло. Эти затраты обычно нивелируются с ростом размерности задачи, поэтому с ростом размерности ускорение счета в большинстве случаев растет (см. таблицу 2).

**Key words part:** 0.4642857142857143

=================================

**RuBERT\_KMeans\_Without\_ST/:** Это можно осуществить путем обучения нескольких вспомогательных каналов (при разном значении параметра), на основании данных которых основной канал подбирает, например, простые линейные зависимости коэффициентов предиктора от параметра. cout << seq1[i] << " "; // Если предиктор не был вычислен (это возможно в режиме приема с таймаутом). Видимо, в данном случае достаточно существенными оказались дополнительные временные затраты на обслуживание предицирующих каналов – это объясняет тот факт, что ускорение счета оказалось меньше теоретического значения 12÷16 (ожидаемого для системы с 8 ядрами с технологией HyperThreading). Показано, что применение неявных каналов коллективного решения позволило (при использовании минимальных средств программирования) перейти от последовательной явной схемы счета уравнения теплопроводности к распараллеленной неявной схеме, отличающейся более высокой вычислительной устойчивостью, и получить заметное ускорение счета (в 1,1÷1,69 раза на восьмиядерной 16-потоковой вычислительной системе проекта Google’s Compute Engine).

**Key words part:** 0.6428571428571429

=================================

**RuBERT\_KMeans\_With\_ST/:** В самом деле, достаточно организовать некоторое количество итераций вычислительного процесса по явной схеме (при соблюдении условий вычислительной устойчивости) с передачей получаемых данных в неявный канал коллективного решения, который, фактически, аппроксимирует по этим данным неявную схему. Например, если создается локальный (соединяющий процесс с самим собой) одноточечный канал (P = 4, e = 0,01) для приема-передачи вещественных чисел, то его полная инициализация, включающая создание приемного и передающего коннекторов выглядит так:. if (in->get\_timed((void \*) &seq1[i], -1.0)) { // Предикционный режим. 2. Для апробации линейных каналов коллективного решения рассмотрим задачу перехода от явной разностной схемы к распараллеленной неявной при численном интегрировании одномерного уравнения теплопроводности с источниковым членом.

**Key words part:** 0.6071428571428571

=================================

**RUBERT\_page\_rank\_Without\_ST/:** Предикционно-решающие каналы. Для экстраполяции используются:. Число узлов расчетной сетки. Число узлов расчетной сетки. Предложены два вида каналов – авторегрессионные точечные и линейные (явные или неявные) коллективного решения.

**Key words part:** 0.42857142857142855

=================================

**RUBERT\_page\_rank\_With\_ST/:** Авторегрессионные точечные каналы. На передающей стороне:. vector<\_funnel \*> refs;. На принимающей стороне:. vector<\_funnel \*> refs;.

**Key words part:** 0.35714285714285715

=================================

**RUSBERT\_KMeans\_Without\_ST/:** Важно заметить, что вполне допустима работа канала в случае, если передающая и принимающая стороны совпадают . В параллельном программировании ряда задач моделирования процессов в сплошных средах такие каналы могут использоваться для устранения ожиданий при обменах данными на стыках блоков расчетной области (при использовании геометрического параллелизма), путем предсказания принимаемых значений переменных. Очевидно, что расчет такого экстраполятора эквивалентен решению задачи, обратной расчету однослойного линейного персептрона с N нейронами, имеющими смещение Bi (здесь требуется определить такие значения входов персептрона, при которых наблюдаются указанные значения выходов). funneled\_perceptron\_out \* outpi = new funneled\_perceptron\_out(true, false, refs, "iPFUNNEL", N, 1E-5);.

**Key words part:** 0.6071428571428571

=================================

**RUSBERT\_KMeans\_With\_ST/:** Если новые (прогнозируемые) значения в канале зависят только от предыдущих значений в некотором подмножестве элементов массива, то используются простые линейные соотношения и канал является явным . а) для решения (более корректного с точки зрения численных методов по сравнению с точечными каналами) тех же задач предварительного расчета значений в узлах расчетной сетки на стыках блоков расчетной области, которые в общем случае, несомненно, зависят и от "соседних" узлов и от значений других переменных в текущем и "соседних" узлах;. Также отметим, что подобного рода переход от расчета по исходной разностной схеме, реализованной в программе, к расчету по разностной схеме, аппроксимированной каналом, может, фактически, быть одним из вариантов автоматического распараллеливания расчета , если исходная программа не является параллельной, а канал реализует счет в параллельном режиме. funneled\_predictor\_out \* out = new funneled\_predictor\_out("FUNNEL", 1, 4, 0.01);.

**Key words part:** 0.6071428571428571

=================================

**RUSBERT\_page\_rank\_Without\_ST/:** На передающей стороне:. vector<\_funnel \*> refs;. На принимающей стороне:. vector<\_funnel \*> refs;. cout << "NO PREDICTION" << endl;.

**Key words part:** 0.32142857142857145

=================================

**RUSBERT\_page\_rank\_With\_ST/:** На передающей стороне:. vector<\_funnel \*> refs;. На принимающей стороне:. vector<\_funnel \*> refs;. cout << "NO PREDICTION" << endl;.

**Key words part:** 0.32142857142857145

=================================

**Simple\_PageRank/:** При этом возможна как точечная экстраполяция (исключительно на базе предыдущих значений текущей переменной в данной точке), так и коллективная экстраполяция "в ширину" , при которой для вычисления прогноза текущей переменной (в данной точке) используется одношаговая история группы значений в некотором срезе расчетной области, включающем данную точку;. Например, если создается локальный (соединяющий процесс с самим собой) одноточечный канал (P = 4, e = 0,01) для приема-передачи вещественных чисел, то его полная инициализация, включающая создание приемного и передающего коннекторов выглядит так:. Например, если создается явный канал (зависимый от канала inb) без свободного члена, e = 10-5, для приема-передачи N вещественных чисел, то его полная инициализация, включающая создание приемного и передающего коннекторов выглядит так:. Из данных, приведенных в таблице 1, очевидно, что применение предицирующих каналов позволило существенно увеличить ускорение счета (в сравнении с программой, реализующей стандартный подход со вспомогательными разностными схемами), в 1,4÷6,5 раз. 1. Инициализация двух неявных каналов (основного, работающего со значениями температуры T, и вспомогательного [от него зависит основной], работающего с источниковым членом S) без свободных членов, в локальном режиме (отправителем является тот же процесс, что и получатель). Из данных, приведенных в таблице 2, очевидно, что применение предицирующих каналов позволило перейти от явной разностной схемы к неявной (со стандартным распараллеленным решателем методом Гаусса-Зейделя с красно-черной раскраской) и получить определенное ускорение счета (в сравнении с однопоточной программой), в 1,1÷1,69 раза.

**Key words part:** 0.6428571428571429

=================================

**TextRank/:** В самом деле, достаточно организовать некоторое количество итераций вычислительного процесса по явной схеме (при соблюдении условий вычислительной устойчивости) с передачей получаемых данных в неявный канал коллективного решения, который, фактически, аппроксимирует по этим данным неявную схему. 2. Для апробации линейных каналов коллективного решения рассмотрим задачу перехода от явной разностной схемы к распараллеленной неявной при численном интегрировании одномерного уравнения теплопроводности с источниковым членом. д) сброс основным каналом принятых фиктивных значений Q и их замена результатом предикции T. Как показал эксперимент, данная программа генерирует результат, который с высокой степенью точности (относительная погрешность составляет около 0,001% и объясняется погрешностями решения системы линейных уравнений неявным каналом при поиске предиктора) совпадает с результатами счета контрольной программой, которая реализовывала разгонный расчет по явной схеме и основной расчет по запрограммированной неявной схеме. Из данных, приведенных в таблице 2, очевидно, что применение предицирующих каналов позволило перейти от явной разностной схемы к неявной (со стандартным распараллеленным решателем методом Гаусса-Зейделя с красно-черной раскраской) и получить определенное ускорение счета (в сравнении с однопоточной программой), в 1,1÷1,69 раза. Совмещение каналов со средствами предсказания данных упрощает программирование ряда алгоритмов, связанных с численным моделированием (прогнозирования времени счета для балансировки загрузки, расчета прогнозных значений переменных, например, на стыках блоков расчетной области при параллельной обработке, и других алгоритмов), и позволяет, в ряде случаев, осуществлять скрытые переходы от явных разностных схем к неявным, отличающихся большей устойчивостью счета, а также от последовательных алгоритмов счета к параллельным. Показано, что применение неявных каналов коллективного решения позволило (при использовании минимальных средств программирования) перейти от последовательной явной схемы счета уравнения теплопроводности к распараллеленной неявной схеме, отличающейся более высокой вычислительной устойчивостью, и получить заметное ускорение счета (в 1,1÷1,69 раза на восьмиядерной 16-потоковой вычислительной системе проекта Google’s Compute Engine).

**Key words part:** 0.8214285714285714

=================================

**TF-IDF\_KMeans/:** 1. Авторегрессионные точечные каналы , выполняющие экстраполяцию значений конкретной переменной в некоторой точке, основываясь на истории значений, поступивших ранее в канал в данной точке. Заметим, что такой же подход может использоваться и для обратного перехода – от неявной разностной схемы к явной. Обучение i-го экстраполятора (персептрона) выполняется с помощью метода наименьших квадратов, путем минимизации функционала. которая, в данном линейном случае, сводится к решению системы линейных алгебраических уравнений, возникающих из условий равенства первых частных производных нулю:. Прием в предикционном режиме (с последующей коррекцией предиктора) выполняется в таком случае кодом:. vector<\_funnel \*> refs;. Передача массива seq выполняется вызовом, подобным следующему:. Ускорение на 16 потоках (вариант c предицирующими каналами), t1/tp16.

**Key words part:** 0.7142857142857143

=================================

**Текст:** Математическое моделирование процессов в сплошных средах – важная технология, применяемая при решении разнообразных научно-технических задач в сферах аэро- и гидродинамики, акустики, электротехники, при исследовании и прогнозировании процессов в атмосфере, океанах и магме, а также в ряде иных областей науки. Достаточно часто задачи моделирования имеют существенную размерность и сложность, их решение (за разумное время) требует адекватного применения многопроцессорных вычислительных систем. Существенными особенностями большинства применяемых при этом расчетных программ являются:. а) наличие, как уже упоминалось, значительного количества расчетов по сложным математическим моделям;. б) наличие межпроцессорных обменов данными;. в) наличие неравномерностей загрузки процессоров.. Эти факторы в наибольшей степени негативно влияют на время расчета, соответственно, актуальна задача разработки подходов к их устранению или минимизации их влияния. Достаточно интересным таким подходом является предсказание данных и/или значений функциональных характеристик программы [1], которое позволяет:. а) заменить некоторые фрагменты математической модели прогнозирующими схемами достаточной точности и меньшей трудоемкости [2, 3];. б) в некоторых случаях, не дожидаясь завершения приема данных, пытаться спрогнозировать и использовать их значения, выполнив коррекцию при завершении приема [2]. При этом возможна как точечная экстраполяция (исключительно на базе предыдущих значений текущей переменной в данной точке), так и коллективная экстраполяция «в ширину» , при которой для вычисления прогноза текущей переменной (в данной точке) используется одношаговая история группы значений в некотором срезе расчетной области, включающем данную точку;. в) точечно прогнозировать текущую загрузку (предполагаемое время выполнения кода на определенном этапе исполнения) каждого процессора и выполнять на основе полученных данных простые схемы адекватной балансировки загрузки [2];. г) точечно спрогнозировать время выполнения параллельного фрагмента программы и его непараллельного аналога, что позволяет выбрать вариант с меньшим временем исполнения (известно, что для многих алгоритмов время исполнения имеет разные [часто нелинейные] зависимости от входных данных, в параллельной и непараллельной реализациях) [4].. Большинство современных технологий предсказания, используемых при разработке программ, базируется на применении различных стандартных библиотек и/или сервисов, реализующих полиномиальную регрессию, нейронные сети, деревья решений и иные технологии. Отметим библиотеку C++ Dlib [5], реализующую все упомянутые выше технологии, библиотеку «Метаслой» [1], использующую полиномиальную регрессию, алгоритмико-логические и вероятностные модели, платформу TensorFlow [6], использующую средства глубокого обучения нейронных сетей, библиотеки Facebook Prophet [7] и Apache Spark [8]. Существуют и стандартные библиотеки, входящие в состав основных средств языков программирования, например, SciPy в языке Python [9], поддерживающая регрессионные модели. Все перечисленные платформы и библиотеки требуют не только явного подключения, но и явного обращения к предоставляемым ими средствам. Это приемлемое, но не вполне удобное решение для перечисленных выше задач, возникающих в ходе численного моделирования (сам вспомогательный характер таких задач говорит о необходимости по возможности «скрыть» их от программирующего специалиста, сделав создаваемый код более простым и логичным). Кроме того, средства, предоставляемые указанными выше библиотеками, претендуют на универсальность и, как следствие: а) избыточны, б) вычислительно затратны (значительная часть трудозатрат может быть вызвана анализом данных с целью поиска оптимальных схем предикции).. Поэтому, актуальна цель данной работы – повышение простоты применения (в программировании) и надежности средств предикции с сохранением достаточного качества прогнозов при небольшой вычислительной трудоемкости. Соответственно, ставится задача разработки таких новых технологических средств предикции, которые использовали бы достаточно простые и быстрые (пусть даже и менее совершенные) алгоритмы, были бы, по возможности, «прозрачны» для программиста (например, были бы совмещены со стандартными средствами обменов данными в многопроцессорной среде), привычны с точки зрения интерфейса и гибки в применении. В частности, достаточно естественной представляется инкапсуляция средств предикции в стандартные каналы передачи данных (характерны, например, для стандартной библиотеки языка Planning C (см., например, [10]), являющегося расширением стандартного C++, а также для языков Go [11] и OCaml [12]). Схожий подход применен в работе [13], но лишь в отношении простейшей предикции некоторых служебных данных, поступающих в маршрутизатор из компьютерных сетей в ответ на запрос.. Предикционно-решающие каналы. Введем понятие предикционно-решающего канала межпроцессной или внутрипроцессной передачи данных, который может работать в трех основных режимах, определяемых конкретной вызываемой функцией приема данных и ее параметрами:. а) классический режим асинхронной передачи данных, при котором принимающая сторона в обязательном порядке дожидается данных, переданных отправляющей стороной;. б) предикционный режим, при котором принимающая сторона пытается спрогнозировать переданные ей, но еще не полученные данные, основываясь на накопленной истории их изменения в ходе ранее осуществленных приемов-передач. Если накопленная история недостаточна, то канал осуществляет прием данных в классическом режиме;. в) режим с таймаутом , при котором в принимающую функцию передается значение допустимого времени ожидания. Если данные приходят за меньшее время, чем указано, то они принимаются в классическом режиме. Если же за указанное время данные не поступили, то принимающая функция переводится в предикционный режим.. Важно заметить, что вполне допустима работа канала в случае, если передающая и принимающая стороны совпадают . При этом процесс засылает серию данных в канал и получает из него результат прогнозирования следующих значений для этой серии данных, то есть, в данном случае предикционно-решающий канал является локальным прогностером , который можно использовать по аналогии с метаслоем [1].. Во введении упоминались две наиболее употребимые схемы экстраполяции – точечная и «в ширину». Соответственно, предложим две основные разновидности предикционно-решающих каналов:. 1. Авторегрессионные точечные каналы , выполняющие экстраполяцию значений конкретной переменной в некоторой точке, основываясь на истории значений, поступивших ранее в канал в данной точке. В параллельном программировании ряда задач моделирования процессов в сплошных средах такие каналы могут использоваться для устранения ожиданий при обменах данными на стыках блоков расчетной области (при использовании геометрического параллелизма), путем предсказания принимаемых значений переменных. Возможно даже частичное периодическое исключение обменов вообще (с последующим приемом актуальных данных и коррекцией экстраполяторов канала), в таких случаях работа идет с предсказанными данными, а устранение обменов значительно уменьшает время работы программы. Обычно такая задача решается явным применением экстраполяторов или дополнительных разностных схем, поэтому применение авторегрессионных каналов существенно упрощает программу, соответственно, снижается вероятность внесения в нее ошибок.. Также возможно, например, точечное предсказание различных характеристик вычислительного процесса, например, предполагаемого времени работы численного алгоритма в конкретных узлах, для выбора варианта алгоритма или для эффективного локально статического распределения нагрузки (узлов расчетной сетки) по процессорам, что особенно актуально для задач моделирования процессов аэро- или гидродинамики, осложненных химической кинетикой, в условиях значительного пространственного варьирования температуры.. 2. Линейные каналы коллективного решения (явные и неявные) , выполняющие экстраполяцию массива значений некоторой переменной, основываясь на предыдущих значениях этой же переменной, а также, возможно, актуальных значений переменных из иных каналов (зависимость канала от иных каналов определяется при его создании). Если новые (прогнозируемые) значения в канале зависят только от предыдущих значений в некотором подмножестве элементов массива, то используются простые линейные соотношения и канал является явным . Если же новые значения в произвольном элементе массива в канале зависят также от новых значений в иных элементах массива этого же канала, то для расчета новых значений решается система линейных уравнений и канал является неявным .. Очевидно, что, при адекватном применении, каналы коллективного решения, фактически, могут выполнять аппроксимацию явных и/или неявных разностных схем, что определяет два основных варианта их использования:. а) для решения (более корректного с точки зрения численных методов по сравнению с точечными каналами) тех же задач предварительного расчета значений в узлах расчетной сетки на стыках блоков расчетной области, которые в общем случае, несомненно, зависят и от «соседних» узлов и от значений других переменных в текущем и «соседних» узлах;. б) для решения достаточно неожиданной задачи простого и эффективного перехода (в простых случаях) от явных разностных схем к неявным , что обычно требует применения специальных математических подходов (например, схем расщепления по физическим процессам или по пространственным переменным) и достаточно громоздкого программирования. В самом деле, достаточно организовать некоторое количество итераций вычислительного процесса по явной схеме (при соблюдении условий вычислительной устойчивости) с передачей получаемых данных в неявный канал коллективного решения, который, фактически, аппроксимирует по этим данным неявную схему. Далее организуется цикл отправки «пустых» данных в этот канал с приемом из него данных в строгом предикционном режиме – это, фактически, будут результаты дальнейшего эквивалентного счета по неявной разностной схеме , отличающейся заметно большей вычислительной устойчивостью. Заметим, что такой же подход может использоваться и для обратного перехода – от неявной разностной схемы к явной. Также отметим, что подобного рода переход от расчета по исходной разностной схеме, реализованной в программе, к расчету по разностной схеме, аппроксимированной каналом, может, фактически, быть одним из вариантов автоматического распараллеливания расчета , если исходная программа не является параллельной, а канал реализует счет в параллельном режиме. Необходимо лишь предусмотреть возможность масштабирования построенного каналом предиктора , путем его быстрого одношагового пересчета при изменении каких-либо параметров схемы, например, величины шага по времени. Это можно осуществить путем обучения нескольких вспомогательных каналов (при разном значении параметра), на основании данных которых основной канал подбирает, например, простые линейные зависимости коэффициентов предиктора от параметра. Так можно не только переходить от явных схем к распараллеленным неявным, но и увеличивать при этом шаг интегрирования по времени, что дает дополнительное ускорение счета.. Авторегрессионные точечные каналы. Авторегрессионный точечный канал выполняет или обычный блокирующий прием или экстраполяцию очередных значений Vi некоторой переменной в точках i = 1…N, где N – число точек. Для экстраполяции используется история значений (величины Hi,j, поступившие ранее в канал в данной точке, j = 1…P, где P – глубина истории). Экстраполяция реализуется линейным соотношением:. . где Ki,k – коэффициенты i-го экстраполятора, Si – порядок i-го экстраполятора. Очевидно, что такой экстраполятор эквивалентен линейному персептрону из одного нейрона со смещением Ki,0.. Обучение i-го экстраполятора (персептрона) выполняется с помощью метода наименьших квадратов, путем минимизации функционала. . которая, в данном линейном случае, сводится к решению системы линейных алгебраических уравнений, возникающих из условий равенства первых частных производных нулю:. . Решение системы может осуществляться, например, методом Гаусса-Зейделя, или, если он не сходится – методом LU-разложения.. Порядок экстраполятора Si обычно выбирается из условия минимума F(Ki), однако если для всех Si не выполняется условие F(Ki) ≤ e, где e -- допустимая погрешность экстраполяции, то выбирается Si = 0. Величины N, P, e задаются при инициализации канала. Например, если создается локальный (соединяющий процесс с самим собой) одноточечный канал (P = 4, e = 0,01) для приема-передачи вещественных чисел, то его полная инициализация, включающая создание приемного и передающего коннекторов выглядит так:. funneled\_predictor\_out \* out = new funneled\_predictor\_out("FUNNEL", 1, 4, 0.01);. funneled\_predictor\_in \* in = new funneled\_predictor\_in("FUNNEL", 1, 4, 0.01);. Передача элемента массива seq[i] выполняется вызовом, подобным следующему:. out->put((void \*) &seq[i], sizeof(double));. Прием в предикционном режиме (с последующей коррекцией предиктора) выполняется в таком случае кодом:. if (in->get\_timed((void \*) &seq1[i], -1.0)) { // Предикционный режим. Если заменить параметр «-1.0» положительным значением таймаута, то реализуется режим приема с таймаутом.. double buf = 0.0;. in->get\_and\_correct((void \*) &buf); // Классический режим приема с коррекцией предиктора. cout << buf << "[Предсказано = " << seq1[i] << "] ";. } else. cout << seq1[i] << " "; // Если предиктор не был вычислен (это возможно в режиме приема с таймаутом). Линейные каналы коллективного решения. Линейный канал коллективного решения выполняет или обычный блокирующий прием или экстраполяцию очередных значений Vi некоторой переменной в точках i = 1…N, где N – число точек. Для экстраполяции используются:. а) история значений искомой величины (значения Hi,j, поступившие ранее в канал, j = 1…P, где P – глубина истории);. б) история значений величин, поступивших в каналы, связанные с текущим (значения Ri,j,m, m = 1…M, где M – число каналов, связанных с текущим).. Как уже упоминалось выше, канал коллективного решения может быть явным или неявным. Кроме того, он может (по желанию исследователя-программиста) иметь или не иметь свободный член Bi.. Экстраполяция в явном канале со свободным членом Bi реализуется линейным соотношением:. . где Ai,k и Ci,m – коэффициенты i-го экстраполятора. Очевидно, что расчет такого экстраполятора эквивалентен расчету однослойного линейного персептрона с N нейронами, имеющими смещение Bi.. Необходимо сразу отметить, что в матрице C ненулевыми являются только элементы, соответствующие значимым (для расчета в каждой i-й точке) связанным каналам. Значимость выявляется обычным корреляционным анализом (связанный канал k значим для расчета в i-й точке, если модуль коэффициента корреляции между Ri,j,k и Hi,j превышает некоторую наперед заданную величину D).. Обучение i-го явного экстраполятора (персептрона) выполняется с помощью метода наименьших квадратов, путем минимизации функционала. . которая сводится к решению системы линейных алгебраических уравнений, возникающих из условий равенства первых частных производных по неизвестным коэффициентам нулю.. Экстраполяция в неявном канале со свободным членом Bi реализуется решением системы из N линейных уравнений относительно V:. . где A и C – матрицы коэффициентов коллективного экстраполятора. Очевидно, что расчет такого экстраполятора эквивалентен решению задачи, обратной расчету однослойного линейного персептрона с N нейронами, имеющими смещение Bi (здесь требуется определить такие значения входов персептрона, при которых наблюдаются указанные значения выходов).. Обучение неявного экстраполятора (персептрона) выполняется последовательно для каждого из значений i, с помощью метода наименьших квадратов, путем минимизации функционала. . которая сводится к решению системы линейных алгебраических уравнений, возникающих из условий равенства первых частных производных по неизвестным коэффициентам нулю.. Экстраполяция в каналах без свободного члена также реализуется линейными соотношениями при Bi = 0. Поиск соответствующих экстраполяторов производится по тому же принципу, что и для каналов со свободным членом.. Далее заметим, что независимо от явности/неявности канала, количество ненулевых членов p в Ai обычно выбирается из условия минимума F(Ai,Bi,Ci), однако если для всех p не выполняется условие F(Ai,Bi,Ci) ≤ e, где e -- допустимая погрешность экстраполяции, то выбирается p = 0, в этом случае в расчете участвуют только Bi и Ci.. Величины N, e, признаки «явный/неявный» и «со свободным членом/без него» задаются при инициализации канала. Например, если создается явный канал (зависимый от канала inb) без свободного члена, e = 10-5, для приема-передачи N вещественных чисел, то его полная инициализация, включающая создание приемного и передающего коннекторов выглядит так:. На передающей стороне:. vector<\_funnel \*> refs;. refs.push\_back(inb->getRef());. funneled\_perceptron\_out \* outpi = new funneled\_perceptron\_out(true, false, refs, "iPFUNNEL", N, 1E-5);. На принимающей стороне:. vector<\_funnel \*> refs;. funneled\_perceptron\_in \* inpi = new funneled\_perceptron\_in(true, false, refs, "iPFUNNEL", N, 1E-5);. Передача массива seq выполняется вызовом, подобным следующему:. outpi->put((void \*) seq, N\*sizeof(double));. Прием в предикционном режиме с последующей коррекцией предиктора выполняется по той же схеме, что и для точечного канала (см. выше). При этом предполагается, что передача-прием по связанному каналу inb уже произведены. Прием в предикционном режиме (в массив seqpp) без коррекции предиктора выполняется по схеме:. if (inpi->get\_timed((void \*) seqpp, -1.0)) { // Предикционный режим. Если заменить параметр «-1.0» положительным значением таймаута, то реализуется режим приема с таймаутом.. inpi->cancel\_and\_push((void \*) seqpp); // Классический режим приема – принятое значение сбрасывается, вместо него в историю помещаются значения, только что вычисленные предиктором. } else. cout << "NO PREDICTION!" << endl;. Апробация. 1. Для апробации точечных каналов решим задачу численного интегрирования одномерного уравнения теплопроводности , которое часто фигурирует в различных задачах математического моделирования. Пусть данное уравнение решается в параллельном варианте (геометрический параллелизм) с применением неявной разностной схемы, что приводит к необходимости использования метода прямой и обратной прогонки. Попытка непосредственного переноса метода прогонки на параллельную систему ускорения счета не дает (процессоры "выстраиваются" сначала в прямую, а затем в обратную цепочку передач), поэтому, чтобы обеспечить относительную независимость обработки разных блоков расчетной сетки различными процессорами, используются вспомогательные разностные схемы, дающие прогнозный результат на стыках блоков [2], с последующими локальными прогонками. В нашем случае, для получения прогнозных результатов применены авторегрессионные точечные каналы , что сразу же упростило расчетную программу.. Как показал расчет по составленной программе, применение авторегрессионных каналов с P = 3 и e = 10‑10, позволило получить решение, погрешность которого близка к нулю. Исследуем теперь ускорение счета (на восьмиядерной 16-потоковой вычислительной системе проекта Google’s Compute Engine) в зависимости от размерности задачи, при решении как программой, использующей предицирующие каналы, так и стандартной программой, не использующей таковых. Результаты замеров времени и рассчитанные ускорения сведены в таблицу 1.. Таблица 1. Результаты замеров времени и расчетов ускорения (апробация точечных каналов). Число узлов расчетной сетки. 2000. 3000. 5000. 7000. Время счета на одном потоке t1, с. 1,106. 1,658. 2,769. 3,86. Время счета на 16 потоках (вариант без предицирующих каналов), ts16, с. 0,742. 0,474. 0,746. 0,397. Ускорение на 16 потоках (вариант без предицирующих каналов), t1/ts16. 1,49. 3,48. 3,71. 9,72. Время счета на 16 потоках (вариант c предицирующими каналами), tp16, с. 0,114. 0,133. 0,272. 0,284. Ускорение на 16 потоках (вариант c предицирующими каналами), t1/tp16. 9,7. 12,47. 10,18. 13,6. Из данных, приведенных в таблице 1, очевидно, что применение предицирующих каналов позволило существенно увеличить ускорение счета (в сравнении с программой, реализующей стандартный подход со вспомогательными разностными схемами), в 1,4÷6,5 раз. Этот эффект особенно заметен при относительно небольших размерностях задачи, когда временные затраты на применение дополнительных приемов, связанных с распараллеливанием, особенно критичны. Видимо, в случае предицирующих каналов, эти затраты меньше.. 2. Для апробации линейных каналов коллективного решения рассмотрим задачу перехода от явной разностной схемы к распараллеленной неявной при численном интегрировании одномерного уравнения теплопроводности с источниковым членом. Пусть программа состоит из следующих блоков:. 1. Инициализация двух неявных каналов (основного, работающего со значениями температуры T, и вспомогательного [от него зависит основной], работающего с источниковым членом S) без свободных членов, в локальном режиме (отправителем является тот же процесс, что и получатель).. 2. Выполнение «разгоночного» интегрирования уравнения теплопроводности по явной схеме с отправкой/получением T и S, соответственно, в каналы/из каналов в предикционном режиме с последующей коррекцией предиктора – таким путем обучается, формируя неявную разностную схему, предиктор неявного канала.. 3. Выполнение цикла, на каждой итерации которого производятся:. а) вычисление и отправка во вспомогательный канал величин S;. б) засылка в основной канал фиктивных данных Q (важен лишь синхронизирующий факт передачи);. в) прием из вспомогательного канала значений S;. г) прием T из основного канала в предикционном режиме (при этом срабатывает неявная схема, сформированная предиктором, генерирующая актуальные значения T);. д) сброс основным каналом принятых фиктивных значений Q и их замена результатом предикции T.. Как показал эксперимент, данная программа генерирует результат, который с высокой степенью точности (относительная погрешность составляет около 0,001% и объясняется погрешностями решения системы линейных уравнений неявным каналом при поиске предиктора) совпадает с результатами счета контрольной программой, которая реализовывала разгонный расчет по явной схеме и основной расчет по запрограммированной неявной схеме. Таким образом, потери точности не произошло.. Рассмотрим ускорение счета (на восьмиядерной 16-потоковой вычислительной системе проекта Google’s Compute Engine) в зависимости от размерности задачи. Результаты замеров времени и рассчитанные ускорения сведены в таблицу 2.. Таблица 2. Результаты замеров времени и расчетов ускорения (апробация линейных каналов). Число узлов расчетной сетки. 500. 1000. 2000. 3000. Время счета контрольной программой (в однопоточном варианте с разгонным расчетом по явной схеме и основным расчетом по запрограммированной неявной схеме), t1, с. 0,699. 3,5. 20,18. 47,98. Время счета программой с разгонным расчетом по явной схеме и основным расчетом на предицирующих линейных каналах (с распараллеленной неявной схемой, выведенной каналом), t16, с. 0,698. 3,095. 11,923. 33,35. Ускорение счета, t1/t16. 1. 1,131. 1,69. 1,44. Из данных, приведенных в таблице 2, очевидно, что применение предицирующих каналов позволило перейти от явной разностной схемы к неявной (со стандартным распараллеленным решателем методом Гаусса-Зейделя с красно-черной раскраской) и получить определенное ускорение счета (в сравнении с однопоточной программой), в 1,1÷1,69 раза. Видимо, в данном случае достаточно существенными оказались дополнительные временные затраты на обслуживание предицирующих каналов – это объясняет тот факт, что ускорение счета оказалось меньше теоретического значения 12÷16 (ожидаемого для системы с 8 ядрами с технологией HyperThreading). Эти затраты обычно нивелируются с ростом размерности задачи, поэтому с ростом размерности ускорение счета в большинстве случаев растет (см. таблицу 2). Повысить ускорение, вероятно, можно путем оптимизации алгоритмов обслуживания предицирующих каналов, например, с помощью специальной обработки типовых частных случаев.. Выводы. В данной работе впервые предложен новый вид технологических средств предикции для использовании в параллельном программировании – предикционно-решающие каналы. Предложены два вида каналов – авторегрессионные точечные и линейные (явные или неявные) коллективного решения. Изложены математические аспекты предикции в таких каналах, кратко описаны базовые средства программирования.. Совмещение каналов со средствами предсказания данных упрощает программирование ряда алгоритмов, связанных с численным моделированием (прогнозирования времени счета для балансировки загрузки, расчета прогнозных значений переменных, например, на стыках блоков расчетной области при параллельной обработке, и других алгоритмов), и позволяет, в ряде случаев, осуществлять скрытые переходы от явных разностных схем к неявным, отличающихся большей устойчивостью счета, а также от последовательных алгоритмов счета к параллельным.. Предложенные подходы и средства программирования апробированы при решении задач численного моделирования на базе нестационарного уравнения теплопроводности. Показано существенное повышение ускорения (в 1,4÷6,5 раза на восьмиядерной 16-потоковой вычислительной системе проекта Google’s Compute Engine) для параллельной моделирующей программы при переходе от классической схемы с обычными каналами и предвычислением с помощью дополнительной разностной схемы, к схеме с применением авторегрессионных точечных каналов. Показано, что применение неявных каналов коллективного решения позволило (при использовании минимальных средств программирования) перейти от последовательной явной схемы счета уравнения теплопроводности к распараллеленной неявной схеме, отличающейся более высокой вычислительной устойчивостью, и получить заметное ускорение счета (в 1,1÷1,69 раза на восьмиядерной 16-потоковой вычислительной системе проекта Google’s Compute Engine). Такой переход может иметь определенную ценность при разработке программ с нестандартными разностными схемами, для которых не существует стандартных, хорошо распараллеленных математических библиотек.. Полученные результаты подтверждают достоверность, обоснованность и эффективность предложенных в данной работе решений.