

1 2005 год

1.1 Файловый индекс

1.1.1 Общее описание

Система предназначена для сбора, хранения и анализа информации о состоянии локальных дисков параллельной вычислительной системы с целью обеспечить возможность работы с группами файлов, представляющих состояния вычислительных процессов.

Основными сущностями в системе управления ЛДП являются: диск — локальная файловая система некоторого вычислительного узла, файл, задача и контрольная точка — группа файлов, ассоциированная с задачей. Создаваемые по мере работы приложения контрольные точки регистрируются в системе, после чего происходит планирование операций создания избыточной информации для предотвращения возможных сбоев.

Благодаря тому, что вся информация находится в файловом индексе, можно обеспечить выполнение операций восстановления в случае сбоев или временной недоступности вычислительных узлов, проверку целостности файлов и их групп, а также автоматическое удаление устаревшей информации.

Основной менеджер системы реализован с использованием языка высокого уровня Haskell, что позволяет обеспечить удобное изменение алгоритмов поддержания целостности и легкость построения и проверки математических моделей.

1.1.2 Дисковые демоны

Кроме центрального индекса созданы специальные дисковые демоны, обеспечивающие низкоуровневые операции с файлами, такие как хеширование или удаление. На данный момент реализованы некоторые основные команды, такие как удаление файла, подсчет и проверка его контрольной суммы (MD5). Кроме этого, на дисковый демон возлагается задача создания избыточной информации и перемещения файлов между узлами.

При запуске он соединяется с индексом и передает ему информацию о диске, на котором запущен. В случае, если в индексе этого диска не было — он добавляется, иначе информация о нем обновляется. Идентификатор диска создается методом из библиотеки `libccp2` (см. ??).

Дисковый демон хранит список всех файлов, заведенных в индексе для данного диска — при старте он запрашивает у индекса текущий список; после этого регистрируется перенаправление для данного диска и при добавлении файла информация о нем поступает в демон. При выполнении операции, требующей обновления информации в индексе, такой как вычисление контрольной суммы, посылается соответствующая команда.

Создание избыточной информации происходит следующим образом. Сначала на всех демонах инициализируется операция с уникальным идентификатором. После

этого для каждого процесса, участвующего в построении коммуникационной сети, передается набор команд по выстраиванию его локальных `bio` и указания, с какими надо соединяться по сети или какие соединения принимать (подробнее о построение коммуникационных сетей написано в ??). Если в процессе построения произойдет сбой, данная незаконченная операция будет удалена на всех демонах. После этого, всем процессам высылается команда запустить операцию. Не участвующие при этом удалят у себя информацию о ней как ненужную. При таком подходе устраняется опасность возникновения записей, не относящихся ни к одной запущенной или еще готовящейся операции. При этом логика построения сетей вынесена в индекс, где могут быть реализованы рассмотренные в ?? алгоритмы.

На данный момент возможно построение любых сетей с использованием файлов, TCP-соединений и вычисления XOR-сумм, этого достаточно для обеспечения отказоустойчивости по схеме XOR (см. ??)

1.1.3 Интерфейс

Процесс создает сетевой сервер на TCP порту 2563 и считывает команды из стандартного ввода. Кроме этого, создается сервер на TCP порту 2564 принимающий SSLv3 соединения с шифрованием RC4-SHA и проверкой сертификата клиента. Если проверка не проходит, соединение обрывается, таким образом есть возможность ограничить доступ к индексу лишь авторизованным приложениям или интегрировать в систему Globus. На данный момент есть некоторые проблемы с обработкой прокси-сертификатов задач, но их решение не представляет технической трудности и будет решено при внедрении системы.

На данный момент не реализована система разграничения прав в связи с тем, что система еще не подошла к стадии внедрения и простота отладки ставится выше необходимости оградить систему от несанкционированного доступа.

Ниже приводятся стандартные команды с описанием их действия и разбиению по группам.

Контрольные команды. Команды, используемые для управлением работой файлового индекса, такие как отключение сетевого сервера, отключение клиента или загрузки другой базы.

Команды, частично обрабатываемые в серверной части Эти команды не поступают напрямую в ядро индекса, а обрабатываются в C-сервере.

Команды приведения индекса к целостному виду.

Команды добавления или изменения данных Этот класс команд не предназначен для использования вручную. Они не обрабатываются в C-коде и имеют формат «команда» «тип данных» «данные». Данные команды, не начинающейся с символа /, представляют из себя строковое представление Haskell типов.

1.1.4 Маршрутизация сообщений

Команды в индексе отправляются какому-то конкретному получателю, что позволяет отсылать подключенным клиентам ответ на их запрос, а не на чужой. Реализуется это с помощью связи символического имени с соединением, при этом, одному соединению может быть присвоено несколько имен. Так, клиент может указать какие дополнительные символические имена ему нужны с помощью команды `/cntl/alias`. Таким образом, дисковый демон при подключении может указать, что кроме присвоенного ему при открытии TCP-соединения имени вида `inet-[номер соединения]`, он должен получать сообщения для `disk-[UUID диска]`. Кроме этого, могут существовать специальные имена. Например, `disk:broadcast`, при отправке команды на которое, копия рассылается всем дисковым демонам. Оно используется для построения коммуникационных сетей.