позволяет для каждой сущности в случае отключения или выхода из строя узла с основной копией выбрать другой узел с копией, который будет использоваться в качестве основного для данной сущности. В совокупности с равномерным распределением основных и неосновных копий по узлам, в случае отключения одной из вершин, произойдет автоматическое перераспределение ролей копий и работа всей системы продолжится. Отдельным случаем стоит рассмотреть вариант при разделении сети на 2 части, где связь в каждой половины будет работать, но соединение между вершинами двух половин будет утеряно. Для того, чтобы у нас не было параллельного функционирования двух основных копий для некоторых сущностей, можно добавить условие, что копия может являться основной лишь при наличии связи с большей половиной узлов, содержащих копию данной сущности.

Таким образом, была предложена модель системы для обеспечения распределенного хранения данных. Выбранная структура данных позволила обеспечить возможность распределенного хранения данных, а также увеличить надежность системы. В качестве базы для обеспечения доступности и надежности использован принцип реплицируемости данных. А использование метода основной копии, хеш-функции для определения вершин для хранения сущностей и алгоритма поведения системы в случае потери связи между вершинами позволили добиться согласованности, доступности, распределения нагрузки и надежности.

Список использованных источников:

1. Mustaque Ahamad , Mostafa H. Ammar , Shun Yan Cheung - "Replicated Data Management in Distributed Systems" - Emory University, 1992

2. Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, Werner Vogels - "Dynamo: Amazon's Highly Available Key-value Store" - Amazon.com, 2007

МЕТРИКИ ОЦЕНКИ ФУНКЦИОНИРОВАНИЯ ПРИЛОЖЕНИЙ ПОД УПРАВЛЕНИЕМ JAVA EE СЕРВЕРОВ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Бродницкий В.В.

Куликов С. С. – к. т. н., доцент

Мониторинг состояния и производительности приложений является критически важной составляющей при поддержке работы приложений, и чем больше возрастает сложность и масштаб системы, тем большую важность приобретает анализ её работы для сохранения стабильности функционирования и целостности данных. Большинство Java-приложений являются не одиночными программными средствами, а состоят из множества различных компонентов, обладают масштабируемой структурой, что значительно увеличивает зависимость состояния системы от ряда внешних факторов.

Количество показателей и метрик работы приложения, доступных для мониторинга, зависит от типа анализируемого программного средства, типа его компонентов и использованных сервисов, а также от конкретной реализации виртуальной машины JAVA. Важной задачей является группировка их по критерию описываемой области, что позволит увеличить эффективность их выборки и анализа. Можно выделить несколько ключевых областей, характеризующих стабильность и производительность любой программной системы:

Показатели виртуальной машины. В данную группу выделяют частоту событий виртуальной машины (количество вызовов и откликов методов, ошибок), степень параллелизма потоков, информация о блокировках, данные о компиляции.

Показатели загруженности памяти. Основными критериями данной группы являются показатели отношения свободной и занятой памяти к общему её количеству. Однако важными показателями являются также сообщения менеджеров памяти о превышении пороговых значений загруженности памяти для определённых пулов и необходимости её увеличения.

Показатели сборщика мусора. Для платформ с автоматическим управлением памятью сборка мусора является ключевым моментом, от эффективности работы которого зависит общая производительность системы. Здесь следует выделить показатели количества произведённых сборок, освобождённой в них памяти, подробные данные о состоянии поколений.

Показатели операционной системы. Так как приложения, выполняемые виртуальной машиной, располагаются на конкретном аппаратном и программном обеспечении, то характеристики их эффективности и готовности непосредственно влияют на показатели производительности самой программной системы. Очевидно, что мониторинг их состояния является важной частью диагностики любого программного средства. В основном здесь выделяют уровень загрузки процессора, доступность и скорость передачи по сети, операций ввода-вывода информации.

Виртуальная машина Java представляет собой сложную многоуровневую систему, включающую в себя множество компонентов и крайне тяжело выделить единую группу показателей, полностью характеризующую её производительность и надёжность. В 5 версии был добавлен специальный пакет объектов и интерфейсов, который позволяет выделить определённые группы метрик, сгруппированные по описанию определённых подсистем виртуальной машины. Мониторинг

производится за счёт вызовов методов МХ-объектов, реализующих предоставленные интерфейсы. Пользователь не имеет прямого доступа к их жизненному циклу, инициализацией, контролем и уничтожением данных объектов занимается сама виртуальная машина.

Интерфейс системы выполнения обеспечивает доступ к большому списку информации, описывающей работу виртуальной машины по обеспечению выполнения приложений. Среди них выделяют:

- Пути расположения данных: расположение классов системы, сторонних библиотек, загрузчика классов и системных свойств инициализации виртуальной машины, которые используются запущенными приложениями.
- Информация виртуальной машины: содержит наименование, версию, имя поставщика текущей реализации виртуальной машины, а также аналогичную информацию о её спецификации.
- Время запуска: предоставляет время запуска виртуальной машины, приостановок и возобновлений работы в миллисекундах.
- Информация о текущем потоке: содержит уникальный идентификатор текущего потока, процессорное время, затраченное на его работу, время, которое поток находился в режиме ожидания и его текущий статус.
- Количество активных потоков: общее количество незавершённых потоков виртуальной машины, включая потоки-демоны.
- Количество потоков в режиме ожидания: включает также общее время их простоя и процессорное время, затраченное на их выполнение. Как только процесс переходит в активный статус, счётчик времени ожидания сбрасывается.

Таким образом мониторинг данных видов загрузки позволяет составить цельную картину работы программных средств и во многих случаях диагностировать способы оптимизации производительности выполнения приложений в рамках виртуальной машины.

Список использованных источников:

1. Alistair Croll Complete Web Monitoring/ Alistair Croll, Sean Power: O'Reilly Media, 2009.
2. D. Jones Creating Unified IT Monitoring and Management in Your Environment/ D. Jones: Realtime Publishers, 2012.
3. EMC Smarts: интеллектуальный мониторинг ИТ – среды и бизнес – процессов / Комптек: Москва.

АВТОМАТИЗИРОВАННЫЕ МЕТОДЫ ОБРАБОТКИ ПСИХОЛОГИЧЕСКИХ ТЕСТОВ С ВОПРОСАМИ ОТКРЫТОГО ТИПА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Быковский А. С.

Серебряная Л. В. – кандидат техн. наук, доцент

Условием успешного проведения психодиагностики является необходимость принятия решений в сфере измерения индивидуально-психологических особенностей человека. Основным источником информации, отражающей указанные особенности, является психологический тест и его последующая обработка.

Основной целью данной работы является разработка автоматизированных методов обработки психологических тестов с вопросами открытого типа. Методы должны давать правильные результаты при обработке нечисловых данных и быть применимы при разработке программного обеспечения. Методы должны охватывать стандартизированные методики и анкетные опросы, а так же способствовать уменьшению количества времени, необходимого для выведения психологического заключения.

Исходя из этих требований и особенностей интерпретации ответов на вопросы, было выделено три метода обработки психодиагностической информации на компьютере: автоматический, полуавтоматический и ручной. Автоматический метод хорошо подходит для закрытых вопросов, так как выбор ответа не связан с непосредственным его вводом в свободной форме и отсутствует фактор внесения непреднамеренной ошибки в ответ. В процессе автоматической обработки принимается однозначное решение о правильности ответа. После автоматической обработки будет получено психологическое заключение с указанием вычисленных выходных параметров теста. Так же данный метод подходит для открытых вопросов.

Полуавтоматический метод применяется, если в вопросе, для которого есть правильное решение, предлагается свободный ответ. Это связано с вводом текстовой информации и существует вероятность ввода правильного ответа с орфографической ошибкой, или опечаткой, а они не должны влиять на результат психологического теста. В данном варианте программная система сопоставляет ответ с эталонным, вычисляя процент соответствия. Такой процент по каждому спорному вопросу выдается, психологу при обработке результатов и предлагается указать: является ли данный ответ верным. Если между ответом респондента и эталоном найдено полное соответствие, решение о правильности ответа система принимает автоматически.

Для анкет, вопросы которых предлагают ответ-описание или ответ, выражающий отношение,