Современные способы повышения производительности выполнения программ при обработке данных связаны с использованием вычислительных кластеров и распределенных вычислительных систем – GRID-систем. Повышение производительности выполнения программ связано с их параллельной реализацией в составе кластера, либо с распределенным выполнением в GRID-системах.Одним из возможных способов повышения производительности выполнения программ обработки данных в составе кластера является их (программ) конвейеризация. Т.е. обработка данных соответствующих им программами осуществляется в составе программного макроконвейера. Конвейеризированное выполнение программ (обработка данных) соответствует классу программных систем МПОД [1] (много программ– одни данные), что предполагает организацию обработки одного потока данных последовательностью программ (последовательностью фрагментов программ) – реализацию программного конвейера. Развитие идей конвейеризации выполнения программ предполагает обработку потоков данных различных типов (в общем случае *n* типов) соответствующими каждому из этих типов данных программами при реализации последовательного обмена данными между вычислительными сегментами программного конвейера (обмен данными между программами выполняется только после окончания их обработки на сегментах конвейера).

Конвейеризация программ предполагает их разделение на фрагменты, каждый из которых закреплен для выполнения за соответствующим сегментом конвейера (вычислительным устройством, обрабатывающим прибором). Введем в рассмотрение следующие обозначения: *i –* номер множества однотипных данных, характеризующих одинаковые объекты, которые должны быть обработаны в системе, *I* –множество всех данных, которые будут обработаны в вычислительной системе, *n –* количество множеств данных, тогда ),  – количество элементов в множестве однотипных данных, характеризуемых индексом *i* (множество содержит данные об  одинаковых объектах, в общем виде ). Данные, входящие в некоторое *i-*е множество, обрабатываются соответствующей им программой. Индекс *i* соответствует программе, выполняемой в составе конвейера, обрабатывающей данные *i-*го типа (соответствует типу выполняемой в составе конвейера программы, обрабатывающей данные *i-*го типа). Однократное выполнение конвейеризированной программы *i-*го типа обеспечивает обработку одного элемента множества данных *i-*го типа. Если множество данных *i-*го типа содержит  элементов, то обрабатывающая эти данные программа должна быть выполнена в конвейерной системе  число раз. Цель функционирования конвейерной системы в этом случае состоит в обработке поступающих на ее вход данных выполняющимися в системе конвейеризированными программами. При этом предполагается, что выполняющие обработку данных конвейеризированные программы находятся в оперативной памяти каждого из сегментов конвейера (загружены в оперативную память вычислительных устройств кластера). Тогда управление вычислительным процессом в конвейерных системах предполагает определение порядка запуска программ обработки данных на выполнение. Т.к. объемы вычислений на каждом сегменте различны, являются различными длительности выполнения программ на соответствующих сегментах, тогда может быть сформировано расписание выполнения конвейеризированных программ обработки соответствующих данных, представляющее собой порядок запуска программ на выполнение. Определение подхода по конвейеризации обработки данных выполняющимися в многостадийных системах программами рассматривается в [2-4].

Постановка задачи управления вычислительным процессом, рассматриваемой в данной работе, предполагает, что при  () реализуется обработка единичных данных (однократный запуск на выполнение соответствующих программ), для действий с которыми должно быть сформировано расписание выполнения программ. Введем обозначение для интервала времени функционирования системы при обработке данных в виде , где z–индекс временного интервала, тогда , где Z– количество временных интервалов , в течение которых выполняется обработка данных. В этом случае управление вычислительным процессом предполагает формирование совокупности данных, обрабатываемых в течение каждого из временных интервалов и расписаний обработки данных каждой из этих совокупностей. Введем в рассмотрение понятия задания на обработку как совокупности единичных данных, обрабатываемых в течение одного временного интервала . Для задания на обработку введем в рассмотрение обозначение , тогда должно быть сформировано *Z* заданий (т.е.). Тогда управление вычислительным процессом состоит в формировании эффективных составов заданий  () на обработку данных и расписаний обработки данных этих заданий. Т.к. задачи теории расписаний при количестве приборов более трех и отсутствии каких либо ограничений на длительности обработки данных являются трудно разрешимыми (NP–трудными, [5]), тогда для их решения должны быть применены приближенные методы. Одним их способов формирования приближенных методов теории расписаний является «жадный» подход, применение которого к сформулированной задаче будет развито в данной работе.