Л5. Вычисление частных сумм последовательности числовых значений

1. Гергель В. П. Теория и практика параллельных вычислений. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория Базовых Знаний, 2007. – 427 с.

1. Модельный пример для анализа проблем, возникающих при разработке параллельных методов вычислений

Постановка задачи нахождения частных сумм последовательности чисел (prefix sum problem):

$$S_k = \sum_{i=1}^{k} x_i, \quad 1 \le k \le n,$$

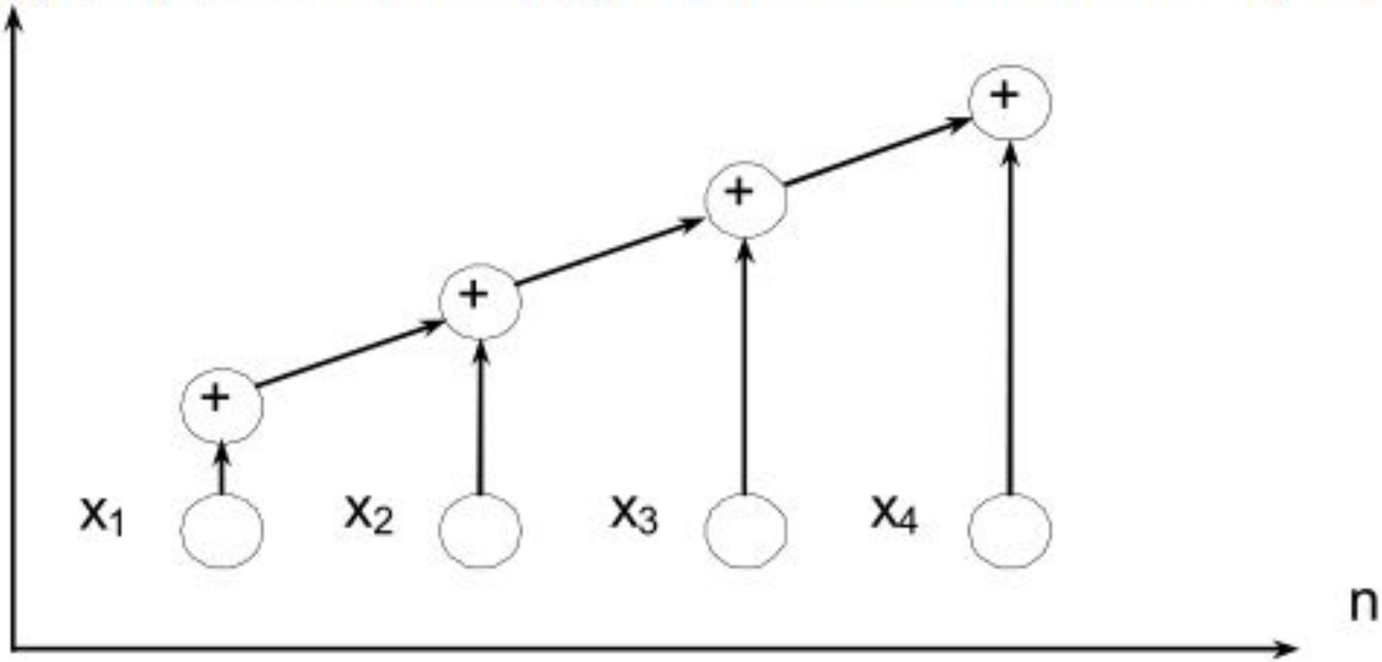
где n есть количество суммируемых значений.

Изучение возможных параллельных методов решения задачи начнем с еще более простого варианта ее постановки – с задачи вычисления общей суммы (в таком виде задача суммирования является частным случаем общей задачи редукции):

$$S = \sum_{i=1}^{n} x_i .$$

2. Вычислительная схема алгоритм суммирования

Последовательный алгоритм вычисления общей суммы состоит в выполнении следующих действий: $S=0$; $S+=x_1$, …, $S+=x_n$. Эти действия представляются $G_1 = (V_1, R_1)$,
где $V_1=\{v_{01}, \dots, v_{0n}, v_{11}, \dots, v_{1n}\}$ - есть множество операций (вершины $v_{01}, \dots, v_{0n}$ обозначают операции ввода, каждая вершина v_{1i}, $1 \le i \le n$ соответствуют операции суммирования S и x_i, а R1={(v0i, v1i), (v1i, vii+1), 1≤ i ≤n-1} - множество дуг, определяющих информационные зависимости операций.



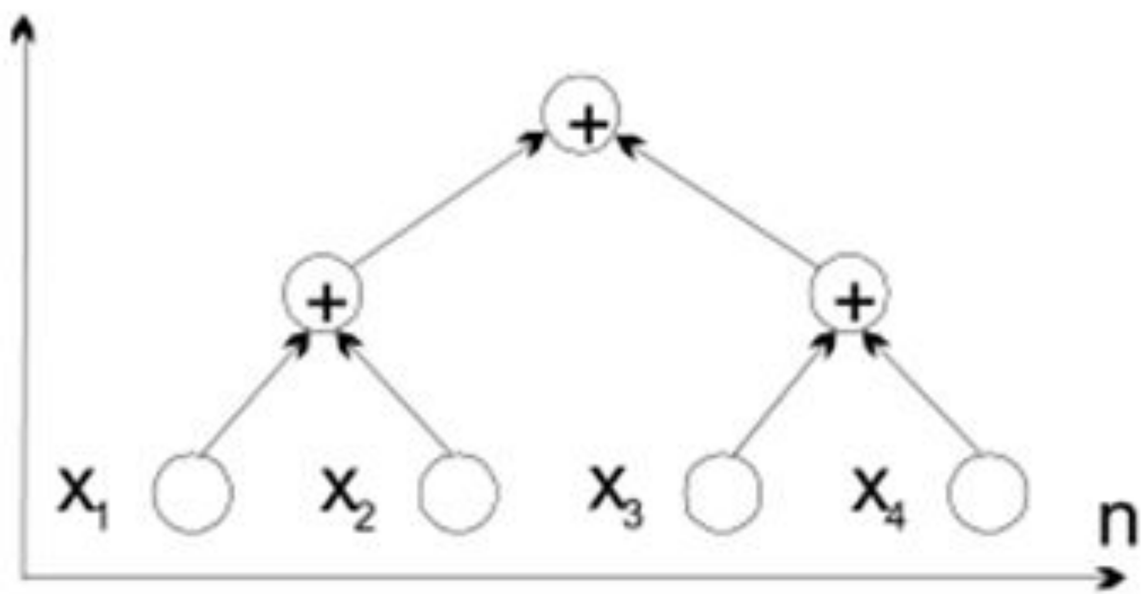
3. Каскадная схема суммирования

Пусть $n = 2^k$. Вычислительная схема определяется графом $\boldsymbol{G}_2 = (\boldsymbol{V}_2, \boldsymbol{R}_2)$,

$$V_2 = \{(v_{i1},\ldots,v_{il_i}),\quad 0 \le i \le k,\quad 1 \le l_i \le 2^{-i} n\}$$

, где $(v_{01}, \ldots, v_{0n})$ – операции ввода, $(v_{11}, \ldots, v_{1n/2})$ – операции первой итерации, а множество дуг графа определяется соотношениями:

$$R_2 = \{(v_{i-1,2j-1} v_{ij}), (v_{i-1,2j} v_{ij}),\quad 1 \le i \le k,\quad 1 \le j \le 2^{-i} n\}.$$



4. Анализ каскадной схемы

Количество итераций каскадной схемы равно $k=\log_2 n$, а общее количество операций суммирования $K_{посл}=n-1$. При параллельном выполнении каскадной схемы общее количество параллельных операций равно $K_{пар}= k=\log_2 n$. Тогда $T1=K_{посл}$, $Tp= K_{пар}$ и показатели ускорения и эффективности каскадной схемы алгоритма суммирования оцениваются:

$$S_p = T_1 / T_p = (n-1) / \log_2 n,$$

$$E_p = T_1 / pT_p = (n-1) / (p \log_2 n) = (n-1) / ((n/2) \log_2 n),$$

где $p=n/2$ - необходимое для выполнения каскадной схемы количество процессоров.

5. Оценка эффективности каскадной схемы

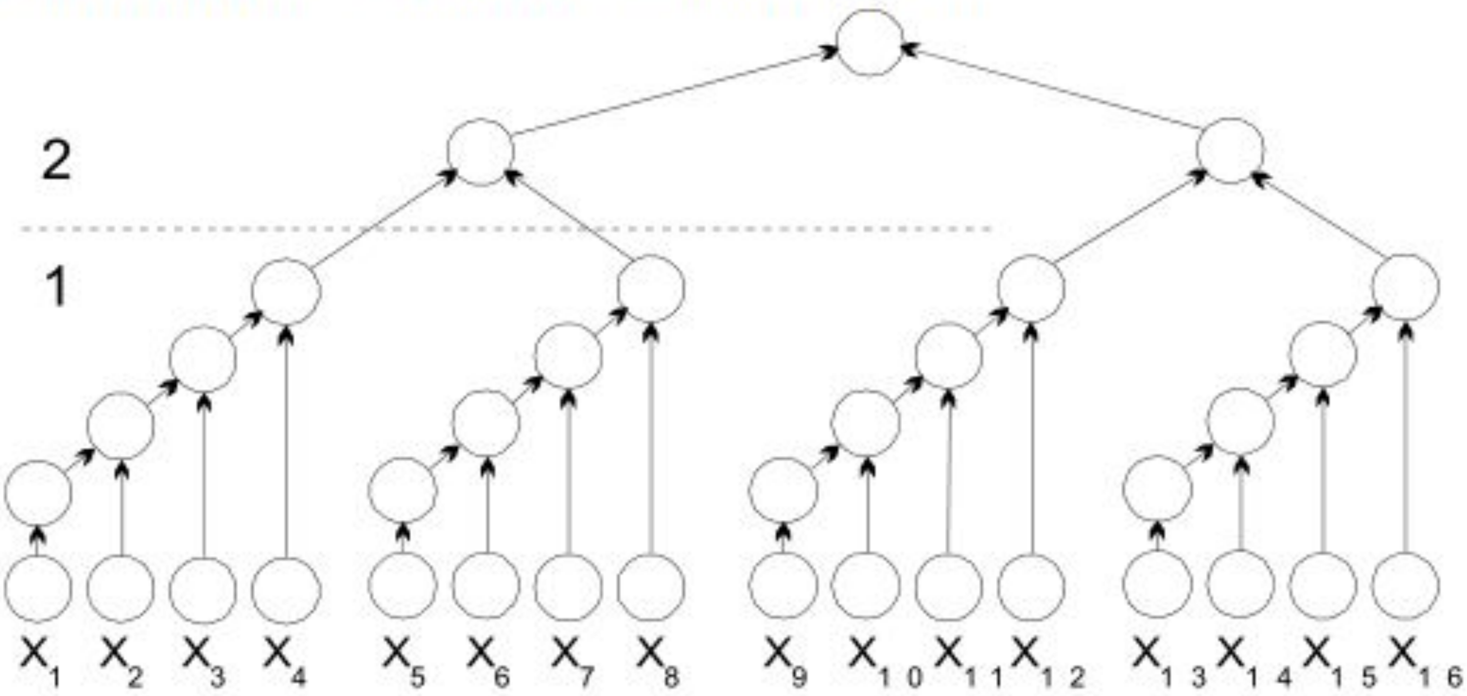
Анализируя полученные характеристики, можно отметить, что время параллельного выполнения каскадной схемы совпадает с оценкой для паракомпьютера в теореме 2. Однако при этом эффективность использования процессоров уменьшается при увеличении количества суммируемых значений:

$$\lim E_p \to 0 \quad при \quad n \to \infty .$$

6. Модифицированная каскадная схема

Пусть $n=2^k$, $k=2^s$. Все вычисления подразделяются на два последовательно выполняемых этапа суммирования:

1. На первом этапе вычислений все значения подразделяются на $(n/\log_2 n)$ групп, в каждой из которых содержится $\log_2 n$ элементов; далее для каждой группы вычисляется сумма при помощи последовательного алгоритма; вычисления в каждой группе выполняются независимо друг от друга (параллельно), и для этого необходимо наличие не менее $(n/\log_2 n)$ процессоров); 2. На втором этапе для полученных $(n/\log_2 n)$ сумм применяется обычная каскадная схема.



7. Анализ модифицированной каскадной схемы

На первом этапе для выполнения $\log_2 n$ параллельных операций требуется $p_1 = (n/\log_2 n)$ процессоров. На втором этапе для выполнения $\log_2 (n/\log_2 n) \leq \log_2 n$ параллельных операций требуется $p_2 = (n/\log_2 n)/2$ процессоров. Т. о. временная сложность равна: $Tp=2\log_2 n$ при $p=(n/\log_2 n)$. Тогда
$Sp=T1/Tp=(n-1)/(2\log_2 n)$,
$Ep= T1/(pTp)=(n-1)/((n/\log_2 n)(2\log_2 n))= (n-1)/(2n)$
По сравнению с обычной каскадной схемы ускорение для данного параллельного алгоритма уменьшилось в 2 раза, однако для его эффективности можно получить асимптотически ненулевую оценку снизу:

$\lim E_p \to 0.5 \quad при \quad n \to \infty$. Данные значения показателей достигаются при количестве процессоров, определенном в теореме 5. В отличие от обычной каскадной схемы, модифицированный каскадный алгоритм является стоимостно-оптимальным, поскольку его стоимость пропорциональна времени выполнения последовательного алгоритма:

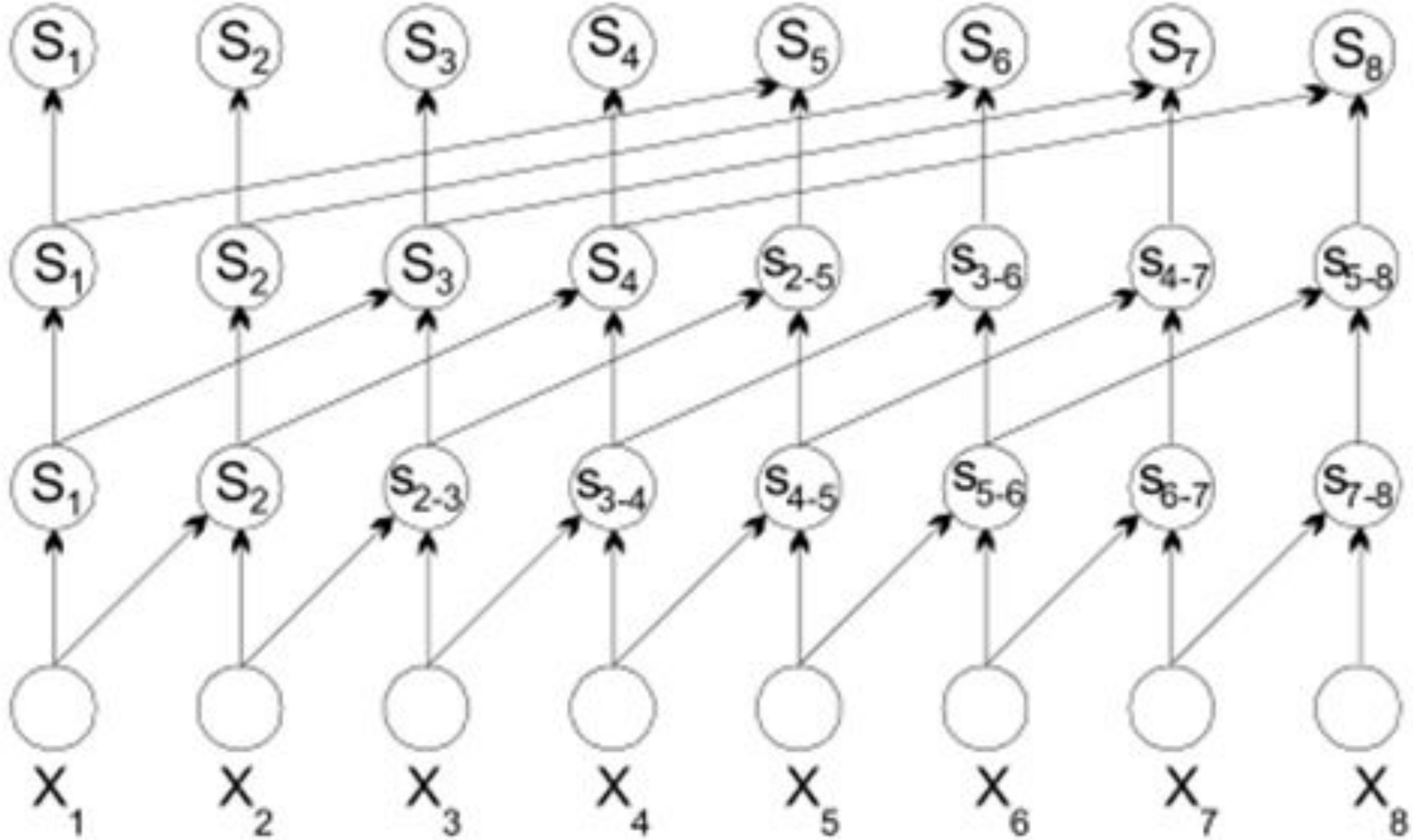
$$C_p = pT_p = (n/\log_2 n)(2\log_2 n)$$

8. Вычисление всех частных сумм

Вычисление всех частных сумм на скалярном компьютере может быть получено при помощи обычного последовательного алгоритма суммирования при том же количестве операций, что и при вычислении одной полной суммы: T1=n. При параллельном исполнении каскадной схемы в явном виде не приводит к желаемым результатам; достижение эффективного распараллеливания требует привлечения новых подходов для разработки новых параллельно-ориентированных алгоритмов решения задач. Для задачи нахождения всех частных сумм ниже рассматривается алгоритм, обеспечивающий получение результатов за $\log_2 n$ параллельных операций [4]. Перед началом вычислений создается копия S вектора суммируемых значений: S=x. На каждой итерации суммирования i, $1 \le i \le \log_2 n$ формируется вспомогательный вектор Q путем сдвига вправо вектора S на 2^{i-1} позиций (освобождающиеся при сдвиге позиции слева устанавливаются в нулевые значения); итерация алгоритма завершается параллельной операцией суммирования векторов S и Q.

9. Вычисление всех частных сумм (продолжение)

$S_{i\text{-}j}$ означают суммы значений от i до j элементов числовой последовательности



Параллельный алгоритм выполняется за $\log_2 n$ параллельных операций сложения. На каждой итерации параллельно выполняются n скалярных сложений, общее количество скалярных операций равно $K_{пар} = n\log_2 n > K_{посл} = n$.

10. Анализ алгоритма вычисления всех частных сумм

Необходимое количество процессоров определяется количеством суммируемых значений: p=n. Показатели ускорения и эффективности параллельного алгоритма вычисления всех частных сумм оцениваются следующим образом:

$$S_p = T_1 / T_p = n / \log_2 n,$$

$$E_p = T_1 / pT_p = n /(p \log_2 n) = n /(n \log_2 n) = 1 / \log_2 n\,.$$

Эффективность алгоритма также уменьшается при увеличении числа суммируемых значений, при необходимости повышения эффективности потребуется модификация алгоритма, например, по схеме модификации обычной каскадной схемы.

11. Оценка максимально достижимого параллелизма

Оценка качества параллельных вычислений предполагает знание наилучших (максимально достижимых) значений показателей ускорения и эффективности. Получение идеальных величин $S_p=p$ для ускорения и $E_p=1$ для эффективности может быть обеспечено не для всех вычислительно трудоемких задач. Так, минимально достижимое время параллельного вычисления частных сумм числовых значений составляет $\log_2 n$. Существует ряд закономерностей, которые могут быть чрезвычайно полезны при построении оценок максимально достижимого параллелизма.