

A 05.13.04  
ԱԼԴ-91

« ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱԿՏՈՄԱՏԱՑՄԱՆ  
ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Մովսիսյան Միքեյ Յուրիի

ԿԼԱՍՏԵՐՈՅԻՆ ՄԻՋԱՎԱՅՐՈՒՄ ԽՆԴԻՐՆԵՐԻ ԴԻՆԱՄԻԿ  
ՂԵԿԱՎԱՐՄԱՆ ՀԱՄԱՐ ԳՈՐԾԻՔԱՅԻՆ ՄԻՋՈՑՆԵՐԻ ՄՇԱԿՈՒՄ

Ե. 13.04 «Հաշվողական մեթոդների, համալիրների, համակարգերի և  
ցանցերի մաթեմատիկական և ծրագրային սպահովում»  
մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի  
գիտական աստճանի հայցման ատենախոսության

Ս Ե Ղ Մ Ա Գ Ի Ր

Երևան 2008

---

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА

Мовсисян Мгер Юрьевич

РАЗРАБОТКА ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ДИНАМИЧЕСКОГО  
УПРАВЛЕНИЯ ЗАДАНИЯМИ В КЛАСТЕРНОЙ СРЕДЕ

А В Т О Р Е Ф Е Р А Т

диссертации на соискание ученой степени кандидата технических наук по  
специальности 05.13.04 “ Математическое и программное обеспечение  
вычислительных машин, комплексов, систем и сетей ”


Ереван 2008

Ատենախոսության թեման հաստատվել է ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում:

Գիտական ղեկավար՝ ֆիզ. մաթ. գիտ. թեկ., դոցենտ Վ. Գ. Սահակյան  
Պաշտոնական ընդդիմախոսներ՝ տեխ. գիտ. դոկտ., պրոֆ. Ա. Խ. Պալյան  
տեխ. գիտ. թեկ. Ս. Լ. Կյուրջյան  
Առաջատար կազմակերպություն՝ Երևանի ավտոմատ կառավարման համակարգերի ԳՀԻ

Ատենախոսության պաշտպանությունը տեղի կունենա 2008թ. հունիսի 15-ին ժամը՝ 16:00 ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի 037 «Ինֆորմատիկայի և հաշվողական համակարգերի» մասնագիտական խորհրդի նիստում, հետևյալ հասցեով՝ Երևան, 0014, Պ. Սևակ, 1:

Ատենախոսությանը կարելի է ծանոթանալ ինստիտուտի գրադարանում:  
Սեղմագիրը առաքված է 2008 թ. մայիսի 15-ին:

Մասնագիտական խորհրդի գիտական քարտուղար  ֆ. մ. գ. դ. Ս. Ե. Հարությունյան

Тема диссертации утверждена в Институте проблем информатики и автоматизации НАН РА.

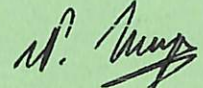
Руководитель: к.ф.м.н., доцент В. Г. Саакян

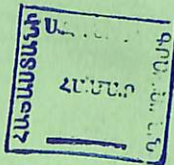
Официальные оппоненты: д.т.н., профессор А. Х. Палян  
к.т.н. М. К. Горджян

Ведущая организация: Ереванский НИИ автоматизированных систем управления

Защита диссертации состоится 15 июня 2008 г. в 16:00 часов на заседании специализированного совета 037 "Информатика и вычислительные системы" Института проблем информатики и автоматизации НАН РА по адресу: 0014, г. Ереван, ул. П. Севака, 1.

С диссертацией можно ознакомиться в библиотеке института  
Автореферат разослан 15-го мая 2008 г.

Ученый секретарь  
специального совета  д.ф. м. н. М. Е. Арутюнян



## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

### Актуальность темы

Вычислительные кластеры, сконструированные из узлов персональных компьютеров и соединенные обычной сетью, приобретают все большую популярность. Такие кластеры являются более дешевой альтернативой суперкомпьютерам. В отличие от суперкомпьютеров кластеры не предоставляют интегрированные сервисы операционной системы. Для организации вычислений в кластерной системе пишется параллельная программа на основе модели передачи сообщений (message-passing model). В модели передачи сообщений вычисления производятся группой процессов, которые общаются между собой посредством сообщений. Процессы вызывают библиотечные функции для отправки и приема сообщений. Данные от одного процесса к другому отправляются вызовом функции отправки и не принимаются пока получающий процесс не вызывает соответствующую функцию приема. Среди многочисленных реализации модели передачи сообщения самым распространенным является интерфейс передачи сообщений MPI (Message Passing Interface), позволяющий создавать параллельные программы модели SPMD (Single Program Multiple Data) и обеспечивающий механизм взаимодействия между процессами, работающими на различных узлах кластера в рамках одной параллельной программы.

Кластеры являются многопользовательскими системами. Пользовательские параллельные программы отправляются кластеру в виде заданий. Система управления заданиями кластера планирует и распределяет выполнение задания по узлам. Существующие системы управления заданиями предоставляют широкий спектр возможностей управления заданиями. Но во время выполнения заданий операции управления ограничиваются операциями выяснения статуса и аварийного завершения. Такие динамические операции как временная остановка выполнения заданий или миграция процессов параллельной программы отсутствуют в стандартных инструментальных средствах управления.

В то же время при организации вычислений в кластерных системах существует ряд проблем, которые могут быть решены лишь на основе возможности динамического управления заданиями. Важнейшими из этих проблем являются приоритетные прерывания заданий, миграция процессов и обеспечение отказоустойчивости. Поэтому разработка программных средств динамического управления заданиями для использования как пользователями для управления выполняемыми заданиями, так и самой системой для устранения проблем возникающих в аппаратуре, является актуальной задачей.

### Цель и задачи работы

Данная работа посвящена созданию инструментальных средств обеспечения механизма динамического управления выполняемыми заданиями на основе технологии создания контрольных точек и восстановления параллельной программы, их интегрированию в кластерную программную инфраструктуру и внедрению в рабочую среду высокопроизводительной вычислительной системы "Armcluster".

Для решения этих задач:

- исследованы технологии создания контрольных точек параллельных программ для методов коммуникации MPI и предложена комбинированная модель с гибкими возможностями применения;

- разработана библиотека прозрачного создания контрольных точек для MPI программ;
- исследованы функциональные возможности существующих систем управления заданиями с точки зрения динамической управляемости;
- разработана модель системной архитектуры кластера, обеспечивающая возможность динамического управления заданиями и предоставляющая возможность расширения базовой функциональности динамического управления;
- в среде вычислительного кластера разработана и внедрена система управления заданиями, реализующая описанную модель динамического управления.

#### Объект исследования

Объектами исследования являются методы организации выполнения параллельных программ на кластере, методы управления заданиями на кластере и методы восстановления параллельных программ на основе созданных контрольных точек.

#### Методы исследования

Теоретической основой исследования является теория параллельных вычислений. Прикладная часть основана на системном программировании на C для UNIX систем, клиент/серверной модели программирования.

#### Научная новизна

- Разработана технология прозрачного создания контрольных точек и восстановления MPI программ, которая учитывает особенности методов коммуникации MPI;
- Разработана модель системной архитектуры кластера, которая поддерживает динамическое управление заданиями и предоставляет возможность расширения операций динамического управления.

#### Практическая значимость полученных результатов

На основе разработанной технологии создания контрольных точек и восстановления MPI программ реализована библиотека CROM (Checkpointing and Recovery of MPI), которая предоставляет возможность создания контрольных точек и восстановления выполняемых MPI программы без изменения кода программы. Библиотека поддерживает все методы коммуникации MPI и накладывает на программу минимальные ограничения.

На основе библиотеки CROM реализована система динамического управления заданиями DJM (Dynamic Job Manager), которая может быть внедрена в систему управления заданиями кластера и использована для организации приоритетного прерывания, миграции процессов или повышения отказоустойчивости. Программный интерфейс системы управления заданиями предоставляет возможность реализации системных утилит для повышения отказоустойчивости.

#### Внедрение

Модель динамического управления заданиями внедрена в состав программного обеспечения высокопроизводительной вычислительной системы "Armcluster".

#### На защиту выносятся следующие положения

- Технология создания контрольных точек для MPI программ
- Модель системной архитектуры вычислительного кластера, обеспечивающая динамическое управление заданиями и предоставляющая возможность расширения операций динамического управления
- Реализация библиотеки прозрачного создания контрольных точек для MPI программ

#### Апробация полученных результатов

Основные положения и результаты обсуждались на семинарах Института проблем информатики и автоматизации НАН РА, докладывались на международной конференции CSIT2007.

#### Публикации

По теме исследований напечатано четыре научных статьи, список которых приводится в конце автореферата.

#### Структура и объем работы

Диссертация состоит из введения, трех глав, заключения и списка использованной литературы. Объем работы – 106 стр. и 66 наименования цитируемой литературы.

### СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы, описываются основные цели, задачи, объекты и методы исследования. Отмечается новизна результатов, их практическая значимость и апробация, приводятся реализованные внедрения. Также описываются поставленные задачи, решенные в диссертационной работе и их логическая связь.

В первой главе вводятся основные понятия теории параллельных вычислений, описывается современное состояние программной инструментальной базы для организации параллельных вычислений на распределенных вычислительных системах кластерного типа. Приводится описание технологии создания параллельных программ с использованием стандарта MPI. Обозначаются задачи, возникающие при организации вычислений в кластерных системах.

Описываются системы управления заданиями, и рассматривается их роль в организации процесса вычисления в кластерных системах. Описываются существующие архитектуры систем управления заданиями и их основные компоненты. Дается обзор функциональности систем управления заданиями, и рассматриваются предоставляемые возможности с точки зрения динамической управляемости. Приводятся достоинства и недостатки рассмотренных систем управления заданиями, предоставляющие возможность динамического управления.

Рассматривается проблема динамического управления заданиями с использованием технологии восстановления выполняемой программы от заранее созданных контрольных точек. По этой технологии выполняемая программа сохраняется в виде данных, а потом выполнение возобновляется в новой среде. Сохраненные состояния называются контрольными точками и содержат всю информацию необходимую для восстановления выполнения программы.

Приводятся различия между созданием контрольных точек последовательных и параллельных программ. В отличие от последовательных программ, создание контрольных точек для параллельных программ усложняется тем, что сообщения, с помощью которых взаимодействуют процессы параллельной программы, приводят к межпроцессной зависимости. При независимом создании контрольных точек глобальное состояние параллельной программы может отражать такое состояние, которое невозможно при нормальном выполнении. Описываются существующие механизмы создания контрольных точек для последовательных программ, и дается обзор протоколов создания контрольных точек для параллельных программ. Описываются существующие программные средства, предоставляющие возможность создания контрольных точек и восстановления MPI программ. Приводятся достоинства и недостатки рассмотренных систем, и дается анализ этих средств для организации динамического управления заданиями.

В последнем параграфе первой главы приводятся постановки задач решенных в диссертации, обосновывается их актуальность и логическая связанность друг с другом. Приводятся проблемы, покрываемые решением этих задач.

Во второй главе исследования посвящены разработке технологии прозрачного создания контрольных точек и восстановления MPI-программ.

Дается определение MPI-программы, описываются основные способы организации MPI программ, которые допустимы при применении предложенной технологии создания контрольных точек и восстановления. Перечисляются ограничения, которые накладывает технология на MPI-программы.

Для того, чтобы предлагаемая технология создания контрольных точек и восстановления была применима для разных реализаций стандарта MPI (стандарт MPI описывает только синтаксис и семантику функций отправки сообщения между процессами параллельной программы, а разные реализации MPI имеют разные структуры выполняемых MPI-программ), разработана и описана модель выполняемой MPI программы. Согласно этой модели структура выполняемой MPI программы разделена на две части: MPI-процессы и окружение MPI. MPI-процессы выполняют параллельную программу, вызывая функции MPI для отправки и приема сообщений, а окружение создает процессы параллельной программы и предоставляет информацию для взаимной связи. В модели также описаны основные свойства MPI-программ и каналов коммуникаций. В этой модели процессы MPI взаимодействуют между собой посредством сообщений, которые отправляются по надежным каналам коммуникаций. Каналы не являются типа FIFO, каждый процесс соединен с другим процессом единственным каналом коммуникации и все сообщения, независимо от коммутаторов и тегов, отправляются по этому каналу.

Рассматривается понятие непротиворечивого состояния параллельной программы. Непротиворечивое состояние параллельной программы - это одно из состояний, через которое проходит программа во время нормального выполнения. В непротиворечивом состоянии параллельной программы если состояние одного процесса свидетельствует, что получено сообщения, то состояния соответствующего

процесса отправителя должно свидетельствовать об отправке этого сообщения. Описываются условия непротиворечивости для параллельных программ, использующие модель отправки сообщения. Рассматривается непротиворечивое состояние MPI-программы, и вводятся дополнительные условия непротиворечивости. Дополнительные условия непротиворечивости гарантируют, что после создания контрольных точек и восстановления семантика MPI-функции не изменится. Например, для синхронных функций отправки сообщений требуется, чтобы возврат из функции был только после доставки сообщения, независимо от того, были ли во время вызова функции созданы контрольные точки или нет.

Описывается процесс создания контрольных точек на основе созданной модели MPI-программы. Согласно этой модели сохраняются только состояния MPI-процессов. Окружение MPI не сохраняется. При восстановлении выполнения окружение MPI создается заново. Сохраненное состояние MPI-программы состоит только из множество контрольных точек MPI-процессов. Состояние каналов коммуникации также не входят в состояние MPI-программы.

Описывается протокол создания контрольных точек для MPI-программ. Для того, чтобы создать контрольные точки для MPI-программ используется координированный протокол создания контрольных точек. По этому протоколу создание контрольных точек выполняется в четыре этапа:

1. Блокирование отправки сообщений
2. Очищение каналов коммуникаций
3. Сохранение отдельных процессов
4. Возобновления коммуникаций

На первом этапе блокируется отправка сообщений, поскольку во время создания контрольных точек отправленные сообщения могут привести к противоречивому состоянию.

На втором этапе выполняется очищение каналов коммуникации. Перед созданием контрольных точек каналы коммуникации очищаются от транзитных сообщений. Очищение каналов коммуникации преследует две цели. Во-первых, оно гарантирует глобальное непротиворечивое состояние - отсутствие транзитных сообщений лишает межпроцессной зависимости. Во-вторых, оно упрощает реализацию, поскольку трудно сохранить сообщения, находящиеся в каналах коммуникации. Транзитные сообщения во время создание контрольных точек могут находиться в таких труднодоступных местах, как сеть или буферы операционной системы. Очищение каналов коммуникации выполняется посредством отправки специальных сообщений. Все процессы обмениваются сообщениями, которые содержат информацию для определения состояния каналов коммуникации. Во время очищения каналов коммуникации все транзитные сообщения буферизуются в адресных пространствах MPI-процессов.

На третьем этапе процессы сохраняют свои состояния. Состояние MPI-программы состоит из множество созданных контрольных точек.

На четвертом этапе выполняется возобновление работы коммуникаций. MPI-процессы синхронизируют момент завершения работы создания контрольных точек и восстанавливают процесс отправки сообщений.

Описывается алгоритм определения состояния каналов коммуникаций, используемый протоколом создания контрольных точек на этапе 2 для выяснения наличия транзитных сообщений в каналах коммуникаций. Приводится псевдокод

алгоритма. По алгоритму наличие транзитных сообщений определяется сравнением количества полученных и отправленных сообщений.

Рассматриваются семантики всех методов коммуникаций MPI во время создания контрольных точек и восстановления. Определяется понятие точки блокирования. Точка блокирования - эта самая последняя безопасная точка, которую может достичь сообщение во время создания контрольных точек. Пересечение точки блокирования может привести к противоречивому состоянию. Для каждого метода коммуникации MPI определяется точка блокирования. Определение точки блокирования показывает, что для большинства функции MPI можно обойтись без блокирования, отложив пересечение точки блокирования после окончания процесса создания контрольных точек и, при этом, не нарушить семантику функции. То есть, введение понятия точки блокирования позволяет избежать блокирования во время создания контрольных точек.

Приведено описание библиотеки создания контрольных точек и восстановления MPI программ. Описаны основные модули библиотеки, их функциональность и методы взаимодействия. Библиотека состоит из модуля создания контрольных точек, реализации стандарта MPI, обработчика запросов создания контрольных точек, интерфейса управления (PMI) клиента и программного интерфейса библиотеки (рис. 1).

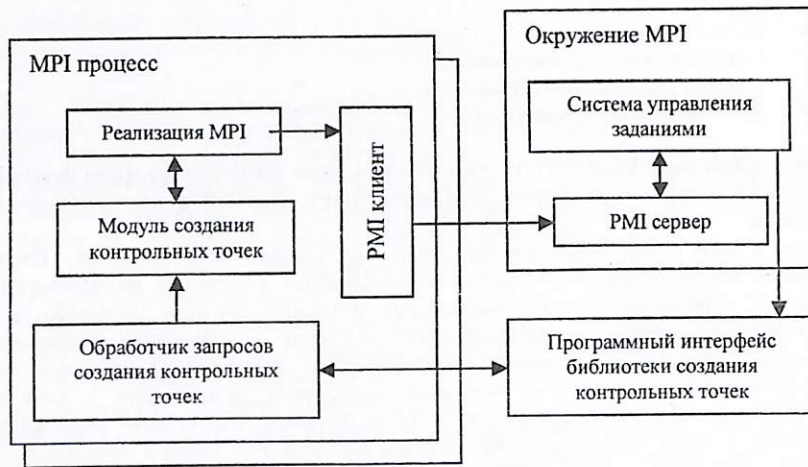


Рис. 1

Библиотека создания контрольных точек и восстановления позволяет реализовать предложенную модель MPI-программы. Для взаимодействия MPI-процессов и окружения MPI спроектирован и описан интерфейс управления MPI-программ. На примере показан механизм взаимодействия MPI-процессов с окружением.

Описано использование библиотеки для подготовки MPI-программы с функциональностью создания контрольных точек и восстановления. Приведено описание программного интерфейса библиотеки для внедрения в систему управления заданиями.

Рассмотрены такие проблемы реализации, как прозрачное создание контрольных точек, переносимость и ограничения на программное обеспечение при создании

контрольных точек для последовательных программ. Описан механизм обеспечения прозрачности и преодоления ограничений программных пакетов при создании контрольных точек и восстановления последовательных программ.

Дана оценка эффективности создания контрольных точек и восстановления. Эффективность технологии рассмотрена по трем основным характеристикам: время создания контрольных точек и восстановления, воздействие процесса создания контрольных точек и восстановления на производительность программы и объем контрольных точек.

Экспериментально полученные результаты представлены графиками зависимости времени создания контрольных точек от числа процессов в параллельной программе и от размера физической памяти процессов.

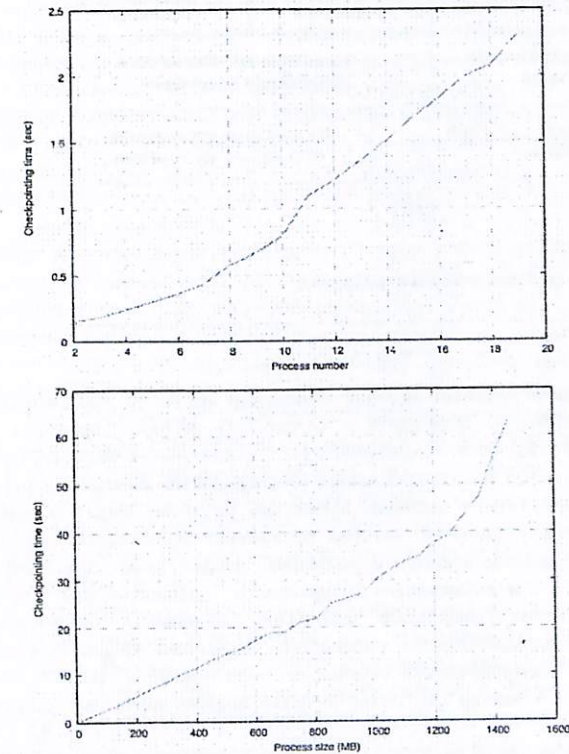


Рис. 2

Полученная оценка эффективности может быть использована для определения пригодности технологии для конкретного применения.

В третьей главе приводится описание создания системы динамического управления заданиями.

Дается понятие динамического управления, и описываются его базовые операции. Описывается созданная модель кластерной архитектуры, которая предоставляет возможность динамического управления (рис. 3). В этой модели на верхнем уровне находится менеджер очереди кластера. Основные обязанности менеджера очереди являются прием заданий от пользователей и распределение их по очередям.



Рис. 3

Менеджер очереди отправляет задания менеджеру ресурсов для выполнения. Менеджер ресурсов состоит из серверной и клиентских частей. Серверная часть расположена на головной машине и взаимодействует с клиентскими частями, которые находятся на каждом из узлов кластера. Главной обязанностью менеджера ресурсов является организация выполнения задания. Менеджер ресурсов имеет подсистему динамического управления заданиями, которая организует выполнение заданий на узлах кластера и их динамическое управление. Подсистема динамического управления заданиями предоставляет возможность вышестоящим уровням реализовать разные сценарии управления заданиями. Основными динамическими операциями подсистемы динамического управления заданиями являются создание контрольной точки для выполняемого задания и возобновление выполнения из контрольной точки. После создания контрольной точки задание может завершить или продолжить выполнение.

Серверная часть менеджера ресурсов использует клиентские части для запуска и управления процессов параллельной программы на конкретных узлах. На каждом из узлов имеется подсистема управления контрольными точками. Подсистема управления контрольными точками ответственна за перемещение контрольных точек в стабильное хранилище. Подсистема управления контрольными точками выполняет транспортировку контрольных точек не зависимо от процесса выполнения.

На нижних уровнях находятся библиотека MPI и параллельная программа. Библиотека MPI взаимодействует с подсистемой управления контрольными точками

и менеджером ресурсов для организации выполнения MPI программы и создания контрольных точек.

Применение описанной системной архитектуры отличается от обычной конфигурации кластера, поскольку в данном случае помимо обеспечения среды для выполнения параллельной программы, обеспечивается функциональность динамического управления заданиями, которая может быть использована для разных операций над заданиями, находящимися в очередях системы.

Предложенная архитектура имеет удобную платформу для расширения области применения динамического управления. На основе представленной архитектуры могут быть реализованы разные сценарии динамического управления, такие как миграция процессов или приоритетное прерывание. Могут быть также реализованы сложные и нестандартные операции динамического управления поверх базовых операций.

Динамическое управление в представленной модели реализуется подсистемой динамического управления выполняемыми заданиями, которая организует выполнение параллельной программы и предоставляет возможность управления.

Описывается модель подсистемы динамического управления выполняемыми заданиями. Спроектированная подсистема управления состоит из шести основных частей: сервера управления заданиями, демонов управления заданиями, менеджеров процессов, модуля управления контрольными точками, набора пользовательских команд и библиотеки программного интерфейса системы (рис. 4).

Сервер управления заданиями является главной функциональной частью системы. Он несет ответственность за запуск и управление, сбор информации о состоянии и завершение выполнения задания. Сервер управления заданиями устанавливается на головной машине кластера и является связующим звеном между остальными частями системы. Сервер управления заданиями при помощи клиентов (демонов) управления заданиями запускает процессы параллельной программы на узлах и посылает команды управления.

Клиенты (демоны) управления заданиями устанавливаются на каждом из узлов кластера и несут ответственность за организацию запуска конкретного процесса параллельной программы. Для запуска процесса параллельной программы сначала создается менеджер процесса, который в свою очередь создает процесс параллельной программы. Клиенты (демоны) управления заданиями запускаются во время инициализации системы динамического управления заданиями после запуска сервера управления заданиями.

Менеджеры процессов обеспечивают окружение параллельной программы. Они предоставляют информацию для взаимодействия процессов параллельной программы. Процесс параллельной программы для отправки сообщения другому процессу получает адрес от менеджера процесса. Менеджер процесса запускает процесс параллельной программы и отправляет команды создания контрольных точек.

Модуль управления контрольными точками несет ответственность за сбор контрольных точек. Взаимодействуя с менеджерами процессов, контрольные точки параллельной программы от всех узлов отправляются в стабильное хранилище, управление которого осуществляет модуль управления контрольными точками. Во время восстановления задания модуль управления контрольными точками предоставляет соответствующую контрольную точку задания для организации процесса восстановления.

Библиотека программного интерфейса системы предоставляет функции для запуска и динамического управления задания на кластере. С помощью библиотеки программного интерфейса можно интегрировать систему динамического управления заданиями в систему управления задания кластера. Пользовательские команды взаимодействуют с системой с помощью библиотеки программного интерфейса.

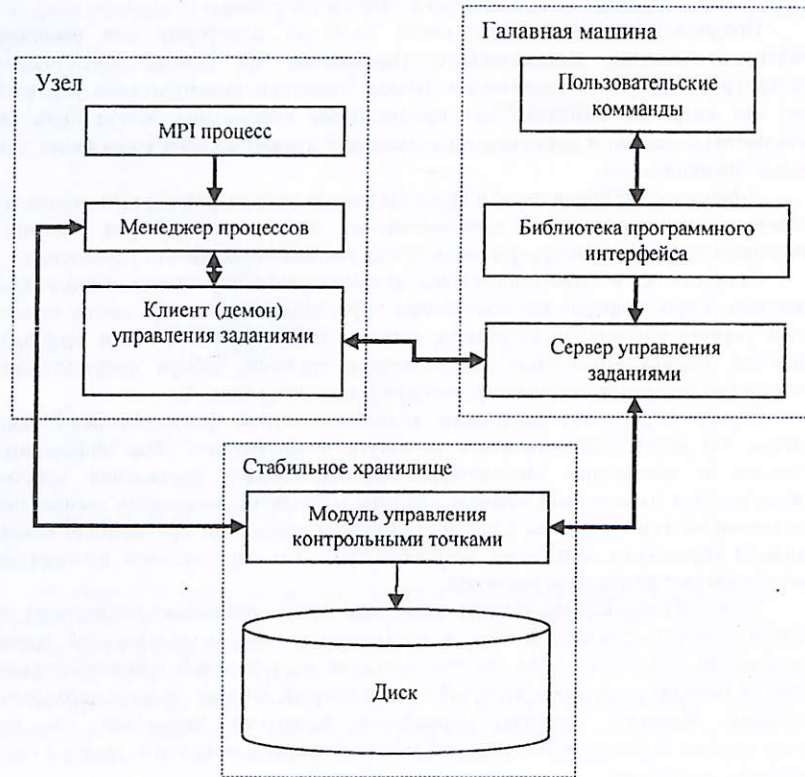


Рис. 4

Пользовательские команды доступа к системе позволяют взаимодействовать с системой с помощью приложений командной строки. Существуют команды для запуска задания и динамического управления. Пользовательские команды позволяют интегрировать систему динамического управления заданиями в систему управления заданиями кластера с помощью файлов конфигурации.

Детально описываются процессы организации выполнения и динамического управления для MPI-программ.

Описывается разработанная система управления контрольными точками, которая выполняет роль стабильного хранилища. Описываются форматы контрольных точек и способы хранения. Вводятся понятия последней и текущей контрольной точки и описывается механизм транзакции контрольных точек.

Вводится понятие импорта и экспорта заданий и описывается соответствующая функциональность, которая позволяет переместить выполняемое задание из одной системы управления задания в другую.

Представлены пользовательские команды системы динамического управления и основные сценарии использования. Представлены методы внедрения в систему управления заданиями. Рассмотрена и описана библиотека программного интерфейса системы. Приведены примеры использования этой библиотеки для разработки программ повышенной отказоустойчивости.

#### Основные результаты и выводы

- Разработана технология прозрачного создания контрольных точек для MPI программ и реализована в виде библиотеки. Основными отличиями технологии от существующих являются поддержка асинхронных и коллективных методов коммуникаций MPI и наличие программного интерфейса для внедрения в систему управления заданиями [1, 2].
- Разработана модель системной архитектуры кластера, поддерживающая динамическое управление заданиями. Модель предоставляет общую платформу для динамических операций управления и возможности расширения [4].
- Создана система динамического управления заданиями, реализующая модель приведенной системной архитектуры кластера. Система дает возможность внедрения функциональности динамического управления в систему управления заданиями как программно, используя программный интерфейс, так и с помощью конфигурации используя пользовательские команды [4].
- Осуществлено внедрение созданных программных пакетов в систему высокопроизводительного вычислительного кластера Armcluster.

#### Перечень публикаций по теме диссертации

1. Movsisyan M., Sahakyan V., Transparent checkpointing protocol for MPI programs with decentralized initiator. Proceedings of International Conference on Computer Science and Information Technologies (CSIT2007), pp. 227 – 229, Yerevan, 2007
2. Мовсисян М., Создание контрольных точек и восстановление MPI программ. Mathematical Problems of Computer Science, Transactions of IAP NAS RA, Volume 30, pp. 47 – 53, Yerevan, 2008
3. Մովսիսյան Մ., Խնդիրների ղեկավարման համակարգի կազմակերպումը հաշվողական կլաստերներում. Գիտություն և տեխնիկա, 01, 14-19, Երևան, 2008
4. Мовсисян М., Организация динамического управления заданиями в кластерных системах. Препринт ИПИА НАН РА, 2008. – 28 с.

Սովսիսյան Սիեր Յուրիի

Կլաստերային միջավայրում խնդիրների դինամիկ ղեկավարման համար գործիքային միջոցների նշակում

ԱՄՓՈՓԱԳԻՐ

Աշխատանքը նվիրված է կլաստերային միջավայրում խնդիրների դինամիկ ղեկավարման համար գործիքային միջոցների նշակմանը, իրականացմանը և դրանց ներդրմանը բարձր արտադրողականության Armcluster հաշվողական կլաստերի ծրագրային ապահովման կազմում:

Այդ կապակցությամբ ձևակերպվել են հետևյալ խնդիրները՝

- հետազոտել զուգահեռ ծրագրերի պահպանման և վերաթողարկման միջոցները հաղորդագրությունների փոխանցմամբ զուգահեռ համակարգերի համար,
- հետազոտել զուգահեռ ծրագրերի պահպանման և վերաթողարկման տեխնոլոգիան MPI կոմունիկացիայի բոլոր մեթոդների համար,
- մշակել և իրականացնել MPI ծրագրերի թափանցիկ պահպանման և վերաթողարկման համար գրադարան, որը հնարավորություն է ընձեռնում ներդնել այն խնդիրների ղեկավարման համակարգի մեջ,
- հետազոտել գոյություն ունեցող խնդիրների ղեկավարման համակարգերի ֆունկցիոնալ հնարավորությունները դինամիկ ղեկավարման տեսանկյունից,
- մշակել կլաստերային խնդիրների ղեկավարման մոդել, որը դինամիկ ղեկավարման հնարավորություն է ընձեռնում և կիրառելի է ավելի բարդ դինամիկ ղեկավարման միջոցների իրականացման համար,
- մշակված մոդելի հիման վրա իրականացնել խնդիրների դինամիկ ղեկավարման համակարգ, որը հնարավորություն է ընձեռնում ինտեգրել կլաստերային ծրագրային միջավայրի մեջ,

Լուծելով այդ խնդիրները՝ ստացվել են հետևյալ արդյունքները և եզրահանգումները՝

- մշակվել է կլաստերի ղեկավարման մոդել, որը դինամիկ ղեկավարման հնարավորություն է ընձեռնում,
- ստեղծվել է DJM (Dynamic Job Manager) խնդիրների դինամիկ ղեկավարման համակարգը, որը իրականացնում է կլաստերի դինամիկ ղեկավարման համար մշակված մոդելը,
- մշակվել է MPI ծրագրերի թափանցիկ պահպանման և վերաթողարկման տեխնոլոգիա և իրականացվել է CROM (Checkpointing and Recovery of MPI) գրադարանի տեսքով,
- խնդիրների դինամիկ ղեկավարման DJM համակարգը և MPI ծրագրերի թափանցիկ պահպանման և վերաթողարկման գրադարանը ներդրվել են բարձր արտադրողականության Armcluster հաշվողական կլաստերի ծրագրային ապահովման կազմում:

Mher Y. Movsisyan

The Design of the Tools for the Dynamical Job Control in Cluster Environments

Ծավալը – 1 տ.թ. Տպաքանակը – 100 օր.  
Տպագրված է ՀՀ ԳԱԱ ԻԱՊԻ կոմպյուտերային  
պոլիգրաֆիայի լաբորատորիայում

ՀՀ Ազգային գրադարան



NL 1694693

