

Нагрузочные модули для высокопроизводительных систем

М.Ю. Юрич, Р.К. Кудерметов

Запорожский национальный технический университет, г.Запорожье, Украина

Введение

Как известно, существуют некоторые общепринятые этапы создания программно-го обеспечения (ПО), такие как анализ, проектирование, реализация, тестирование, внедрение, сопровождение. Каждый из них является очень важным для полноценного и правильного функционирования разрабатываемого ПО. Остановимся более подробно на одном из таких этапов – этапе тестирования. Для разных видов ПО существуют различные признаки, по которым принято проводить классификацию тестирования.

Одним из пунктов в классификации по объекту тестирования стоит нагрузочное тестирование, включающее в себя:

- тестирования производительности (performance/stress testing);
- тестирования стабильности (stability/load testing) [1].

По данным, представленным в [2], нагрузочное тестирование применяется для анализа работы информационных систем на различных уровнях нагрузки.

В Performance Lab различают три источника нагрузки характерные для большинства IT-систем:

1. Работа пользователей системы (Workload);
2. Взаимодействие с внешними системами;
3. Автоматические и пакетные серверные процессы (Batch Processes).

В ходе испытаний эмулируются все три типа нагрузки.

Нагрузочное тестирование на данном этапе развития IT-технологий в основном применяют для web-серверов. Обычными результатами нагрузочного тестирования считаются:

1. Показатели производительности на различных уровнях нагрузки (времена отклика, число операций в единицу времени, использование системных ресурсов, др.);
2. Количественные характеристики максимально допустимой нагрузки и предельной производительности системы;
3. Описание "узких мест" (bottlenecks) - факторов, которые препятствуют выполнению требований производительности на высоких нагрузках [2].

Другими словами, основной целью нагрузочного тестирования является проверка реальных возможностей ПО, разработанного для web-серверов.

Необходимость и важность использования нагрузочного тестирования продемонстрирована «в цифрах» в докладе П. Липского «Мифы о нагрузочном тестировании» на Конференции разработчиков высоконагруженных систем (2008 г.) , где говорится, что по исследованию, проведенному компанией Google в 2006 г., при увеличении времени загрузки страницы на 0,4 секунды трафик и рекламные доходы сокращаются на 20% [3].

Таким образом, нагрузочное тестирование и анализ производительности – это совокупность методов, направленных на выявление соответствия характеристик производительности тестируемой системы заявленным и их анализ [4]. Множество литературы посвящено актуальности, необходимости и непосредственному описанию нагрузочного тестирования, применяемого для web-серверов. Однако необходимость в тестировании присутствует не только для данной сферы.

Нагрузочное тестирование высоконагруженных систем

Широко известны проблемы, возникающие при необходимости выполнения объемных вычислений (т.е. объемных по времени выполнения и/или требующих больших объемов памяти, и/или требующих больших процессорных ресурсов, и т.п.). Для решения такого рода задач, как известно, используют различные «компьютерные объединения»: кластеры, локальные компьютерные сети, распределенные вычислительные сети или их различные комбинации. К числу таких представителей «компьютерных объединений» можно также отнести и систему GRID.

Однако, существование подобных вычислительных систем не гарантирует быстрого решения существующей объемной вычислительной задачи или потока задач. И это, прежде всего, связано с необходимостью правильного и рационального управления существующей системой, которое, в первую очередь, связано с планированием нагрузки вычислительной системы.

Для эффективного планирования нагрузки используются различные балансировщики/планировщики нагрузки (load balancer) [5, 6]. Тем не менее, не смотря на множество существующих планировщиков нагрузки, постоянно возникают попытки разработки новых алгоритмов и новых программных продуктов, позволяющих более эффективно распределять нагрузку между узлами вычислительной сети. И выяснить, какой из планировщиков лучше подходит для использования в какой-либо определенной системе, является не простой задачей. В основном выбор решается в пользу дешевизны продукта или навыками работы с ним, либо же просто, опираясь на теоретические описания работы того или иного планировщика. Все эти и подобные подходы к выбору планировщиков нагрузки, к сожалению, являются не объективными, а потому и мало эффективными. Естественно, какой бы планировщик не был использован в системе, он повысит эффективность решения задач в системе. Однако, этот показатель можно повысить на порядки, точно зная, что какой-то из планировщиков будет более эффективнее работать в определенной системе и использовать именно его.

Следовательно, возникает вопрос, как же определить, какой из планировщиков выбрать для более эффективного использования ресурсов определенной вычислительной системы? К сожалению, ответ не может быть найден сразу. Для этого необходимо провести некую процедуру. Для того, чтобы сделать правильный выбор в такой ситуации, предлагаем провести нагрузочное тестирование планировщиков нагрузки, результатом которого станут не просто различные показатели работоспособности и производительности исследуемых продуктов, а показатели, по которым можно будет оценить преимущества или недостатки определенного программного продукта перед аналогами.

На данном этапе развития информационных технологий применение нагрузочного тестирования в таком направлении не рассматривается.

Нагрузочные модули

Для проведения нагрузочного тестирования необходимо разрабатывать специальные программные модули, запуск которых на предоставленных аппаратных средствах с использованием определенных тестируемых программных средств отображает адекватную реальности ситуацию. Т.е. с применением таких модулей появляется возможность отследить реакцию тестируемых программных продуктов на возможно возникающие при реальной работе ситуации.

Для реализации описанных выше функций нагрузочные модули должны оказывать характерное влияние на тестируемые программные продукты, т.е. в зависимости от их (продуктов) назначения будут различные характерные воздействия. И это, конечно же, непосредственно связано с самими задачами, поступающими для решения какой-то определенной вычислительной системы. Таким образом, в случае тестирования планировщиков нагрузки необходимо создать тесты, нагрузочные модули, которые

реализуют различные виды вычислительных задач. Следовательно, возникает необходимость в классификации поступающих для решения на вычислительной системе задач.

Как уже было отмечено ранее, от наиболее удачной классификации поступающих для решения в вычислительной системе задач зависит адекватность используемых тестов, а следовательно, и правильность выбора необходимого программного продукта. Существуют классификации задач по различным признакам. Однако, для рассматриваемого нами случая необходимо классифицировать задачи особым образом. Так, предположим, что в данном случае наиболее эффективной будет классификация по занимаемым задачами ресурсам. Тогда, к примеру, возможным вариантом классификации будет разделение задач на следующие группы:

1. Требующие большие затраты оперативной памяти;
2. Требующие большие затраты процессорного времени;
3. Требующие высокой пропускной способности сети.

Естественно, в ходе исследований этот список может быть продолжен, либо, в случае неэффективности предложенного подхода, вообще изменен. Но в любом случае, после этапа классификации задач необходима реализация конкретных вычислительных задач, попадающих под эту классификацию, на основе которых и будут разработаны нагрузочные модули.

Выводы

Таким образом, в данной работе предложено использование нагрузочного тестирования в виде специальных нагрузочных модулей для оптимального выбора планировщика распределения задач (балансировщика нагрузки) в вычислительных системах. И одной из основных задач, которая при этом возникает, является наиболее подходящая классификация поступающих на выполнение в вычислительной системе задач.

Разработка нагрузочных модулей позволит не только выбрать наиболее подходящий балансировщик нагрузки для определенной системы, но и в будущем даст возможность разрабатывать новые, более эффективные планировщики распределения заданий на основе выявленных недостатков в предыдущих версиях.

Литература

1. Свободная энциклопедия – Википедия. Режим доступа: http://ru.wikipedia.org/wiki/Тестирование_программного_обеспечения.
2. Официальный сайт сервис-провайдера в области обеспечения производительности ИТ Performance-Lab. Режим доступа: <http://performance-lab.ru/load-testing.php>
3. П. Липский. Мифы о нагрузочном тестировании. / Конференция разработчиков высоконагруженных систем HighLoad++ 2008. Режим доступа: <http://highload.ru/papers2008/7149.html>
4. В. Банад. Методология нагрузочного тестирования: лучшие практики. / Конференция SQA Days 2008. Режим доступа: <http://www.slideboom.com/>
5. gLite: новое поколение промежуточного программного обеспечения для гридов в рамках EGEE. Режим доступа: http://egee.pnpi.nw.ru/doc/gLite_ru.pdf
6. Condor High Throughput Computing. Режим доступа: <http://www.cs.wisc.edu/condor>