

Г.А. Звонарева¹, А.А. Терехов², А.А. Федоров²

¹ Московский авиационный институт, Москва, Россия

² АО «Научно-исследовательский институт «Аргон»», Москва, Россия

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ УПРАВЛЕНИЯ КЛАСТЕРОМ ВЫСОКОЙ ГОТОВНОСТИ ДЛЯ ОСРВ QNX NEUTRINO

Представлены аспекты разработки программного обеспечения управления для кластера высокой готовности, являющегося специализированным бортовым вычислительным комплексом на базе операционной системы реального времени QNX Neutrino, которая обладает высокой надежностью и необходимым функционалом. Рассмотрена отладка и тестирование разработанного программного обеспечения в среде разработчика Qt Creator, в том числе и на опытном образце кластера высокой готовности.

Ключевые слова: управление кластером высокой готовности, кластер высокой готовности, операционная система реального времени, QNX Neutrino, программное обеспечение управления, бортовой вычислительный комплекс.

Для цитирования: Звонарева Г. А., Терехов А. А., Федоров А. А. Программное обеспечение управления кластером высокой готовности для ОСРВ QNX Neutrino // Радиопромышленность. 2017. № 1. С. 6–12.

G. A. Zvonareva¹, A. A. Terekhov², A. A. Fedorov²

¹ Moscow Aviation Institute, Moscow, Russia

² Joint-stock Company Science Research Institute «Argon», Moscow, Russia

SOFTWARE FOR CONTROL OF HIGH-AVAILABILITY CLUSTER FOR RTOS QNX NEUTRINO

Herein are presented aspects of development of control software for high-availability cluster, which is a specialized on-board computational facility based on QNX Neutrino real-time operating system, which has a high reliability and the required set of functions. Debugging and testing of the developed software in Qt Creator developer environment have been reviewed, and inter alia, on the prototype of a high-availability cluster.

Keywords: control of high-availability cluster, high-availability cluster, real time operating system, QNX Neutrino, control software, on-board computational facility.

For citation: Zvonareva G. A., Terekhov A. A., Fedorov A. A. Software for control of high-availability cluster for RTOS QNX Neutrino. Radiopromyshlennost, 2017, no. 1, pp. 6–12 (In Russian).

DOI 10.21778/2413-9599-2017-1-6-12

Введение

Бурное развитие информационных технологий, рост обрабатываемых и передаваемых данных и в то же время повышение требований к надежности, степени готовности, отказоустойчивости и масштабируемости делает более привлекательной технологию кластеризации. Эта технология позволяет создавать довольно гибкие системы, которые отвечают всем вышеперечисленным требованиям.

Использование технологий кластеризации при разработке бортовых вычислительных комплексов (БВК) крайне актуально ввиду высоких требований к безопасности и надежности таких систем [3]. В случае выхода компонента из строя вся система продолжит работать дальше незаметно для пользователя, перераспределив задачи с вышедшего из строя вычислительного модуля. Особое внимание при разработке кластеров высокой готовности

уделяется программному обеспечению, которое управляет всей системой, выполняя такие функции, как перераспределение процессов с вышедших узлов кластера на рабочие, сбор статистики, выбор главного узла, создание контрольных точек и т.д. На данный момент актуальной является задача разработки программного обеспечения управления кластером высокой готовности для операционной системы реального времени (ОСРВ) QNX Neutrino, которая позволяет реализовать весь функционал кластерных систем для БВК.

ОСРВ QNX Neutrino является сетевой, мультитасочной, масштабируемой системой и соответствует высоким стандартам надежности и безопасности, что особенно важно для БВК [1].

Ниже рассматриваются основные аспекты разработки управляющего программного обеспечения кластера высокой готовности БВК с применением технологий, используемых в ОСРВ QNX Neutrino. Главной особенностью ядра ОСРВ QNX Neutrino является детерминированность, основанная на строгом контроле времени, обеспечивающая своевременную обработку запросов, что является важной особенностью для функционирования БВК.

Обзор используемых технологий

Микроядерность QNX Neutrino дает преимущества в построении систем высокой готовности. На уровне ядра выполняются только самые необходимые службы. Все компоненты работают в своих отдельных модулях, минимально взаимодействующих друг с другом. Все драйверы, стеки протоколов, файловые системы и приложения выполняются в защищенном пространстве пользователя, вне ядра. Благодаря этому не требуется перезагрузка при обновлении какого-либо компонента системы. Также никак не отразится на всей системе и других модулях повреждение одной из частей модуля. Теоретически любой компонент при отказе может быть перезагружен, и это никак не скажется на других компонентах системы. В случае сбоя операционная QNX Neutrino способна должным образом восстановиться в «известное исправное» состояние за счет самотестирования микроядра ОС. Если произошел сбой процесса, связанные с ним ресурсы могут быть очищены, а сам процесс перезапущен из «известного исправного» состояния. В совокупности это позволяет строить оптимизированные и очень доступные системы.

Операционная система реального времени QNX Neutrino позволяет проектировать сложные программные системы, работающие в реальном времени в локальной вычислительной сети. Встроенные средства операционной системы QNX Neutrino обеспечивают поддержку многозадачного режима при взаимодействии параллельно выполняемых

задач на разных компьютерах, работающих в среде локальной вычислительной сети.

Для обеспечения высокой стабильности ОСРВ QNX Neutrino наряду с микроядерной архитектурой с полной изоляцией модулей в оперативном запоминающем устройстве (ОЗУ) обладает списком особенно важных встроенных механизмов для кластерной организации БВК, работающего в режиме реального времени, из которых можно выделить [1, 2]:

- механизм адаптивного квотирования ресурсов процессора и ОЗУ;
- протокол сетевого взаимодействия Qnet;
- устойчивую службу публикации/подписки QNX (Persistent Publish/Subscribe – PPS).

Технология адаптивного квотирования

Данный механизм позволяет разделить процессорное время на квоты и гарантировать получение вычислительных ресурсов для критически важных приложений, что необходимо при функционировании БВК, работающих под управлением ОСРВ. Это не жесткое разделение, а назначение гарантированного бюджета для каждой квоты. Система при изменении внешних/внутренних условий может менять алгоритмы своего функционирования, достигая при этом оптимального распределения ресурсов:

- при полной нагрузке, когда все квоты требуют свой гарантированный бюджет;
- при наличии свободных ресурсов, когда приложения по необходимости могут выходить за рамки своего гарантированного бюджета и использовать свободное процессорное время других квот.

Безопасность приложений обеспечивается в любом состоянии системы. Пример работы адаптивного квотирования в ОСРВ QNX Neutrino приведен на рис. 1.

Технология адаптированного квотирования обеспечивает:

- повышение надежности за счет выделения гарантированного бюджета ресурсов (в том числе и при максимальной загрузке системы);
- повышение безопасности системы благодаря защите приложений от внешних воздействий (например, DoS-атака);
- эффективное использование аппаратных ресурсов за счет оптимального распределения свободного процессорного времени.

QNX Neutrino – система жесткого реального времени. Это значит, что задача будет выполнена даже в самом худшем случае. Это дает дополнительные

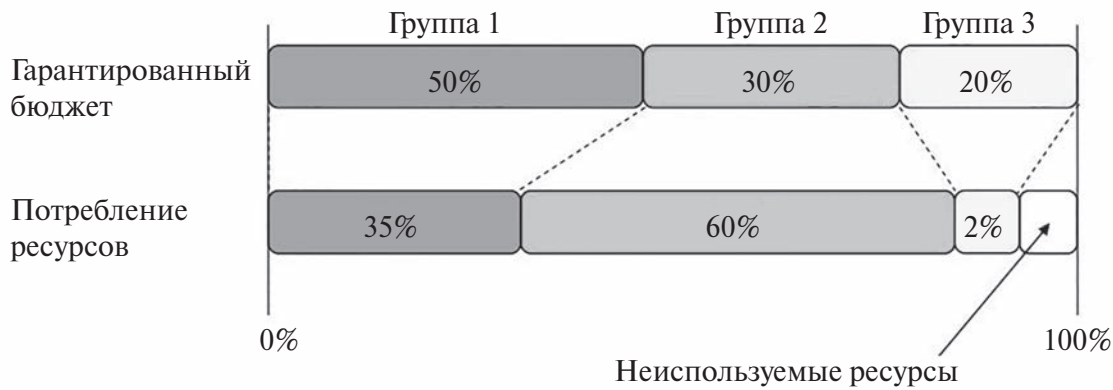


Рисунок 1. Распределение процессорного времени между квотами

плюсы в обеспечении безотказной работы и позволяет запускать даже потенциально ненадежные программы за счет полной защиты памяти.

Протокол сетевого взаимодействия Qnet

В операционной системе реального времени QNX Neutrino за основу собственного сетевого взаимодействия взят протокол Qnet, который реализуется в виде сильносвязанных доверяемых машин. Данный протокол позволяет этим машинам эффективно обмениваться ресурсами с минимальными накладными затратами, что позволяет организовать эффективное сетевое общение узлов кластера высокой готовности БВК. Одной из особенностей протокола Qnet является то, что он не выполняет аутентификацию удаленных запросов, а изменения файлов происходят за счет обыкновенных прав доступа, применяемых для пользователей и пользовательских групп. Кроме того, есть возможность запускать или останавливать процессы (в том числе различные администраторы) на любой машине, находящейся в сети Qnet.

Возможности распределенных вычислений в сети Qnet позволяют эффективно выполнять следующие задачи:

- использовать удаленную файловую систему;
- масштабировать приложения с невероятной легкостью;
- запускать или останавливать процессы (в том числе различные администраторы) на любой машине, находящейся в сети Qnet.

Более того, поскольку протокол Qnet позволяет перенести механизм обмена сообщениями на всю сеть, другие формы межзадачного взаимодействия (например, сигналы, очереди сообщений, именованные семафоры) также могут применяться в этой сети. Это значит, что можно иметь доступ к файлам или администраторам на других узлах сети Qnet, как если бы они находились на вашем локальном узле.

Устойчивая служба публикации/подписки QNX (Persistent Publish/Subscribe – PPS)

Устойчивая служба публикации/подписки QNX (Persistent Publish/Subscribe – PPS) – это небольшая расширяемая служба, которая обеспечивает хранение опубликованных данных при перезагрузке компьютера. Цель создания PPS – обеспечить простой в использовании механизм как для публикации/подписки, так и для устойчивого хранения данных во встроенных системах, требующих связи между независимыми программными компонентами посредством асинхронных публикаций и уведомлений.

Процесс PPS публикации организован асинхронно: подписчик никак не связан с издателем, они даже редко знают друг о друге. Единственная связь между ними – это некий объект, который известен обоим.

Хранение объектов во время функционирования PPS происходит в ОЗУ. При этом PPS, при необходимости, выполняет следующие действия:

- сохраняет объекты в энергонезависимом хранилище по запросу или при завершении;
- при запуске восстанавливает объекты из энергонезависимого хранилища немедленно или при первом доступе (отложенная загрузка).

PPS – это система, содержащая объекты, свойства которых модифицируются издателем, а клиенты, подписанные на данный объект, получают извещения при изменении этого объекта при его модификации издателем. Данный механизм позволяет создать легкий способ обмена данными между узлами кластера высокой готовности, соответствующий требованиям БВК реального времени.

Рис. 2 иллюстрирует разницу в информации, посылаемой подписчикам, которые открыли объект PPS в полном режиме и режиме delta.

В режиме delta подписчик получает только изменения (но все изменения) атрибутов объекта, в то время как в полном режиме (он используется

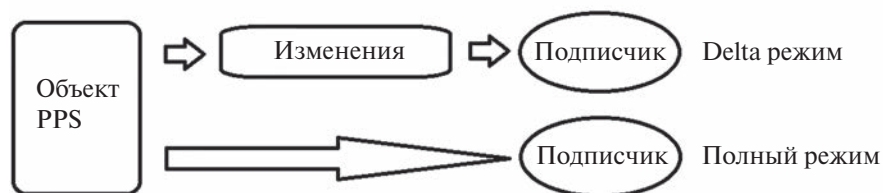


Рисунок 2. Полный режим и режим delta подписки PPS

по умолчанию) подписчик всегда получает целостную версию всего объекта в момент, когда он был запрошен.

Разработка управляющей программы кластера высокой готовности БВК для ОСПВ QNX Neutrino

Важной частью системы с технологией кластеризации является программное обеспечение, которое управляет всей системой. При разработке ПО для рассматриваемого класса систем необходимо реализовать администрирование процессов, миграцию процессов и данных, голосование и выбор главного узла, мониторинг узлов, создание контрольных точек и т.д. Ниже рассматриваются вопросы разработки программного обеспечения кластера высокой готовности, связанные с миграцией процессов с вышедших из строя узлов кластера на рабочие узлы, сбор статистики, выбор главного узла, мониторинг узлов и администрирование/контролирование процессов. В следующей публикации будут рассмотрены другие аспекты программного обеспечения управления кластером высокой готовности БВК для ОСПВ QNX Neutrino, в том числе создание контрольных точек.

Разработанное программное обеспечение управления кластером высокой готовности БВК для

ОСПВ QNX Neutrino, необходимое для реализации задач, рассматриваемых в данной публикации, содержит следующие модули (рис. 3):

- Модуль контроля узлов кластера высокой готовности. Этот модуль позволяет осуществлять скооперированную работу узлов посредством выбора главного узла, который распределяет их нагрузку. В обязанности модуля также входит выполнять мониторинг состояния всех узлов и сбор полной статистики о системе. Оператору предоставляется возможность контролировать, добавлять, изменять, удалять, запускать и останавливать задачи.
- Модуль управления ресурсами кластера высокой готовности. Он организывает возможность распределения процессов и данных на соседние узлы, миграции процессов с одного узла на другой, а также позволяет следить за их доступностью, восстанавливать функционирование в случае возникновения сбоев.

Разработанное программное обеспечение использует вышеперечисленные встроенные механизмы операционной системы реального времени QNX Neutrino. Все процессы на узлах распределены по группам, и каждая группа имеет свой



Рисунок 3. Структурная схема составной части ПО управления кластером высокой готовности БВК для ОСПВ QNX Neutrino

гарантированный бюджет ресурсов для вычисления, обеспеченный механизмом адаптивного квотирования. Данный механизм гарантирует, что каждая группа процессов получит свой минимум процессорного времени в соответствии с начальными установками. Кроме того, он устраняет возможность монополизации каким-либо процессом или потоком всего доступного процессорного времени. Механизм адаптивного квотирования в ОСРВ QNX Neutrino обеспечивает готовность и защиту ресурсов и гарантирует более высокую степень безопасности и готовности рабочей среды.

С помощью сетевого протокола Qnet организован легкий и понятный интерфейс обмена данными между узлами, дающий доступ к удаленным файлам. Устойчивая служба публикации/подписки позволяет хранить информацию и обмениваться ею в удобном виде, содержащем информацию о состоянии кластера, а также запускать процессы на удаленном узле.

Слаженная работа узлов кластера высокой готовности обеспечивается посредством выбора главного узла в блоке голосования и выбора главного узла (рис. 3). Выбор главного узла инициируется путем голосования, где каждый узел, основываясь на данных об узлах, выбирает главный узел и сравнивает результаты, полученные от остальных. В случае несовпадения голосования происходит перевыбор главного узла. Голосование происходит на основе актуальности информации о состоянии кластера на узлах на момент выбора главного узла. Данные об узлах хранятся в файлах, имеющих имена, идентичные названию узлов, и, используя механизм Qnet, осуществляется передача этих файлов на узлы кластера высокой готовности.

Мониторинг узлов происходит посредством отслеживания состояний каждого из узлов. Важно знать, если какой-то из узлов откажет, а особенно если откажет главный узел, так как он собирает всю информацию о состоянии кластера (информация о процессах, узлах, данных и проблемах). Вся информация хранится в файлах и, используя устойчивую службу публикации/подписки PPS с сетевым протоколом взаимодействия Qnet, предусмотрено быстрое изменение и синхронизация данных на все узлы кластера высокой готовности. Главный узел производит проверку узлов на работоспособность путем использования таймера, который запрашивает актуальную информацию от каждого узла. В случае отсутствия запросов от главного узла происходит проверка его работоспособности. Каждым узлом выполняется сбор статистики о полученных процессах, которая передается главному узлу для формирования текущего снимка состояния

кластера высокой готовности. Данный снимок по окончании формирования передается на ведомые узлы, также используя сетевой протокол взаимодействия Qnet, и позволяет узлам «подхватывать» управление системой в случае выхода из строя главного узла.

Администратор процессов кластера высокой готовности позволяет автоматически контролировать и, по необходимости, в соответствии со сценарием на отказ, восстанавливать работоспособность процессов. Он выполняет роль «умного сторожа», который контролирует работу выбранных программных объектов (процессов). Основная причина разработки собственного администратора процессов, а не использование уже существующего в системе менеджера высокой готовности (High Availability Manager (HAM)), – это отсутствие возможности запуска приложений с заданным приоритетом. Также в менеджере высокой готовности ОСРВ QNX Neutrino нет возможности программно изменять приоритет для многопоточных приложений, что необходимо при функционировании БВК. Для многопоточных приложений тоже нет возможности программно изменять приоритет. В дальнейшем, благодаря возможности детально настроить работу администратора процессов, был создан максимально удобный интерфейс администратора процессов для работы с ним. Администратор процессов также защищен от внутренних сбоев, в случае таковых его работу продолжит с момента завершения преждевременно запущенный дублер. Администратор процессов создан на основе устойчивой службы публикации/подписки PPS и вся информация о процессах хранится в файлах, что позволяет легко и быстро корректировать содержимое информации, а также получать моментальные уведомления об изменениях в файлах. Высокое быстрое действие администратора процессов обеспечивается хранением файлов в ОЗУ во время функционирования PPS.

Блок миграции процессов и данных кластера высокой готовности занимается распределением нагрузки на все узлы кластера на основе данных о весе узла и предпочитаемого узла для процесса. Вес узла высчитывается из веса каждого процесса, который запущен на данном узле. Вес процесса указывается в конфигурационном файле процессов. Миграция процессов осуществляется группами, в случае, если процессы имеют зависимость при работе. Миграция процесса осуществляется путем передачи главным узлом начальных параметров запуска процесса на целевой узел, где администратор процессов, запущенный на целевом узле, запускает на исполнение процесс и занимается его контролем и сбором необходимой статистики о процессе.

Разработка ПО управления кластера высокой готовности БВК велась с использованием кроссплатформенного инструментария разработчика Qt для языка высокого уровня C++, который позволяет создавать программное обеспечение для различных операционных систем без изменения исходного кода программы [4]. Для отладки была использована кроссплатформенная свободная IDE для разработки – Qt Creator, включающая в себя графический интерфейс отладчика и возможность отображения содержимого контейнеров. Была проведена как автономная отладка программного обеспечения, так и комплексная на опытном образце – кластере высокой готовности, состоящем из трех вычислительных модулей. Автономная отладка управляющего программного обеспечения, проводившаяся в кроссплатформенной IDE Qt Creator, показала корректность работы отдельных его модулей. В Qt Creator подключалась вычислительная машина с ОСПВ QNX Neutrino, как устройство для запуска и отладки разрабатываемого программного обеспечения, что позволяло напрямую запускать и отлаживать программный код из Qt Creator, установленного на операционной системе Windows. Тестирование разработанного программного обеспечения на опытном образце с использованием тестовых вариантов процессов подтвердило корректность работы управляющей программы в условиях, близких к реальным.

Выводы

В статье рассмотрены основные аспекты разработки составной части ПО управления кластером высокой готовности БВК под управлением ОС реального времени QNX Neutrino, позволяющей реализовать:

- администрирование процессов с использованием технологии адаптивного квотирования ОСПВ QNX Neutrino. Администрирование процессов обеспечивает отказоустойчивую и сбалансированную работу запущенных процессов с использованием устойчивой службы публикации/подписки (PPS), а также легкий и структурированный способ хранения истории жизненного цикла процесса;
- миграцию процессов и данных на основе протокола сетевого взаимодействия Qnet, предоставляющего доступ к файловой системе любого из узлов кластера и возможность запускать или останавливать процессы;
- выбор главного узла на основе актуальности информации об узле, организованный на устойчивой службе публикации/подписки (PPS);
- мониторинг узлов, собирающий полную статистику о работе кластера высокой готовности и хранящий ее с использованием технологии устойчивой службы публикации/подписки (PPS). Данная статистика позволяет отслеживать текущее состояние и дает возможность узлам «подхватить» управление системой в случае выхода из строя главного узла.

Разработанное ПО обеспечивает работу кластера высокой готовности и предоставляет оператору возможность контролировать, добавлять, изменять, удалять, запускать и останавливать процессы, что является необходимым для БВК.

Результатом работы стал программный продукт, входящий в состав комплекса кластерного программного обеспечения для БВК, прошедший полное тестирование и удовлетворяющий поставленным требованиям.

СПИСОК ЛИТЕРАТУРЫ

1. Проектирование систем бортового информационного обмена и их функциональных элементов: монография / П.П. Парамонов, А.А. Бобцов, Б.В. Видин, И.О. Жаринов, О.О. Жаринов, Ю.И. Сабо, Р.А. Шек-Иовсепянц. Тула: Гриф и К, 2010. 208 с.
2. Операционная система реального времени QNX Neutrino 6.5.0. Системная архитектура. СПб.: БХВ-Петербург, 2014. 400 с.
3. Кртен Р. Введение в QNX Neutrino. Руководство для разработчиков приложений реального времени. СПб.: БХВ-Петербург, 2011. 368 с.
4. Qt Creator. URL: <http://www.qt.io/ide/> (дата обращения: 10.11.2016).

REFERENCES

1. Paramonov P. P., Bobtsov A. A., Vidin B. V., Zharinov I. O., Zharinov O. O., Sabo Yu. I., Shek-lovsepyants R. A. *Proektirovanie sistem bortovogo informatsionnogo obmena i ikh funktsionalnykh elementov: monografiya* [Designing of systems of on-board information exchange and their functional elements: a monograph]. Tula, Grif i K Publ., 2010, 208 p. (In Russian).
2. *Operazionnaya sistema realnogo vremeni QNX Neutrino 6.5.0. Systemnaya arhitektura* [Real-time operating system QNX Neutrino 6.5.0. System architecture]. Saint-Petersburg, BHV-Petersburg Publ., 2014, 400 p. (In Russian).
3. Krten R. *Vvedenie v QNX Neutrino. Rukovodstvo dlya razrabotchikov prilozhenii realnogo vremeni* [Guidelines for real-time application developers]. Saint-Petersburg: BHV-Petersburg Publ., 2011, 368 p. (In Russian).
4. Qt Creator [Электронный ресурс] (In Russ.). Available at: <http://www.qt.io/ide/> (accessed 10.11.2016)

ИНФОРМАЦИЯ ОБ АВТОРАХ

Звонарева Галина Александровна, к.т.н., доцент, Московский авиационный институт, 125993, Москва, А-80, ГСП-3, Волоколамское ш., д. 4, тел.: 8 (916) 220-24-36, e-mail: zvonarev@umail.ru.

Терехов Алексей Анатольевич, начальник отдела, АО «Научно-исследовательский институт «Аргон»», 117587, Москва, Варшавское ш., д. 125, стр. 1, тел.: 8 (910) 475-57-12, 8 (495) 388-54-00, e-mail: javaos@mail.ru.

Федоров Александр Андреевич, магистрант Московского авиационного института, инженер-программист 1-й категории, АО «Научно-исследовательский институт «Аргон»», 117587, Москва, Варшавское ш., д. 125, стр. 1, тел.: 8 (903) 182-97-60, e-mail: pabl2010@yandex.ru.

AUTHORS

Zvonareva Galina, PhD, Docent, Moscow Aviation Institute (MAI), 4, Volokolamskoe sh., A-80, GSP-3, Moscow, 125993, Russian Federation, tel.: +7 (916) 220-24-36, e-mail: zvonarev@umail.ru.

Terekhov Aleksey, head of department, Science Research Institute «Argon», 125, Varshavskoe sh., Moscow, 117587, Russian Federation, tel.: +7 (910) 475-57-12, +7 (495) 388-54-00, e-mail: javaos@mail.ru.

Fedorov Aleksandr, magistrant, Moscow Aviation Institute, software engineer I, Science Research Institute «Argon», 125, Varshavskoe sh., Moscow, 117587, Russian Federation, tel.: +7 (903) 182-97-60, e-mail: pabl2010@yandex.ru.