

Эта часть работы выложена в ознакомительных целях. Если вы хотите получить работу полностью, то приобретите ее воспользовавшись формой заказа на странице с готовой работой:

<https://studservis.ru/gotovye-raboty/kurovaya-rabota/279650>

Тип работы: Курсовая работа

Предмет: Программирование (другое)

Введение 3

Глава 1. Понятия тестирования и отладки программного обеспечения 7

Глава 2. Стратегия тестирования и отладки программного обеспечения 18

Заключение 35

Библиография 37

Приложение 1. Глоссарий 39

Управление производительностью разработки программного обеспечения является ключевым вопросом в организациях, занимающихся разработкой программного обеспечения. Требования бизнеса по сокращению времени выхода на рынок при сохранении высокого качества продукции заставляют организации, занимающиеся разработкой программного обеспечения, искать новые стратегии повышения производительности разработки.

Современные системы, как программно-алгоритмическое, так и аппаратное обеспечение, часто состоят из подсистем от разных производителей. Установить причинно-следственную связь между системами сложно. Тестирование программного обеспечения лежит в основе получения качественного программного продукта. Если компания-производитель ПО не в силах в достаточной мере и тщательно проверить свои продукты до того, как ими воспользуются заказчики, есть риск получить ПО с ошибками, который не будут оплачивать заказчики.

Тестирование производительности программного обеспечения в зависимости от их назначения учитывает, прежде всего, функционал приложения и требования, предъявляемые к нему. Также при тестировании следует учитывать тип операционной системы, для которой создано приложение. Производительность программного обеспечения (рассматриваемая здесь как связанная с производительностью и своевременностью) - это всепроникающее качество, которое трудно понять, поскольку на него влияет каждый аспект проектирования, кода и среды выполнения. По общепринятому мнению, производительность является серьезной проблемой в значительной части проектов.

Она вызывает задержки, перерасход средств, сбои при внедрении и даже отказ от проектов, но такие сбои редко документируются. Недавний опрос руководителей компаний, занимающихся информационными технологиями, показал, что половина из них сталкивались с проблемами производительности по крайней мере 20% приложений, которые они развернули.

Обзор литературы показывает, что в основном, исследования по тестированию программного обеспечения направлены на особенности тестирования веб-приложений. В работах авторов [1,2] рассматриваются именно основные аспекты тестирования таких приложений. В работе автора [3] рассмотрены аспекты тестирования производительности высокопроизводительного приложения с обработкой больших данных на основе подхода BOINC-грид. В данной работе проводился эксперимент, который показал в программном обеспечении масштабируемость, надежность, потребление ресурсов [3]. Также при тестировании в данном эксперименте использованы интеллектуальные алгоритмы обработки данных.

Так же как существует жизненный цикл разработки программного обеспечения для создания цифровых продуктов, существует жизненный цикл тестирования программного обеспечения для их проверки. Различные члены организации участвуют в этом процессе на разных этапах, чтобы помочь бизнесу достичь своих целей в области качества стратегическим, документированным образом.

Эволюцию разработки программно-алгоритмического обеспечения можно проследить с историей тестирования программных продуктов.

Как мы можем заметить, исторически, разработка программно-алгоритмического обеспечения была направлена на выполнение заказов военных - Министерства обороны. В таких программах особое внимание уделялось масштабным базам данных, проектирование таких программ проектировалось на базе универсальных электронно-вычислительных машин.

Все сценарии и алгоритмы тестирования фиксировались на бумаге. При помощи этих сценариев проверялись целевые потоки управления, вычисления сложных алгоритмов и манипулирование данными.

Результирующий набор тестовых процедур мог эффективно протестировать всю систему полностью. Как правило, тестирование начиналось после завершения плана-графика проекта и выполнялось персоналом разработчиков, которые начинали проект.

Возможность стандартизации процессов тестирования и разработки программного обеспечения возникла с появлением персональных компьютеров

Появление персональных компьютеров открыло новую эру и привело к быстрому и бурному росту коммерческих разработок. Так как на рынке возникла жесткая конкуренция среди разработчиков программного обеспечения, к нему стали предъявляться также жесткие требования по функционированию и распространению. Тем самым восприятие пользователей к наиболее распространенным программным продуктам и их наполнению стали восприниматься как стандарты «де факто» к программному обеспечению. С точки зрения технологий, пакетная обработка заменялась системами, работающими в реальном времени.

Тестирование систем реального времени требовало иного подхода к проектированию и тестированию. Связанно это было с тем, что рабочие потоки могли быть вызваны в любом порядке. Это явление привело к появлению большого количества процедур тестирования, способных поддерживать бесконечное число перестановок, сочетаний и комбинаций.

Стоит отметить, что причинами многих неудач компаний – разработчиков программно-алгоритмического обеспечения являлись ошибки в их программах. Ошибки в программах дорого стоили компаниям – разработчикам, поскольку такие программы заказчики не намерены оплачивать и заказывать. Тогда как сами разработчики – наемные сотрудники из-за ошибок в программах могли потерять работу.

Исторически, ошибки в программном обеспечении рассматривались как неточности и им не уделялось достаточно внимания. Также есть примеры компаний – разработчиков, которые были вынуждены закрыться из-за их некорректно работающего ПО.

Со всеобщей цифровизацией и компьютеризацией во всех областях и сферах деятельности, прежде всего таких, как управление системами жизнеобеспечения, медицина, дорогую компьютерную аппаратуру, ошибки в программном обеспечении уже определялись как значимые на всех этапах разработки и тестирования.

Наиболее яркими примерами ошибок программного обеспечения являются западные компании, где некорректно работающее программное обеспечение привело к катастрофическим последствиям. Например, один из сравнительно недавних сбоев в работе программного обеспечения, подтвержденный самой компанией Yahoo 22 сентября 2016 года. Yahoo заявила, что из сети организации была украдена копия информации об учетных записях конкретных пользователей. Информация об учетной записи могла включать адреса электронной почты, имена, даты рождения, номера телефонов, хэшированные пароли и, в большинстве случаев, незашифрованные и зашифрованные вопросы и ответы безопасности. Расследование показало, что украденная информация не содержала данных платежных карт, незащищенных паролей и информации о банковских счетах. Yahoo считает, что была украдена информация, связанная примерно с 500 миллионами учетных записей пользователей.

Yahoo призвала своих пользователей изменить секретные вопросы и ответы, а также пароли от любых других учетных записей, если в них использовалась аналогичная информация. Компания напомнила своим пользователям, чтобы они не переходили по ссылкам и не загружали вложения из подозрительных писем. Компания также предупредила пользователей о необходимости проверять свои учетные записи на предмет подозрительной активности и не сообщать свою личную информацию при запросе на веб-странице. Данный случай говорит не только об уязвимости кибербезопасности, а о некорректно работающем ПО.

Устраняя неполадки в системе планирования вакцинации в больницах Нью-Джерси, одна из больниц обнаружила от 10 до 11 тысяч дубликатов назначений. Команда из 200 человек сделала более 10 400 телефонных звонков, чтобы устранить проблему, а затем определила, что около 70% назначений были дубликатами. Большинство людей, пострадавших от сбоя, были пожилыми, как сообщил сотрудникам больницы представитель компании, занимавшийся тестированием и устранением неполадок ПО.

В мае 2015 года компания Airbus испугалась, когда сообщила о программной ошибке в своем самолете A400M. Компания немедленно приняла меры по устранению проблемы, которая привела к смертельной аварии, произошедшей в Испании. До этих действий во время испытательного полета в Севилье уже сообщалось о травмировании двух человек, а четыре члена экипажа BBC погибли в этом инциденте.

В целом, можно отметить, что тестирование занимает не менее 50% времени всей разработки, если модульное тестирование начинается вовремя, то можно устранить практически все уязвимости ПО и сократить время разработки.

Очень трудно выявить источник ошибки. В этом заключается основная проблема тестирования. Поскольку ни один из модулей не проверен как следует, в большинстве из них есть ошибки. Получается, что вопрос не столько в том, в каком модуле произошла обнаруженная ошибка, сколько в том, какая из ошибок во всех вовлеченных в процесс модулях привела к полученному результату. И когда накладываются ошибки нескольких модулей, ситуацию может быть гораздо труднее локализовать и повторить.

В работе рассматриваются основные аспекты тестирования программного обеспечения, работа состоит из двух глав, введения, заключения. В приложении приведены определения терминов, используемых для описания тестирования ПО.

Глава 1. Понятия тестирования и отладки программного обеспечения

1.1 Принципы тестирования и отладка программного обеспечения

В данной главе рассмотрим основные аспекты тестирования программного обеспечения, а также тестирования производительности ПО. Производительность можно охарактеризовать как способность обрабатывать программой большой объем входных данных за короткий промежуток времени. Тестирование программного обеспечения — это процесс анализа или эксплуатации программного обеспечения с целью выявления ошибок. Измерения используются группами проверки для обеспечения того, чтобы тестируемая система соответствовала своим спецификациям, для моделисты производительности для построения и проверки моделей, и проектировщиками для поиска и устранения "горячих точек" в коде.

Интерес к измерению производительности компьютерной системы восходит к разработке самых первых систем. Сегодня же состояние промышленных методов измерения производительности и тестирования отражено в серии статей зарубежных исследователей, включая проблемы планирования, выполнения, инструментария и интерпретации. Российские исследователи делают упор именно на вопросах тестирования веб-приложений.

При разработке тестов производительности важным вопросом является определение рабочей нагрузки, под которой тестирование выполняется. Один из подходов заключается в проведении эксплуатационных испытаний в условиях, аналогичных ожидаемым условиям эксплуатации профилем приложения в полевых условиях.

Тестирование производительности - это процесс тестирования программного обеспечения, используемый для проверки скорости, времени отклика, стабильности, надежности, масштабируемости и использования ресурсов программного приложения при определенной рабочей нагрузке. Основной целью тестирования производительности является выявление и устранение узких мест в работе программного приложения. Это подмножество инженерии производительности и также известно как "Perf Testing".

Основное внимание при тестировании производительности уделяется проверке следующих параметров программного обеспечения:

1. Темичев А.А., Файзрахманов Р.А. Аналитический обзор средств автоматизации тестирования производительности применительно к системам мониторинга // Вестник ПНИПУ. Электротехника, информационные технологии, системы управления. 2015. №15. URL: <https://cyberleninka.ru/article/n/analiticheskiy-obzor-sredstv-avtomatizatsii-testirovaniya-proizvoditelnosti-primenitelno-k-sistemam-monitoringa> (дата обращения: 02.07.2022).
2. Киселев, Павел Владимирович. "Прогрессивные веб-приложения: объединяющая технология для веб-и нативных приложений." Политехнический молодежный журнал 2 (2020): 43.
3. Ивашко Евгений Евгеньевич, Ицкарь Мария Сергеевна, Колчина Мария Владимировна Тестирование производительности приложения по анализу данных на базе BOINC-грид // Ученые записки Петрозаводского государственного университета. 2014. №4 (141). URL: <https://cyberleninka.ru/article/n/testirovanie-proizvoditelnosti-prilozheniya-po-analizu-dannyh-na-baze-boinc-grid> (дата обращения: 04.07.2022).
4. Основы алгоритмизации и программирования: учеб. пособие / Т.А. Жданова, Ю.С. Бузыкова. – Хабаровск : Изд-во Тихоокеан. гос.ун-та, 2011. – 56 с. Режим доступа: http://pnu.edu.ru/media/filer_public/2013/02/25/book_basics.pdf.
5. Программирование и основы алгоритмизации: Для инженерных специальностей технических университетов и вузов. /А.Г. Аузяк, Ю.А. Богомолов, А.И. Маликов, Б.А. Старостин. Казань: Изд-во Казанского национального исследовательского технического ун-та - КАИ, 2013, 153 с. Режим доступа: http://au.kai.ru/documents/Auzyak_Progr_osn_alg_C_2013.pdf.
6. Основы алгоритмизации и программирования : учебное пособие / Г. Р. Кадырова. – Ульяновск : УлГТУ, 2014.

7. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем [текст] / Б. Бейзер; - Питер, 2004, 320 с. ISBN 5-94723-698-2.
8. Брауде Э.Д. Технология разработки программного обеспечения [текст] / Э.Д. Брауде; - Питер, 2004, 656 с. ISBN 5-94723-663-Х.
9. Винниченко И.В. Автоматизация процессов тестирования [текст] / И. В. Винниченко; - Питер, 2005, 208 с. ISBN 5-469-00798-7.
10. Канер С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений [текст] / С. Канер; - ДиаСофт, 2001, 544 с, ISBN 966-7393-87-9.
11. Калбертсон Р. Быстрое тестирование [текст] / Р. Калбертсон, К. Браун, Г. Кобб; - Вильямс, 2002, 384 с. ISBN 5-8459-0336-Х.
12. Коликова Т.В. Основы тестирования программного обеспечения. Учебное пособие [текст] / Т.В. Коликова, В.П. Котляров; - Интуит, 2006, - 285 с. ISBN 5-85582-186-2.
13. Касперски К. Техника отладки программ без исходных текстов [текст] / К. Касперски; - БХВ-Петербург, 2005, 832 с. ISBN 5-94157-229-8.
14. Макгрегор Д. Тестирование объектно-ориентированного программного обеспечения. Практическое пособие [текст] / Д. Макгрегор, Д. Сайкс; - ТИД «ДС», 2004, 432 с. ISBN 966-7992-12-8.
15. Пласкин М. Тестирование и отладка программ - для профессионалов будущих и настоящих [текст] / М. Пласкин; - Бином. Лаборатория знаний, 2007, - 168 с. ISBN 978-5-94774-458-3.
16. Роберт М. Быстрая разработка программ: принципы, примеры, практика [текст] / М. Роберт, Д. Ньюкирк; - Вильямс, 2004, 752 с. ISBN 5-8459-0558-3.
17. Фолк Д. Тестирование программного обеспечения [текст] / Д. Фолк, Е. К. Нгуен, С. Канер; - Диасофт, 2003, 400 с. ISBN 966-7393-87-9.
18. Элфрид Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация [текст] / Элфрид Д., Джефф Р., Джон П.;- Лори, 2003, ISBN 5-85582-186-2.
19. Petersen, Kai. "Measuring and predicting software productivity: A systematic map and review." Information and Software Technology 53, no. 4 (2011): 317-343.

Эта часть работы выложена в ознакомительных целях. Если вы хотите получить работу полностью, то приобретите ее воспользовавшись формой заказа на странице с готовой работой:

<https://studservis.ru/gotovye-raboty/kurosovaya-rabota/279650>