

АЛГОРИТМЫ ПЕРВОНАЧАЛЬНОГО РАЗБИЕНИЯ ПОЛИГОНА НА ПЛОЩАДИ ДЛЯ СИСТЕМЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ КОЛЛЕКТИВНОГО ПОВЕДЕНИЯ РОБОТОВ НА КЛАСТЕРЕ

Вендин А.С.

*Национальный исследовательский университет «Высшая школа экономики»,
г.Москва*

Ключевые слова: имитационное моделирование, кластеры, алгоритм разбиения на равные площади, алгоритм разбиения без остатка, алгоритм разбиения с остатком, агенты.

Аннотация. В статье проводится обзор алгоритмов первоначального распределения агентов по вычислительным узлам кластера. Данная работа является частью задачи проектирования системы имитационного моделирования коллективного поведения роботов на кластере. Приводятся описания алгоритма разбиения на равные площади, алгоритмы разбиения с остатком и без остатка с оценкой сложности алгоритмов.

ALGORITHMS FOR ORIGINAL DIVISION OF THE POLYGON IN THE AREA FOR THE SOLUTION OF THE PROBLEM OF CONSTRUCTING A SIMULATIVE MODELING SYSTEM FOR COLLECTIVE BEHAVIOR ROBOTS ON THE CLUSTER

Vendin A.S.

National research University «Higher school of Economics», Moscow

Keywords: simulation modeling, clusters, algorithm for splitting into equal areas, algorithm for splitting without remainder, algorithm for splitting with remainder, agents.

Abstract. This article reviews the algorithms for the initial distribution of agents among the computing nodes of a cluster. This work is part of the task of designing a system for simulating the collective behavior of robots on a cluster. Descriptions of the algorithm for splitting into equal areas, algorithms for splitting with remainder and without remainder are given with an estimate of the complexity of the algorithms.

Введение

При разработке и проектировании различных роботизированных технических систем на современном этапе актуальной становится задача управления взаимодействием коллективом роботов, выполняющих отдельные задачи общей целевой функции, или выполнение отдельных задач с учетом граничного взаимодействия и других условий определяемых технической системой. Применение коллектива роботов [1] открывает огромные перспективы в таких областях, как робототехника и искусственный интеллект. Имитационное моделирование упрощает проектирование таких систем. Трудность проектирования таких систем заключается в сложности описания таких систем в аналитическом виде.

Проектируемая система имитационного должна уметь проводить моделирование с помощью вычислительных кластеров [2]. Система имитационного моделирования разбивается на модули, т.к. моделирование производится в кластерной системе, необходимо предусмотреть распределение модели по узлам кластера. Агент – это моделируемый элемент системы. Полигон

– пространство, описываемое декартовой системой координат, на котором расположены агенты.

Ниже рассматривается задача разбиения агентов по вычислительным узлам кластера с учетом топологии. В качестве подходов к распределению модели по узлам кластера предлагается 2 основных подхода.

1. С выбором центрального (опорного) агента.

2. Без выбора центрального (опорного) агента.

В подходах без опорных агентов происходит геометрическое разбиение модели. Далее в работе рассматриваются алгоритмы без выбора центрального (опорного) агента.

Общая постановка задачи исследований

Основная задача исследований состоит в том, что число агентов и число узлов кластера может различаться. Поэтому при проведении моделирования необходимо распределить агентов на узлы кластера, чтобы каждый узел был загружен для работы. Для решения поставленной задачи определим основные используемые в ней параметры:

- n – число узлов кластера;
- m – число агентов;
- $nodes[1..n]$ – массив узлов кластера;
- $animates[1..m]$ – массив, в котором содержатся все агенты.

$m \gg n$ – предполагается, что число агентов значительно превосходит число имеющихся узлов кластера.

Одними из возможных решений поставленной задачи могут быть использованы следующие алгоритмы разбиения агентов по вычислительным узлам кластера: алгоритм разбиения на равные площади; алгоритм разбиения без остатка; алгоритм разбиения с остатком.

1. Алгоритм разбиения на равные площади

Полигон можно попытаться разбить на приблизительно равные площади в зависимости от узлов кластера и назначить получившиеся площади соответствующим узлам. Данный алгоритм нельзя распространить на все случаи. Проведем формализацию алгоритма, выделив основные параметры и переменные.

Формализация алгоритма. Основные параметры:

a – длина исходного прямоугольника;

b – ширина исходного прямоугольника;

N – число узлов кластера;

S – площадь полигона;

K – средняя обрабатываемая площадь на узел кластера - S/N ;

$clusterNodes[1..N]$ – массив узлов кластера.

MIN_K – минимальная средняя площадь, обрабатываемая узлом кластера.

Переменные:

$currentPoint(x, y)$ – точка, хранящая текущие координаты при сканировании.

Замечания

Отношение K определяет, какую в среднем площадь будет обрабатывать каждый узел. Если $MIN_K = 1$, то максимальное число узлов кластера, которое можно назначить на полигон, рассчитывается как $a * b$.

Работа алгоритма, в случае, когда число узлов кластера превосходит размер полигона

Рассмотрим алгоритм, когда число узлов кластера много больше, чем площадь разбиваемого полигона $N \gg S$. В данном случае каждому узлу кластера назначается соответствующая площадь (рис. 1).

В цикле по всем узлам кластера создается полигоны. Для каждого полигона задаются начальные координаты, размеры по координатным осям X и Y . Полигон связывается с соответствующим узлом кластера. К текущей координате по X прибавляется единица и проверяется, не достигнута ли граница справа по координате X при смещении слева на право. Правой границей является параметр a – длина исходного полигона. Если текущая координата сравнялась с a , то происходит установка в 0 координаты по X , координата по Y увеличивается на 1. Если при смещении по оси X правая координата не достигнута, то координата по X увеличивается на 1. Происходит переход на следующую итерацию цикла.

Сложность алгоритма – $O(N)$, где N – число узлов кластера.

{Входные параметры: a – длина исходного прямоугольника; b – ширина исходного прямоугольника; N – число узлов кластера; MIN_K – минимальная средняя площадь, обрабатываемая узлом кластера.}

{Выходные параметры: $clusterNodes[1..N]$ – массив узлов кластера, содержащий координаты обрабатываемых полигонов}

AVERAGE-SQUARE-ALG(a, b, N, MIN_K)

for $i \leftarrow 0$ to N

 Создатьполигон - rectangle

 rectangle.beginPoint.x $\leftarrow$ currentPoint.X

 rectangle.beginPoint.y $\leftarrow$ currentPoint.Y

 rectangle.A $\leftarrow$ 1

 rectangle.B $\leftarrow$ 1

 clusterNodes[i] $\leftarrow$ rectangle

 if currentPoint.X + 1 = a

 currentPoint.X $\leftarrow$ 0

 currentPoint.Y $\leftarrow$ currentPoint.Y + 1

 else

 currentPoint.X $\leftarrow$ currentPoint.X + 1

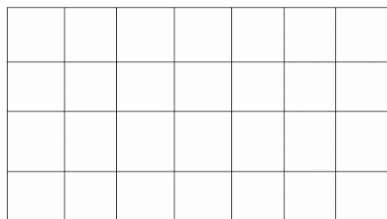


Рис. 1. Полигон, разбитый на элементарные квадраты с размерами 1x1

Далее рассматривается случай, когда число узлов кластера не превосходит размеров рабочего полигона - $N < S$. В данном случае производится расчет параметра K - средней обрабатываемой площади на узел кластера. Параметр K при делении можно получить как с остатком, так и без. Рассмотрим случай, когда K – целое число без остатка, полученное в результате деления S/N (рис. 2).

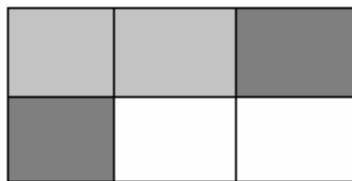


Рис. 2. Пример разбиения полигона размером 3x2 на 3 узла кластера построчным сканированием

2. Алгоритм разбиения без остатка

В цикле по всем узлам кластера инициализируется каждый узел. Для узла инициализируется полигон. Задаются начальные координаты полигона и его размеры по осям X и Y . Затем инициализируется цикл по K – размеру средней рассчитанной площади. Размер полигона увеличивается на 1, координата текущей сканирующей точки смещается по координате X . Правой границей является параметр a – длина исходного полигона. Если текущая координата сравнялась с a , то происходит установка в 0 координаты по X , координата по Y увеличивается на 1. Если при смещении по оси X правая координата не достигнута, то координата по X увеличивается на 1. В случае, если средняя площадь больше длины полигона или средняя площадь меньше и распределение площади на узел кластера ещё не заполнено, то создается дополнительный полигон, длина полигона устанавливается в 0, ширина в 1. Задаются начальные координаты полигона на основе текущей сканирующей координаты.

Сложность алгоритм – $O(ab)$. Время работы алгоритма зависит от размеров рабочего полигона, а не от числа узлов кластера.

AVERAGE-SQUARE-ALG2(a, b, N, MIN_K)

for $i \leftarrow 0$ to N

 Создать узел кластера

 Создать полигон - rectangle

 rectangle.beginPoint.x $\leftarrow$ currentPoint.X

 rectangle.beginPoint.y $\leftarrow$ currentPoint.Y

 rectangle.A $\leftarrow$ 0

 rectangle.B $\leftarrow$ 1

 clusterNodes[i] $\leftarrow$ rectangle

 for $j \leftarrow 1$ to K

 currentPoint.X $\leftarrow$ currentPoint.X + 1

 rectangle.A $\leftarrow$ rectangle.A + 1

 if currentPoint.X = a

 currentPoint.X $\leftarrow$ 0

 currentPoint.Y $\leftarrow$ currentPoint.Y + 1

 if $k > a$ OR [$i < a$ AND $k < a$]

 Создать полигон – rectangle

 Связать полигон с узлом кластера

 rectangle.beginPoint.x $\leftarrow$ currentPoint.X

 rectangle.beginPoint.y $\leftarrow$ currentPoint.Y

 rectangle.A $\leftarrow$ 0

 rectangle.B $\leftarrow$ 1

3. Алгоритм разбиения с остатком

Алгоритм аналогичен алгоритму разбиения без остатка за одним исключением. На последней итерации цикла, когда происходит связывание оставшегося пространства полигона с вычислительным узлом кластера, в узел кластера заполняется средней площадью и оставшейся частью. Добавляется проверка выхода за границы полигона по оси Y.

Сложность алгоритма – **O(ab)**. Время работы алгоритма зависит от размеров рабочего полигона, а не от числа узлов кластера.

AVERAGE-SQUARE-ALG3(a, b, N, MIN_K)

for i ← 0 to N

 Создать узел кластера

 Создать полигон - rectangle

 rectangle.beginPoint.x ← currentPoint.X

 rectangle.beginPoint.y ← currentPoint.Y

 rectangle.A ← 0

 rectangle.B ← 1

 clusterNodes[i] ← rectangle

 averageSquare ← K

 if i = N

 averageSquare ← ab-KN

 for j ← 1 to averageSquare

 currentPoint.X ← currentPoint.X + 1

 rectangle.A ← rectangle.A + 1

 if currentPoint.X = a

 currentPoint.X ← 0

 currentPoint.Y ← currentPoint.Y + 1

 if currentPoint.Y = b

 break

 if averageSquare > a

OR

[i < a AND averageSquare < a]

 Создать полигон – rectangle

 Связать полигон с узлом кластера

 rectangle.beginPoint.x ← currentPoint.X

 rectangle.beginPoint.y ← currentPoint.Y

 rectangle.A ← 0

 rectangle.B ← 1

Табл. 1. Оценка сложности работы приведенных алгоритмов

Название алгоритма	Критерий	Сложность алгоритма
Алгоритм разбиения на равные площади	Равенство площадей	O(n)
Алгоритм разбиения на равные площади	Равенство площадей, средняя площадь – целое число	O(ab)
Алгоритм разбиения на равные площади	Равенство площадей в среднем	O(ab)

Выводы

Для разбиения агентов по вычислительным узлам кластера могут быть использованы алгоритм разбиения на равные площади, алгоритм разбиения без остатка и алгоритм разбиения с остатком.

Преимуществами алгоритма разбиения на равные площади является простота реализации. Сложность алгоритма – линейная.

Использование алгоритма разбиения без остатка позволяет распределять исследуемую модель по узлам кластера. Сложность алгоритма зависит от размеров модели, а не от исходного числа кластеров. Данный алгоритм сложнее в реализации по сравнению с алгоритмом разбиения на равные площади.

Применение алгоритма разбиения с остатком является более универсальным алгоритмом. Вычислительная сложность алгоритма равна вычислительной сложности алгоритма разбиения без остатка. Данный алгоритм является более универсальным в использовании.

Список литературы

1. Вендин А.С. Исследование и разработка архитектуры имитационной системы моделирования коллективного поведения роботов на кластере // Межвузовская научно-техническая конференция студентов, аспирантов и молодых специалистов им. Е.В. Арменского. – М., 2018. – С. 8-9.
2. Вендин А.С. Задача создания системы имитационного моделирования на кластере // Сборник статей по материалам XV-XVI международной научно-практической конференции «Технические науки: проблемы и решения». – М., 2018. – С. 20-23.

Сведения об авторе:

Вендин Александр Сергеевич – аспирант НИУ ВШЭ, г. Москва.