

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ

ТЕХНОЛОГИИ

Секция „Йерархични системи”

СТАНИСЛАВ ДИМИТРОВ ДИМИТРОВ

ПРИЛОЖЕНИЕ НА ЗАКОНИТЕ НА ТЕОРИЯТА НА

УПРАВЛЕНИЕТО В ПРОГРАМНИ СИСТЕМИ

АВТОРЕФЕРАТ

на дисертация

за присъждане на образователна и научна степен „доктор”

по специалност 02.21.10 „Приложение на принципите и методите на кибернетиката в

различни области на науката”

по научно направление 5.2 „Електротехника, електроника и автоматика”

Научен ръководител:
Проф.д.т.н. Тодор Стоилов

СОФИЯ
2016

Дисертационният труд е обсъден и допуснат до защита на разширено заседание на секция „Йерархични системи“ на 9.12.2016г. в Института по информационни и комуникационни технологии на БАН.

Дисертационният труд обхваща 160 страници, в него са включени 72 фигури, 9 таблици и са посочени 95 литературни източника.

Защитата на дисертационния труд ще се състои на от часа в залана блок 2 на Института по информационни и комуникационни технологии на БАН на открито заседание на научно жури в състав:

1. проф. д-р Гриша Спасов
2. проф. д-р Коста Бошнаков
3. проф. д-р Идилия Бачкова
4. доц. д.т.н Даниела Борисова
5. проф. д.т.н Тодор Стоилов

Материалите за защитата на разположение в стая на Института по информационни и комуникационни технологии на БАН, ул „Акад Г.Бончев“ , бл 2.

Обща характеристика на дисертационния труд

Актуалност и значимост на изследването

Широкото използване на различни компютърни конфигурации за достъп до разнообразни уеб сайтове, предлагащи уеб услуги, генерира изключително натоварен мрежови трафик в устройствата, свързващи клиентите и доставчиците на услуги. Това на свой ред натоварва компонентите, изграждащи машината, на която се поддържат интернет-страниците. Сървърът е система, в която протичат много задачи едновременно и неговото обслужване представлява сложен процес на управление.

Разбираемо, подобно интензивно използване на компютърните и комуникационни системи причинява липса на гарантирано време за обслужване на поставените им задачи и прави статичните системи все по-неприложими. Това изисква да се проектират и прилагат техники, които позволяват динамика и гъвкавост по отношение на промени в натоварването и използването на ресурсите.

В този контекст подходи, основани на теория на автоматичното управление за моделиране, анализиране и проектиране на вградени компютри и комуникационни системи, се превръщат в обещаваща платформа за контролиране на променливостта на големи и сложни системи, работещи в реално време. Системите с обратна връзка стават чест обект на изследване с оглед на проблеми като адаптиране на изчисления в реално време, гарантиране на производителността при възникнали смущения и ефективно планиране на ресурсите. В тази връзка, модели от теорията на автоматичното управление се прилагат за описание на поведението, за контролиране на времето за реакция, както и за повлияване оптималността, устойчивостта и адаптацията на управлението на реални системи.

Всичко това задава актуалността и значимостта на дисертационния труд, който се съсредоточава именно върху приложимостта на принципи от теория на автоматичното управление по отношение на програмни системи. Последното представлява обект на изследване в този текст. Съответно предмет на дисертационния труд е управлението на компютърни системи като сложни системи, следвайки постулатите на теория на автоматичното управление, с цел подобряване на тяхната производителност.

В рамките на тази конкретна теоретична перспектива, особен интерес за настоящото изследване представлява връзката между зададените задачи от потребителите и използваемостта на хардуерните компоненти.

Цели и задачи на дисертационния труд

Целта на изследването в дисертационния труд е да се разработят и приложат модели за управление от теория на управлението при функционирането на сложни програмни системи.

За постигането на целта на дисертационната работа се дефинират и решават следните задачи:

- ✓ Да се разработи функционален модел на компютърната система при различни входни въздействия;
- ✓ Да се реализира управление в изследваната компютърна система и да се синтезира управление чрез настройка на TCP протокола, уеб сървъра Apache и операционната система;

- ✓ Да се приложат методи за идентификация на системи и да се определят динамичните характеристики на компютърната система при управление на програмата за уеб сървър;
- ✓ Да се изследва ефективността на приложените настройки на управлението на системата чрез показателите за качество;
- ✓ Да се синтезират регулатори за управление на изследваната система и да се оценят качествата на затворена система за управление.

Методология на изследването

Работната хипотеза на дисертационното изследване е, че третирането на компютърни системи като обект на управление в съответствие с принципите на теория на автоматичното управление, ще доведе до по-добра производителност и съответно до по-високо качество на обслужване на потребителите.

В хода на проверката на тази хипотеза, са проведени експерименти с конкретнософтуерно приложение за обслужване на потребители – платформата на списание „Автоматика и информатика“. Чрез симулации, касаещи промени в конфигурационния файл и настройки, са изследвани показателите за производителност – време за отговор, латентност и бързодействие по отношение на потреблението на различни видове файлове (текст, аудио, видео и база данни).

Така, в изпълнение на горепоставените цел и задачи, са използвани основни положения от теория на автоматичното управление (ТАУ), реализирани в следната **методология**:

- Генериране на входящи въздействия чрез симулиране на потребители на информационна система с цел получаване на изходящи сигнали за работата на системата;
- Прилагане на методи за идентификация с цел събиране на данни за работата на програмната система;
- Използване на статистически методи за анализ на измерените изходни данни;
- Прилагане на управление с обратна връзка на програмна система с цел проверка на възможностите за подобряване обслужването на потребители на софтуерното приложение;
- Прилагане метода на контролните карти при измерване качеството на показателите на производителността на програмната система;
- Прилагане на нелинеен модел за описание поведението на системата при подобряване на обслужването на потребителите на софтуерното приложение.

В хода на **експерименталната работа** са използвани:

- Генератори на входни въздействия Jmeter и Apache Bench;
- Инструмент SAR за наблюдение на операционната система Linux;
- MATLAB – за статистически анализ на данни и извеждане на динамични модели, за реализиране на симулация на нелинейни модели и за настройка на регулаторите на компютърната система;
- Linux програмата PS и език за програмиране AWK за реализиране на управлението за обратна връзка.

Информационната система е реализирана чрез софтуерни продукти от тип “open source”, като например програмен език PHP, MySQL.

Съдържание на дисертационния труд

Глава 1. Програмни системи и теория на автоматичното управление

Първа глава на дисертационния труд представя основни формализми от теория на автоматичното управление, които са използвани за основа на експерименталната работа, и предлага обзор на програмни системи, при които е приложено управление с обратна връзка. Тази част от текста описва софтуерните обекти, използвани за извършване на проведените изследвания. Разгледани са и начините за моделиране на компютърни системи и са представени примери за измерване на производителността на уеб сървъри.

В първа част на главата се описва приложимостта на теорията на автоматичното управление, както и структурата и характеристиките на системите за автоматично управление. Представени са начините за управление в зависимост от събраната информация за системата и подходите за класифициране на системите за управление, които се основат на типа на модела, на характера на сигналите или на други признаци и свойства на системата. Обоснована е необходимостта от идентифициране и анализиране на изходните сигнали и от получаването на подходящо математическо описание – модел при прилагане на управление. Един от начините за получаване на математическо описание на изследваните системи (процеси, обекти) е чрез реализиране на експерименти, като по този начин се идентифицират характеристиките на поведението им.

Втората част на главата отбелязва примери за прилагане на принципа за управление с обратна връзка към избрания обект. Този вид управление се реализира и в различни софтуерни приложения за управление на бизнес функционалности, за управление на мрежови устройства, за управление на натоварването на клъстери от сървъри, за управлението на поточни медии и др. [DGHPT-02].

Третата част от главата представя софтуерните системи, които се използват за експерименталните цели на дисертационния труд – програмната система – уеб сървър Apache и операционната система Linux. Те позволяват да се осъществи качествен анализ на производителността на системата и да се изградят подходящи хипотези и изследвания за работещите процеси, планиране на нишките, ограничението на паметта и обработката на входно/изходните процеси. Производителността (Performance) е мярка за количеството работа, извършено от компютърната система. Ефективността на една компютърна система в зависимост от контекста, в който се използва, се определя от време за отговор, бързодействие, натоварване на хардуерните ресурси, достъпност до компютърната система или определено приложение, капацитет на използваната мрежата и др. [ApoA-94]. Анализирането на дадена компютърна система се осъществява в услуга на различни цели, но предимно за анализиране на мащабируемостта, т.е. изменението на производителността при промяна в натоварването или изменението на използването на ресурса.

Изследването на софтуерната система може да се осъществи по няколко начина: чрез оценка на производителността, измерване на параметрите на системата, експериментални изследвания, симулации и аналитични модели. Производителността на системата най-ефективно се определя чрез използване на два или повече от гореспоменатите подходи [BoGMT-05]. Моделирането на производителността е основна

област на изследване при уеб сървърите. Точният модел представя прогноза за показатели на производителността и за планиране на необходимия капацитет на ресурсите за изпълнение при различни настройки. За изготвяне на по-задълбочени изследвания на уеб сървърите и компютърните системи се използват QN мрежа от опашки и разширени мрежи от опашки.

В обобщение, глава първа от дисертационния труд разглежда основни положения от теорията на автоматичното управление, спира се на принципите за изграждане на автоматичните системи, както и на закономерностите, характерни за протичащите в тях процеси. Нейните подходи намират все по-голямо приложение при моделирането, анализирането и проектирането на сложни компютърни и комуникационни системи. Тази част от текста представя приложението на управлението с обратна връзка с цел адаптиране на изчисленията, гарантиране на производителността и планиране на ресурсите. За прилагане на ТАУ в софтуерни приложения, какъвто е интересът на дисертационния труд, е необходимо да се познават техните свойства и характеристики. Поради тази причина са описани компонентите и процесите, протичащи в операционната система Linux и уеб сървъра Apache.

В глава първа бе установено, че използването на системи от опашки за моделиране на производителността на компютърни системи осигурява отличен начин за изследване на различни проблеми в последните, разглеждайки ги като мрежа от опашки и сървъри. Те ефективно моделират поведението на компютърните системи в установено състояние. Но трябва да се има предвид, че при опити за моделиране на динамиката на компютърните системи, тяхното използване създава редица затруднения. Това означава, че за описание на динамиката на изследваната в този дисертационен труд система ще се наложи прилагането на линейни диференциални уравнения. Те са характерно средство за описание на системите в рамките на теория на автоматичното управление и са изключително полезни при изграждането на модели чрез статистически данни или входно-изходни данни.

Този подход за проучване на системите се разглежда и в теорията за идентификация на системи. В рамките на дисертационния труд разбирането на техните възможности и ограничения се търси чрез генериране на входни въздействия към системата и със софтуерни инструменти за измерване, наблюдение и диагностика на компютърната система. Това е необходима стъпка по посока на една от основните задачи на дисертационния труд – прилагане принципите на теория на автоматичното управление с цел получаване на адекватна информация за работата на програмната система.

Глава 2. Моделиране на компютърна система при различни входни въздействия

Във втора глава на дисертационния труд са представени начини за анализ на компютърни системи, които да подпомогнат изграждането на модели. Отбелязани са някои фактори, които влияят на производителността на системата. Тази част на текста представя структурата на използваната компютърна система за целите на експерименталната работа. Описани са методологиите за изследване на производителността на компютърните системи и са разгледани възможностите за наблюдение на софтуерните

и хардуерни компоненти на последните. Направен е анализ на програмите за генериране на входни въздействия и е очертан сценарий на тяхното прилагане. В резултат на това е изграден функционален модел при използването на софтуерно приложение, който да послужи за целите на експерименталната част на настоящото дисертационно изследване.

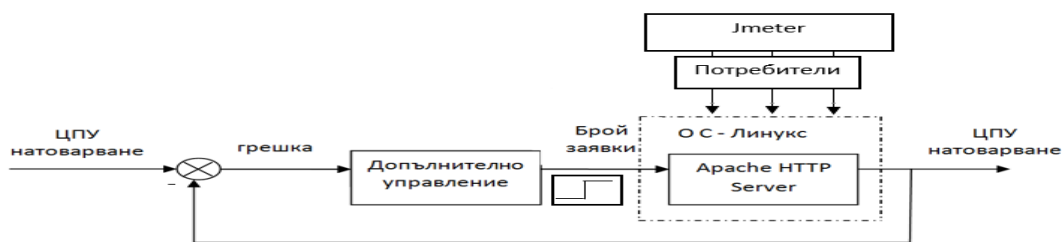
Първата част на главата идентифицира предимствата от изграждането на модел на компютърната система – за анализ на хардуера и софтуера на използваната конфигурация, за планиране на капацитета и отстраняване на ограничения на системата, за анализиране и коригиране на мащабируемостта, и т.н. Получаването на модел е свързано с методи за анализ на производителността. Описани са тяхното приложение, действие и ефективност.

Във втората част на главата са представени някои фактори, които могат да влияят на производителността на компютърната система – планирани системни действия, други потребители или други натоварвания. Бе отбелязано, че е изключително трудно откриването на смущения в големи и сложни компютърни системи. Това е от значение за задачите на дисертационния труд, тъй като данните, които се използват за изграждане на модели, много често са зашумени.

В третата част на главата е представена използваната компютърна система. По своята същност тя е сложна система за управление, в която протичат много процеси едновременно. При нея определени процеси могат да се разглеждат като входни въздействия за системата, други – като смущения за системата, а трети – като изходни данни от нейната работа. Това позволява компютърната система да се представи като система за автоматично управление (САУ). Според теория на информацията САУ е информационна система, която преобразува съответната информация за постигането на определени цели като регулиране, слеждане и програмно управление [СапуГ-07].

За изграждането на система за автоматично управление е необходимо да се проведат експериментални изследвания и да се създаде математически модел на обекта. Експерименти за компютърната система може да се реализират посредством анализа на натоварването и анализа на ресурсите, за са се определи модела на изследвания обект – уеб сървър Apache. Използваната компютърна система за провеждане на експериментални изследвания в дисертационния труд работи под операционна система Linux и инсталиран на нея уеб сървър Apache. Те съвместно осигуряват обслужването на потребители на уеб приложение за достъп до научно списание. При измерване на натоварването на процесите на Apache се реализира управление чрез програма, написана на bash скрипт и езика за програмиране AWK. Тя се изпълнява в потребителския режим на операционната система. Реализираната програма за управление следи нивото на натоварване на процесора от процесите, които създава уеб сървърът Apache. Промяната на броя потребители се явява измеримо смущение за натоварването на процесора. В нормален режим на работа то е много малко – 3-5%. При постъпване на заявки от до 20 потребителя на софтуерното приложение, той се натоварва с до 30 %. За ефективното им обслужване се въвежда първият вариант на настройки – highSetting, а когато натоварването достигне 60% се въвежда вторият вариант на настройките на Apache – bestSettings. По този начин допълнителното управление действа като двупозиционен регулатор.

Реализираната програма за управление преконфигурира настройките на Apache. Структурна схема на изследваната система е представена на фиг. II.4.



Фиг.П.4. Структурна схема на изследваната система

В четвъртата част на главата са представени методологиите за анализ на производителността. Те се разделят на няколко категории в зависимост от използваните продукти и подхода за наблюдение на показателите на производителността: липса на конкретни действия; задаване на въпроси; диагностичен цикъл; методи за изследване на ограниченията; USE метод; и др. [Gregg-13].

В петата част на главата са разгледани критериите за изграждане на програмни системи (време за отговор, производителност, цена) и основните настройки за ефективното им функциониране – настройки на софтуерно ниво (запис на файловата система, размер на блока), настройки на мрежово ниво (размер на буфера).

В шестата част от главата са описани начините за наблюдение на софтуерните и хардуерните компоненти под операционна система Linux – чрез броячи и чрез проследяване.

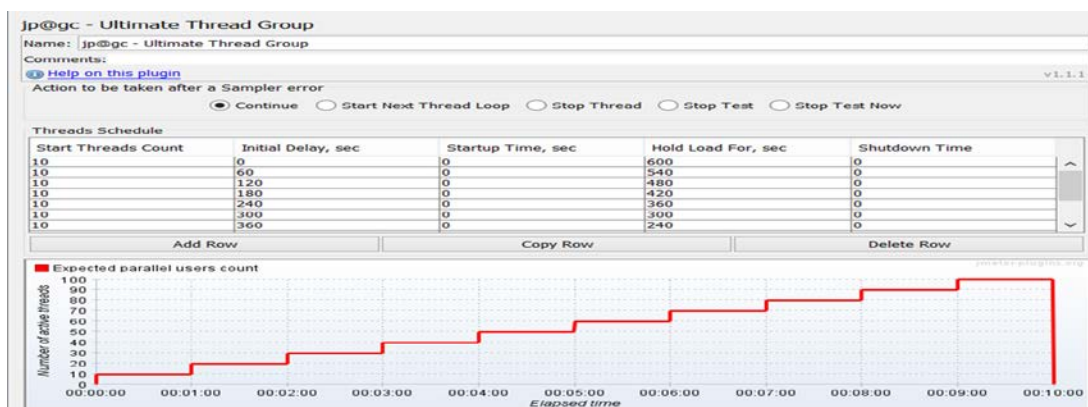
В седма част на втора първа са представени програми за генериране на входни въздействия. За изследването на компютърните системи и събирането на данни за ефективността на тяхната работа се използват програми за генериране на входни въздействия. Чрез тях се планират задачите за изпълнение от приложението, прилагат се различни сценарии и конфигурации с цел събиране на показатели за неговата производителност и надеждност. Този вид програми се използват от софтуерни дизайнери, администратори и обикновени потребители, но изискват познания за системата, която се изследва. Техните функционалности позволяват конфигуриране и генериране на различни входни въздействия, задълбочено наблюдение на поведението на услугите и предоставят показатели за сравняване и оценка на тяхното качество. Тези инструменти се използват, за да посочат недостатъците на системата при емулиране на стотици клиенти, позволяват бързо и ефективно реализиране на входни въздействия и на разпределение на заявките.

В тази част от текста се представя описание на сценария, по който са направени изследванията на софтуерното приложение, използвано за експерименталните цели на дисертационния труд: клиентите зареждат сайта hs22.iccs.bas.bg и се насочват към страница [index.php](#), след което потребители на софтуерното приложение разглеждат последователно страниците за новини ([news.php](#)), контакти ([contats.php](#)) и всички публикувани списания, които се разпространяват по електронен начин чрез посещаване на страницата за списанията ([all_papers.php](#)). Извършеният експеримент е с клиент, който вече е регистриран в системата, така че директно се използва страницата [login.php](#), където се въвеждат потребителско име и парола. Проверката установява, че името и паролата съществуват в базата данни и това позволява на потребителя да извърши търсене в нея на целия текст на статията. Той търси две статии по два критерия. Първият е „име на автор“,

а вторият е „година на публикуване“. След това потребителят прочита статиите в pdf. Всички тези действия, които се извършват от софтуерното приложение се използват, за да се симулират заявки от клиентите на системата в определен момент. По този начин се изследват възможностите на система за обслужване на потребителите на уеб базираното приложение за електронно разпространение на научно списание.

Осъществяване на проучване с различен брой потребители цели да се изгради модел на параметрите, измерващи производителността на системата. За получаването на модел на софтуерното приложение е необходимо да се извършат изследвания, при които се натоварва сървър, като се симулират множество потребители, които потребяват неговите услуги едновременно. По този начин се моделират функциите, изпълнявани от системата като действия, операции или процеси, и се определят реакции на системата като бързодействие, устойчивост, надежност и мащабируемост. Моделът гарантира, че софтуерното приложение ще работи успешно при пускането ѝ в експлоатация за използване от реални клиенти.

За реализиране на модел се използва модула UltimateThreadGroup на програмата Jmeter, който позволява да се планират безкраен брой записи с потребители на системата при провеждане на изследването. В експеримента се симулира използване на сървър от 110 потребителя, които започват да изпращат заявки в различни времеви интервали. На фиг. II.7 е показано, че в първите 60 секунди услугите предлагани от него се използват от 10 потребителя в следващите 60 секунди се добавят нови 10 и така всяка минута се добавят по 10 потребителя.

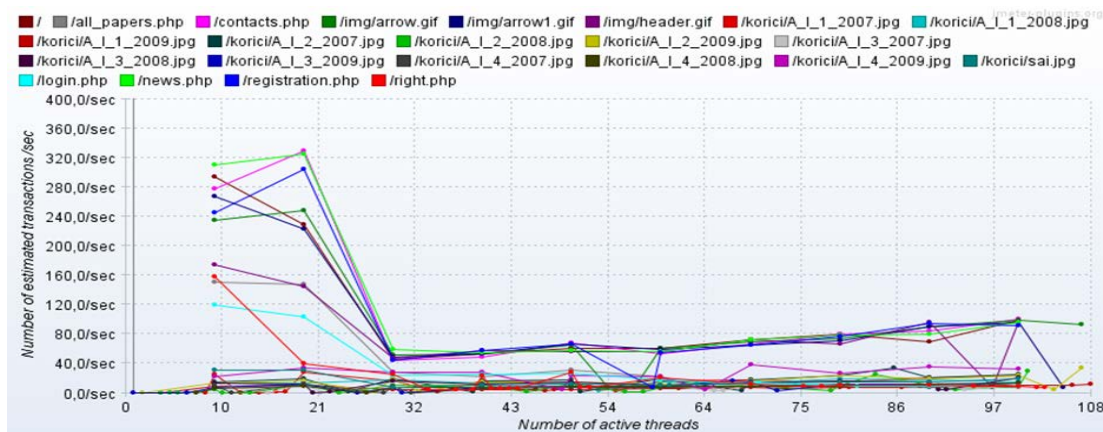


Фиг. II.7. Настройване на потребителите чрез UltimateThreadGroup

Изследват се основни показатели за производителността на системата и се моделират нейните възможности. Те се измерват чрез listener елементите на програмата Jmeter, които съхраняват резултатите от http заявките във файлове и представят визуален модел на данните [AraSF-16].

Throughput измерва броя на операциите, извършени от системата повреме на изследването, и изразява капацитета на един сайт или едно софтуерно приложение. Целта на неговото използване е да се определи броят на заявките за час/минута/секунда, с които може да се справи. Фигура II.8 показва статистически данни за максималния брой възможни транзакции (обслужени заявки от сървър) на базата на броя на потребителите, имащи достъп до софтуерното приложение. Линиите на фигурата представят всяка

страница на уеб приложението, която се изследва. Раздалечеността между линиите се получава, защото една част от страниците съдържат статични файлове - снимки, а друга част са файлове, които се генерират динамично чрез езика php. Фигурата представя ограничението на приложението, което е малък брой обслужени транзакции за статични файлове и необходимостта от промяна големината на снимките.



Фиг.П.8. Зависимост между броя транзакции и броя потребители

Следващият показател, който се моделира чрез програмата Jmeter, е времето за отговор – то показва средното време за обработка на заявките от сървъра в милисекунди. От направените експерименти става ясно, че времето за отговор се изменя в зависимост от изпратените заявки от потребителите, а за да се постигне определено време за отговор трябва се поставят ограничения в работата на системата.

Третият показател, който се моделира чрез Jmeter, е латентността на системата. Анализирано е времето, след което ще започнат да се обработват заявките в зависимост от броя потребители и е установено, че тя се увеличава при увеличаване на потребителските заявки. За да се обяснят промените в латентността и времето за отговор се измерват и хардуерните компоненти на системата – процесор, памет, хард диск и др. За тази цел се използва елементът PerfMonCollector, който позволява да се измерят приложените входни въздействия към сървъра и да се съберат статистически данни за процесора, паметта, swap и др. С негова помощ бе установено, че паметта се променя в зависимост от потребителите на приложението, а процесорът се натоварва константно през цялото време на изследването. Резултатите не показват пренатоварване на хардуерните ресурси и затова се изследват и други показатели на системата. На фиг.П.12 е показано натоварването на TCP комуникационния протокол и твърдия диск на системата. Тези измервания се извършват с цел да се проследи източника на увеличаването на времето за отговор и латентността.



Фиг. II.12 Натоварване на TCP комуникационния протокол и твърдия диск на системата

От представените резултати може да се направи заключение, че TCP комуникационният протокол е в пренатоварено състояние в последния времеви интервал на изследването на системата, което обяснява увеличаването на времето за отговор и латентността ѝ. Става ясно, че неговото натоварване е пропорционално на броя потребители на системата. Синята линия на фигурата демонстрира, че натоварването на твърдия диск на системата е малко, което означава, че увеличаването на времето за отговор не е породено от него. Пиковите в нейното поведение се получават, когато процесите не могат да се обслужват от паметта и се предават към хард диска[GheGF-13].

Втора глава на този текст демонстрира, че опциите на програмата Jmeter позволяват да се изгради модел на функционалните възможности на компютърната система чрез изследване разработеното приложение и хардуера, който осигурява неговото поддържане. По този начин става възможно да се анализира бързо и лесно голямо количество данни от измервания на генерираното натоварване към уеб сървър и да се изгради математически модел за промените в показателите за производителността на компютърната система. Функционалният модел на системата дава възможност да се изследва изменението в показателите за производителност по отношение на нейните устойчивост и надеждност.

Глава 3. Реализиране на управление в изследваната компютърна система

В трета глава на дисертационния труд са представени хардуерните и софтуерните характеристики на компютърната система. Текстът разглежда софтуерното приложение, което се използва за изследване на производителността на програмната система. Приложено е управление чрез настройка на комуникационния протокол TCP и на програмната система - уеб сървър Apache. Описана е програмата за автоматично настройване на конфигурационния файл на Apache. Представена е ефективността на направените подобрения чрез показателите за качество на компютърната система (време за отговор и бързодействие).

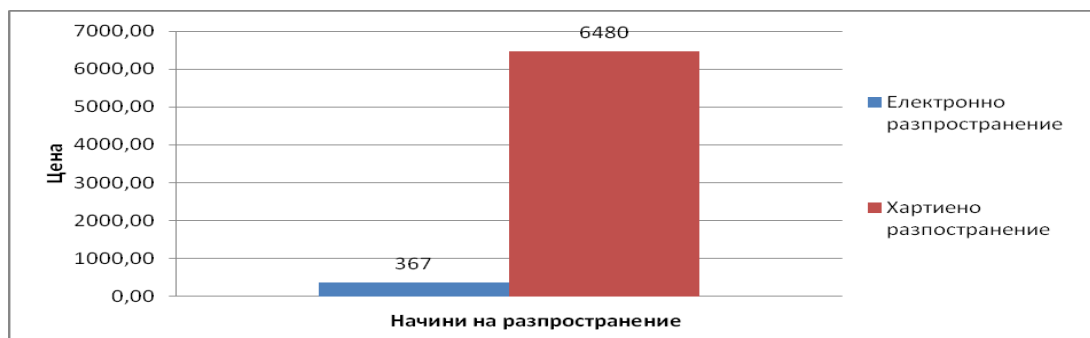
Реализирането на управление в изследваната компютърна система има две основни цели: 1/ да се покаже, че създаденото софтуерно приложение за достъп до статиите на научно списание предоставя по-модерен, по-лесен и по-функционален начин за достъп и търсене в материалите, разпространявани от Съюза по автоматика и информатика; 2/ да докаже, че промяната на параметрите на компютърната система, в това число параметрите

на конфигурационния файл на Apache и на комуникационния протокол TCP, ще подпомогне едновременния достъп на повече читатели до системата.

В първата част на главата са представени хардуерните и софтуерните характеристики на компютърната система, която се използва в дисертационния труд. Описано е интернет-базираното софтуерно приложение, което позволява селективно търсене, обработване и извличане на информация за статиите, публикувани в българско научно списание. Чрез него се предлага информационна услуга в Интернет за електронен абонамент на научно списание. Това софтуерно приложение използва трислоен клиент/сървър модел за динамичното генериране и предоставяне на информация [TricE-11]. Приложението е реализирано чрез няколко софтуерни модула, разработени посредством PHP (HypertextPreprocessor). Чрез тези модули се реализира SQL заявка (request) и SQL отговор (response), необходими за функционирането му. След получаване на SQL заявката се претърсва базата данни MySQL, за да бъде извлечена желаната от потребителя информация. Данните от базата се обработват и структурират в HTML формат с цел отговаряне на SQL заявката като SQL отговор (response). Използваните продукти са от т.нар. свободен код (open source). Те са системно независими, поради което е възможно използването им при Windows-, Unix- и Linux- базирани машини.

Конкретната база данни на софтуерното приложение има четири отделни таблици. В тях се съдържа подробна информация за статиите, регистрираните потребители, потребителите с платен абонамент и направлението, в което са публикациите.

За да се демонстрира икономическата ефективност на решението е реализирано сравнение на електронното и хартиеното разпространение на списание „Автоматика и Информатика“. Разходите за електронно разпространение на списанието са изчислени въз основа на цената на електроенергията и консумацията на електроенергия от сървъра за една година. В разходите за хартиено разпространение са включени тираж на списанието за година (4 книжки годишно $\times$ 500 броя=2000 бр.) и цена на една книжка. Резултатите показват, че разпространението на списанието по електронен път е 17 пъти по-евтино от начина, използван в момента [Димит-13а].

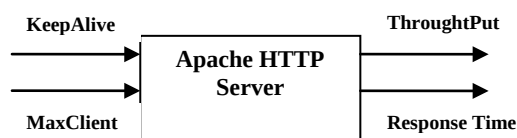


Фиг. III.3. Сравнение на начините на разпространение на списание „Автоматика и Информатика“

Във втората част на главата се представят резултати от изследване производителността на програмната система. Анализирани са натоварванията, които уеб сървърът може да обслужи и е определено натоварването на хардуерния ресурс. Изследването на производителността позволява да се измери количеството работа,

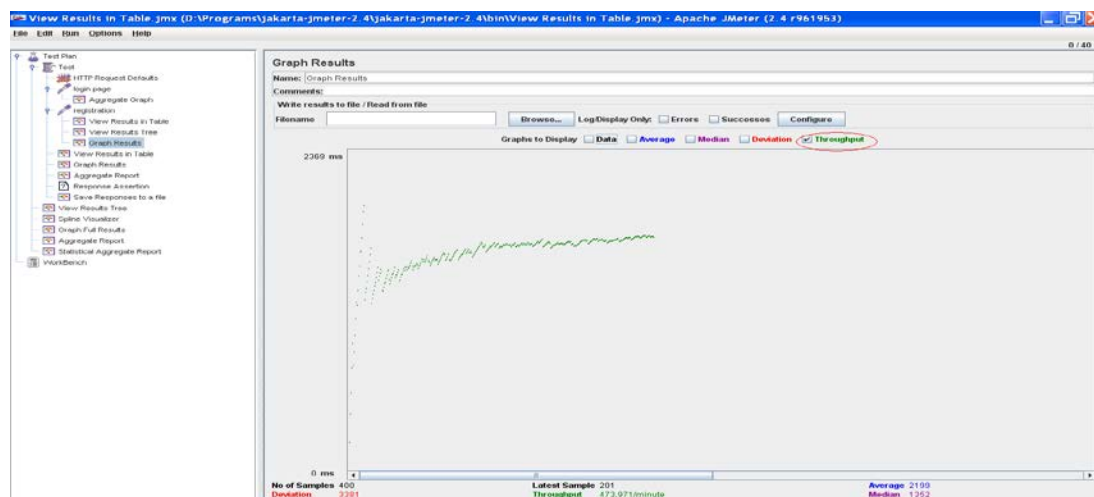
извършено от компютърната система, и да се разберат възможностите на разработения софтуер и на машината, която се използва за неговото поддържане.

В дисертационния труд програмната система се представя като система за автоматично управление. Процесът по изследване на работоспособността се осъществява чрез генериране на входни въздействия във вид на http заявки от Jmeter към уеб сървър Apache, който поддържа софтуерното приложение. Apache HTTP Server е представен като система за автоматично управление, на която са изследвани показателите на производителността – бързодействие и време за отговор, при зададени максимален брой свързвания (определяни от параметъра KeepAlive) между сървъра и потребител на системата, като се разрешава определен брой потребители да се свързват със сървъра (чрез параметъра MaxClients) (Фиг. III.4). По този начин се изследва системата като стабилизираща при зададени постоянни стойности на конфигурационните параметри на Apache KeepAlive и MaxClients.



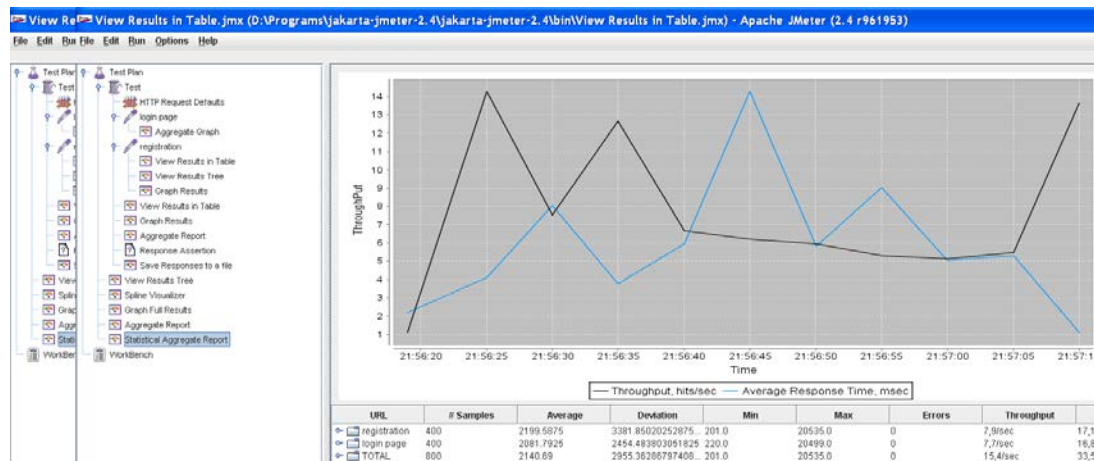
Фиг. III.4. ApacheHTTPServer, представен като система за автоматично регулиране

В изследването се изпращат HTTP заявки към уеб сървъра Apache по метода Post чрез порт 80. Проследява се поведението на сървъра при свързване на 40 потребителя, които изпращат по 10 заявки. Експериментът е проведен при условие, че потребителите серегистрират (registration)и влязат в софтуерното приложение (loginpage). Както се вижда по-долу от резултатите за производителността на системата, тя е устойчива.



Фиг. III.9. Графични резултати

Обобщената статистика предоставя информация за двете изходни величини: време за отговор (ResponseTime) и бързодействие на сървъра (ThroughPut).). Данните са с близки стойности, а това означава, че системата за автоматично управление е стабилна и надеждна. Колебанията в получените изходни резултати се получават от подаваните заявки към база данните MySQL. От друга страна, изходните данни показват, че при задаване на постоянни стойности за параметри KeepAlive и MaxClients на уеб сървъра Apache, той се саморегулира много добре[Димит-11].



Фиг. III.10. Обобщена статистика

Във втората част на главата се представят и резултатите от изследване чрез входни въздействия към програмната система, като се използват видео файлове. За целта софтуерното приложение за разпространение на научни статии е разширено чрез добавяне на модул за популяризиране на конференции, обсъждания и семинари чрез видео файлове. Те се характеризират с голям обем и това води до по-голямо натоварване на ресурсите на хардуерната конфигурация. Последното налага изследване подобряването на работата на програмната система чрез включване на допълнителен модул `mod_deflate` в нея, понеже той компресира съдържанието на файловете и намалява техния размер преди да се изпратят от сървъра към потребителя. Използва се и генератор за статични страници `Apachebench`, който позволява генериране и изпращане на заявки с различна последователност и големина. В експериментите се изпращат общо 250 заявки от `Apachebench`, които са разделени на групи с различна големина. Тези действия, погледнати от перспективата на ТАУ, представляват единични входни въздействия с различна амплитуда. Целта на експеримента е да се сравнят конфигурационните настройки по подразбиране на уеб сървъра Apache с предложените в този дисертационен труд подобрени настройки. Изследва се поведението на хардуерните компоненти - процесора и паметта при натоварване на сървъра чрез различни по големина и начин на изпращане заявки към видео файл. Отчита се включването на параметъра `mod_deflate` на Apache и неговото влияние върху използването на хардуерните компоненти.

Получените резултати от проведените експерименти с `Apachebench` при различни настройки на уеб сървъра (по подразбиране и подобрени) са представени на фиг III.11. Лявата колона на фигурата показва, че прилагането на подобрени настройки увеличава производителността на сървъра, което е отчетено чрез показателите: продължителност на

теста (Time taken for test) и време, необходимо за обслужване на заявка (Time taken for request).

<pre>[root@hs12 ~]# ab -n 250 -c 250 http://hs12.iccs.bas.bg:80/venko.mp4 This is ApacheBench, Version 2.3 <\$Revision: 655654 \$> Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/ Licensed to The Apache Software Foundation, http://www.apache.org/ Benchmarking hs12.iccs.bas.bg (be patient) Completed 100 requests Completed 200 requests Finished 250 requests Server Software: Server Hostname: Server Port: Document Path: Document Length: Concurrency Level: Time taken for tests: Complete requests: Failed requests: Write errors: Total transferred: HTML transferred: Requests per second: Time per request: Time per request: Transfer rate: Connection Times (ms) min mean[+/-sd] median max Connect: 12 15 1.5 16 17 Processing: 896 28446 11862.2 34455 36521 Waiting: 8 6677 3099.0 6761 12673 Total: 914 28461 11861.4 34471 36534 Percentage of the requests served within a certain time (ms) 50% 34471 66% 35601 75% 36147 80% 36150 90% 36389 95% 36533 98% 36534 99% 36534 100% 36534 (longest request)</pre>	
<pre>[root@hs12 ~]# ab -n 250 -c 250 http://hs12.iccs.bas.bg:80/venko.mp4 This is ApacheBench, Version 2.3 <\$Revision: 655654 \$> Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/ Licensed to The Apache Software Foundation, http://www.apache.org/ Benchmarking hs12.iccs.bas.bg (be patient) Completed 100 requests Completed 200 requests Finished 250 requests Server Software: Server Hostname: Server Port: Document Path: Document Length: Concurrency Level: Time taken for tests: Complete requests: Failed requests: Write errors: Total transferred: HTML transferred: Requests per second: Time per request: Time per request: Transfer rate: Connection Times (ms) min mean[+/-sd] median max Connect: 13 16 1.6 16 18 Processing: 974 30516 11947.9 36139 38172 Waiting: 8 6374 3019.0 6304 12674 Total: 991 30532 11946.9 36156 38186 Percentage of the requests served within a certain time (ms) 50% 36156 66% 37262 75% 37794 80% 37798 90% 38029 95% 38185 98% 38185 99% 38185 100% 38186 (longest request)</pre>	

Фиг. III.11. Резултати от експеримента – сравнение на настройките по подразбиране с предложените подобрени настройки

По време на изпращане на заявките чрез Apachebench, с програмата SAR е проследено поведението на хардуерните компоненти процесор и памет, отнасящо се до промените в конфигурационния файл на Apache. SAR е команда на операционната система Linux, която брои и показва дейностите, изпълнявани от последната. По този начин се изследва влиянието на подобрените конфигурационни настройки към основните хардуерни компоненти. От направените експерименти може да се види, че натоварването на процесора и паметта е намаляло, съответно на процесора с 2,37%, а на паметта – с 8,93%.



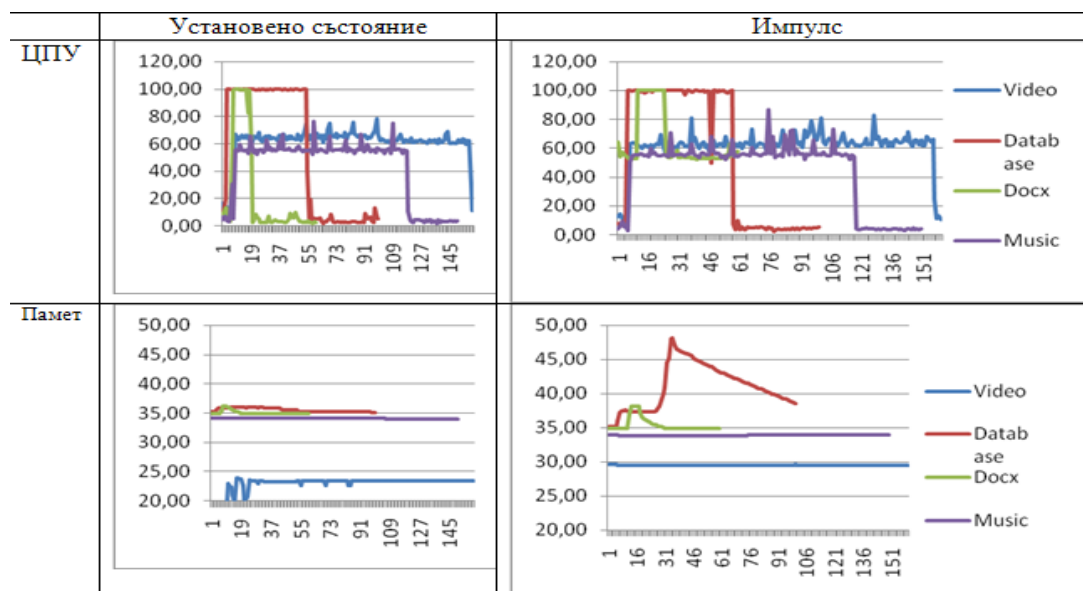
Фиг. III.12. Натоварване на процесора и паметта при различни настройки на конфигурационния файл

При изследване влиянието на модула mod_deflate върху времето за обслужване на заявки от сървъра са използвани три варианта за настройка на уеб сървъра Apache. Първият е настройки по подразбиране на Apache, вторият е оптимизирани настройки на prefork и изключен модул mod_deflate, а третият е подобрени настройки на prefork и включен модул mod_deflate. Става ясно, че включването на модула mod_deflate не влияе

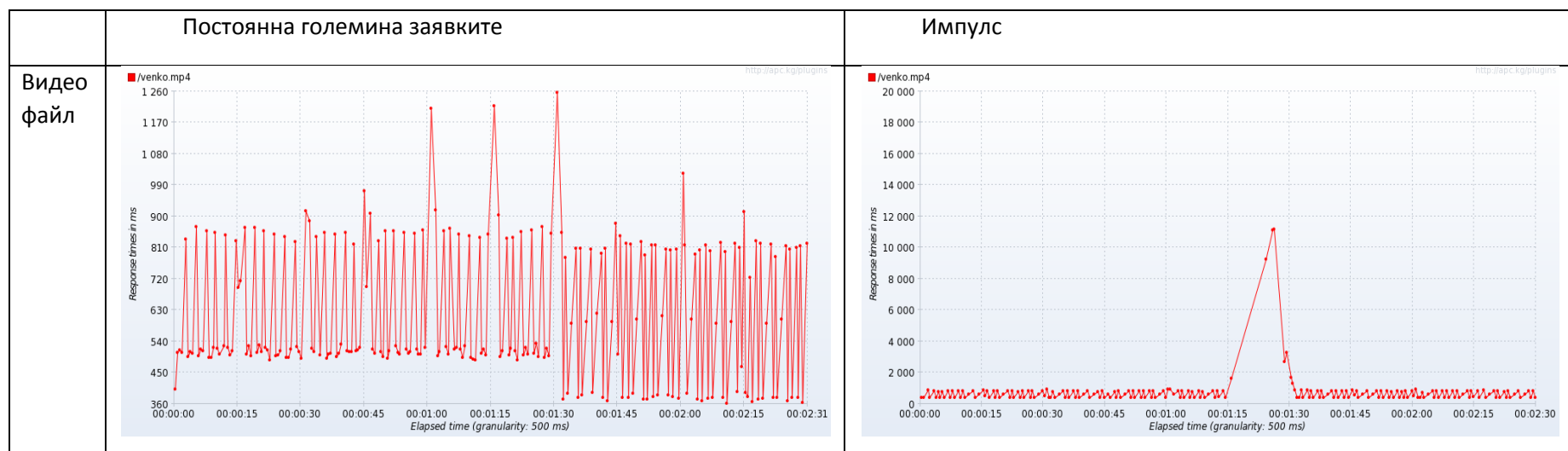
на времето за обслужване на заявките. Неговото самостоятелно включване не е ефективно решение за подобряване на производителността на системата за обработване на заявки към видео файлове [DimiT-13].

Във втората част на главата са представени и резултати от реализиране на входни въздействия към програмната система чрез различни видове файлове. Обектът на управление – Apache сървър е изследван при нови входни въздействия, които са музикални, видео, текстови файлове и база данни. Използва се програмата Jmeter за изпращане на входни въздействия във вид на http заявки и се проследяват промените в натоварването на най-важните хардуерни компоненти – процесора и паметта. Заявките са изпращани с постоянна големина и чрез импулс в определен момент от време, като се наблюдава поведението на показателите за производителност – време за отговор и бройна заявките в приемащата опашка на уеб сървъра. Резултатите показват (фиг.14 а-г), че когато заявките се изпращат с еднаква големина на импулса към програмната система, времето за отговор е постоянно, а при изпращане на единичен импулс към сървъра, времето за отговор се увеличава изключително много. Това е предпоставка да се търси решение за намаляване на времето за отговор от системата. Ако използваме терминологията на теория на автоматично управление, изпращането по този начин на заявки към системата може да се представи като реално входно въздействие и като единична импулсна функция.

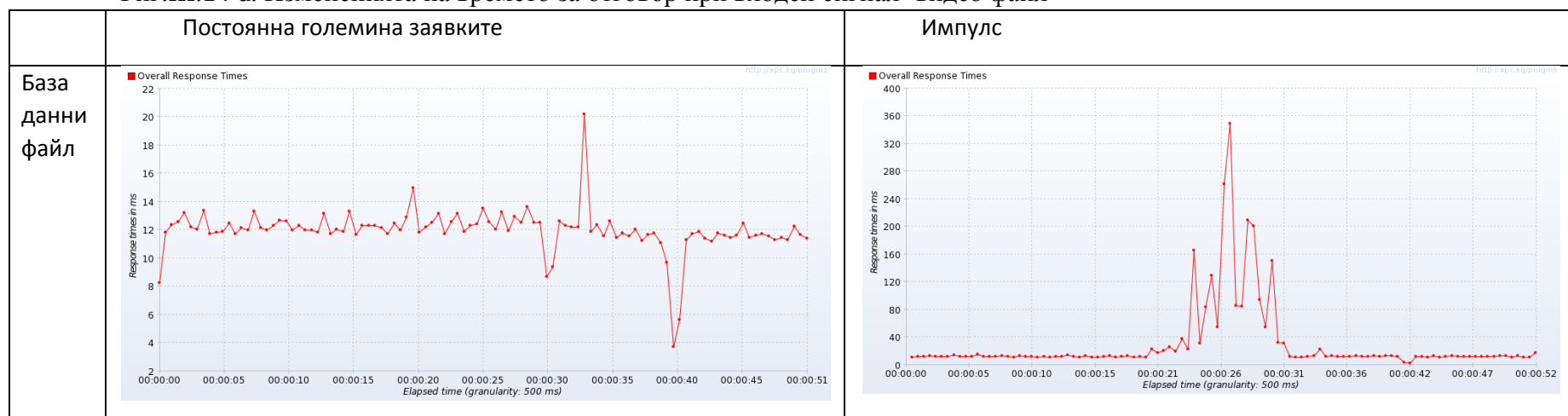
При измерване натоварването на хардуерните компоненти бе констатирано, че при изпращане на импулс се наблюдават по-големи и по-чести флуктуации в използването на процесора и паметта при всички видове файлове. На базата на тези смущения се достига до извода, че е необходима промяна в конфигурационните параметри на уеб сървъра Apache и на ТСРкомуникационния протокол. Така ще се облекчи работата на програмната и компютърната системи и ще намалят смущенията в тях [Димит-13].



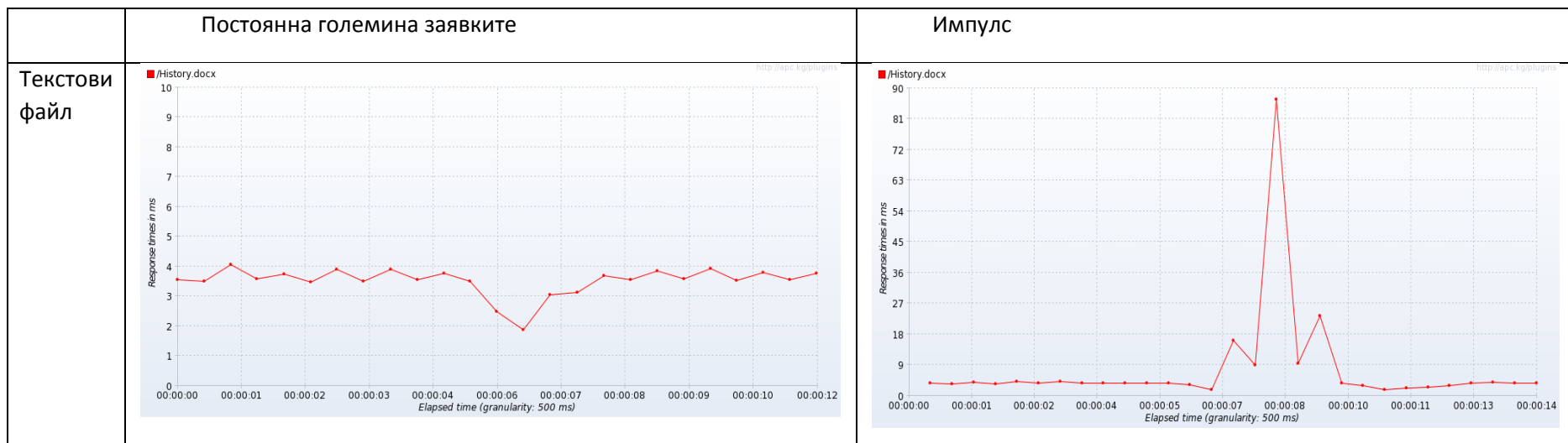
Фиг. III.15. Реакция на изходните сигнали



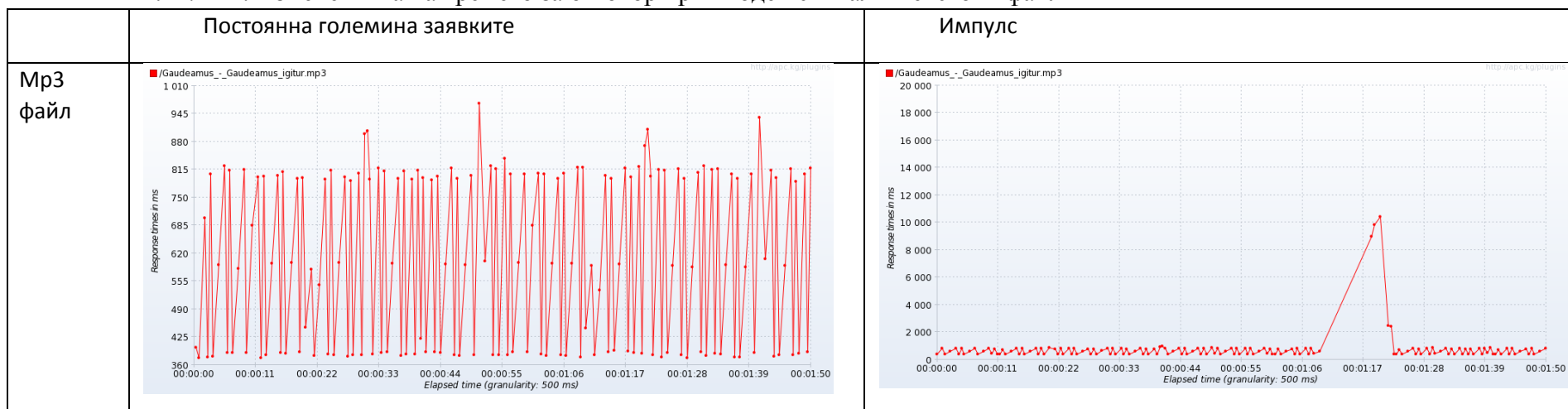
Фиг. III.14-а. Измененията на времето за отговор при входен сигнал- видео файл



Фиг. III.14-б. Измененията на времето за отговор при входен сигнал – файл база данни



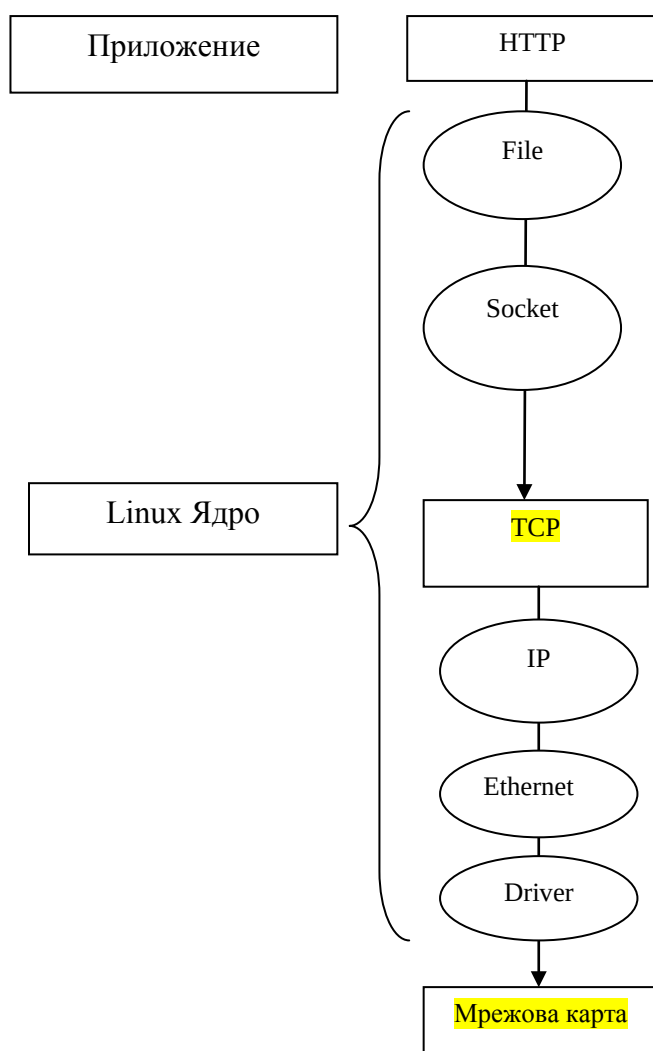
Фиг.ІІІ.14-в. Измененията на времето за отговор при входен сигнал – текстови файл



Фиг.ІІІ.14-г. Измененията на времето за отговор при входен сигнал – мр3 файл

В третата част на главата се представят резултати от експерименти заподобряване производителността на изследваната програмна система чрез промяна на нейни параметри с цел подпомагане на едновременния достъп на повече читатели до софтуерното приложение. Въвеждането на допълнителни настройки на различни нива в компютърната система увеличава производителността на програмната система [Gregg-13]. Това се потвърждава и от факта, че при разработването на софтуерни приложения, усилията на екипите са насочени към бързо внедряване на софтуера и се отделя малко време за измерване на производителността и оптимизирането им. Лошо са организирани бази данните и не са реализирани подходящите настройки на устройствата за съхранение на данните. Тези недостатъци са предпоставка за търсене на начини за подобряване на използваните софтуерно приложение, уеб сървър и операционна система.

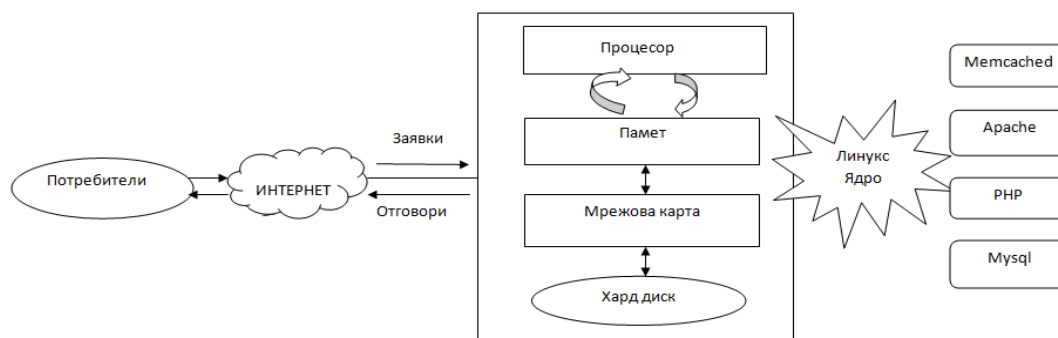
Компютърната система може да се представи като последователност от подсистеми, работещи в синхрон. Тя включва подсистемата на TCP протокола, подсистемата на http протокола и подсистемата на входно/изходните данни. Параметрите на тези подсистеми имат нужда от допълнителна настройка за специфичната хардуерна конфигурация, на която работят [MDVHR-01].



В дисертационния труд тази допълнителна настройка е извършена по отношение на параметрите в Linux ядрото и по-специално в мрежовите настройки и TCP комуникационния протокол (в жълт цвят на фиг. III.16). Комуникацията между крайните потребители преминава през три основни зони – първата зона е самата програмна система – уеб сървър Apache; втората зона е компютърната система, която включва операционната система и TCP настройките; третата зона е пътят по мрежата.

В дисертационния труд се извършва управление в програмната система, т.е. в уеб сървър, както и във втората зона (компютърната система), където се анализират и подобряват настройките на TCP протокола и се инсталират програми за подобряване производителността на операционната система.

Фиг. III.16 Последователни подсистеми



Фиг. III.17. Комуникацията между потребителите и сървъра

Производителността на компютърните мрежи на ниво TCP протокол се ограничава от два прозореца: прозорец на задръстванията (congestion window) и прозорец на получаване (receive window). За постигане на по-качествено управление се изисква да се направят допълнителни настройки на параметрите на TCP протокола. Те се разделят в няколко категории според тяхното предназначение [WigeN-13],[OracC-00],[SoftP-15]

- о За подобряване на производителността – чрез регулиране размера на опашките, изключване на селективното потвърждение, повишаване на прозореца на задръстванията и др. Операционната система разполага с настройки за регулиране размера на опашката между мрежовата подсистема на ядрото и драйвера на мрежовата карта. Размерът на опашката се настройва така, че да няма загуби на данни, поради препълване на локалния буфер. При използване на неподходящи настройки на TCP протокола се попада в състояние на управление на задръстването, което ограничава големината на изпратените пакети и причинява загуби на UDP пакетите.

- о За обслужване на повече клиенти – за сървъри, които обслужват голям брой сесии едновременно се изисква включване на няколко допълнителни опции. Чрез тях се определя диапазонът на използваните портове от UDP и TCP протоколите, оптимизира се производителността при голям брой клиенти, управляват се свързвания по време на използване на TCP протокола и се променят настройките на времето за затваряне на свързването (close connection) и времето за убиване на остарял процес [Moura-00]. Използва се и опция за увеличаване на максималния брой свързвания, които не са получили потвърждение [Garre-09].

- о За подпомагане работата на web server – при инсталиран уеб сървър, който обслужва много кратки TCP свързвания, е подходящо да се настроят следните опции: включване на параметъра TCP_tw_recycle, който изчиства времето за изчакване (time_wait) в сокетите; включване на параметъра TCP_reuse, който позволява повторно използване на сокета и намалява натоварването на сървъра; настройване на параметъра TCP_synack_retries, който определя броя синхронизиращи пакети, изпратени преди ядрото да откаже свързването. Друга опция, подпомагаща работата на уеб сървъра е /Net/core/somemconn. Тя се използва за определяне на броя приети заявки и позволява да се обработят голям брой заявки едновременно [HTCAG-10].

Обикновено уеб сървърът Apache позволява лесно изграждане на среда за поддържане на уеб сайтове, с минимална промяна в конфигурациите и малко усилия от страна на администратора. Но при използването на настройките му по подразбиране се наблюдава голямо натоварване на хардуерния ресурс на компютърната конфигурация и особено на паметта. Това поставя нелеката задача за оптимизиране производителността на системата, тъй като липсват лесни и бързи правила за нейното решение. За да се постигне максимална производителност се изискват много експерименти с различни опции и условия [Avoya-11], които включват вид на заявките, изпълнявани от сървъра, оптимизиране на база данните, кеширане на php скрипта. В хода на дисертационния труд са разгледани и подобрили всички тези условия.

Управлението на програмната система е реализирано чрез оптимизиране на модулите, използвани от Apache, и чрез синтез на директивите на Apache. Извършен е анализ на използваната памет на компютърната система от програмната система и е изчислена стойността на параметъра на конфигурационния файл на Apache MaxClients по два начина: чрез програмни кодове, написани на езика Perl; чрез команди, написани на езика AWK и се стартира Linux програмата Free. При втория начин се използва по-малко хардуер ресурс за установяване на показателите на системата, понеже не е необходимо да се стартират и компилират допълнителни програми. Посредством тези два начина на изчисление се получава информация за паметта, използвана от един процес на Apache и от всички други приложения, работещи с операционната система.

На базата на изчисленията за използване на паметта на хардуерната конфигурация се предлагат два варианта за промяна на настройките на Prefork модула на Apache. При първия се намаляват стойностите на параметрите на prefork модула. Целта е да се разгледат възможностите за реакция на системата и да се реализира управление, при което при по-малък брой потребители на софтуерното приложение, хардуерните ресурси да се използват от другите програми, работещи с операционната система. Във втория се увеличават стойностите на параметрите на prefork модула – по този начин се изисква повече хардуерен ресурс за уеб сървъра Apache и се стартират допълнителни дъщерни процеси при по-голям брой потребители на системата. Това се осъществява чрез промяна на параметрите (bestSettings и highSettings), показани в таблица III.2. Това са основни директиви за настройка на уеб сървъра Apache, които се използват за определяне на процесите, изпълнявани от него.

Таблица III.2. Параметри на директивите на Apache

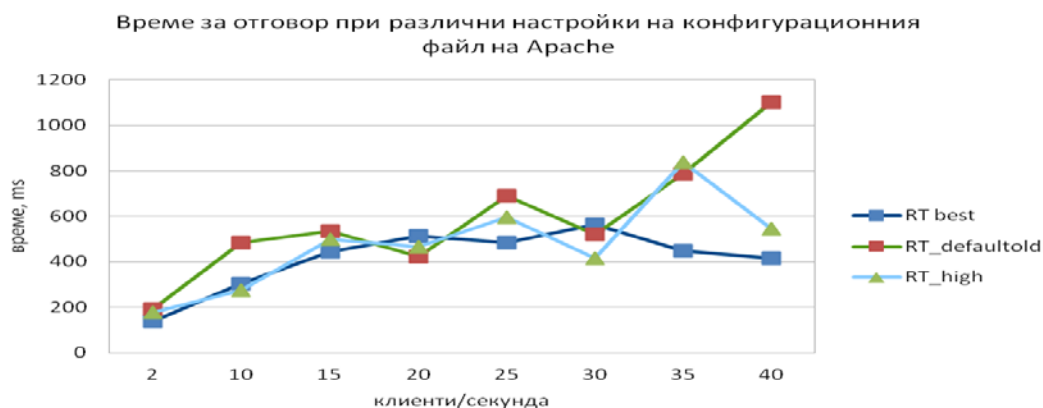
Prefork	bestSettings	highSettings
StartServers	3	10
MinSpareServers	3	10
MaxSpareServers	10	15
MaxRequestWorkers	50	84
MaxConnectionPerChild	1000	1000
Timeout	40	45
MaxKeepAliveRequests	14	14
KeepAliveTimeout	2	2

В резултат от направените промени в настройки на директивите за конфигуриране на уеб сървъра Apache се променя и натоварването на основните компоненти на системата т.е процесора и паметта. Използването на процесора при настройки highSetting е представено в колона “cpu_high”, а при bestSettings – в колона “cpu_best”. Използването на паметта при настройки highSetting е показано в колона “mem_high”, а при bestSettings – в колона “mem_best”. От резултатите става ясно, че когато постъпват по-малко от 20 клиента, настройките highSetting са по-ефективни за обслужването на клиентите, понеже се използва по-голям ресурс на процесора и по-малък ресурс на паметта. За по-голям брой клиенти използването на конфигурационните настройки bestSettings са по-ефективни, тъй като се използва по-малко натоварване на паметта и по-голямо на процесора. Чрез прилагането на два варианта на настройки за конфигурационния файл се постига по-бързо обслужване на клиентите и по-ефективно използване на наличните хардуерни ресурси.

Таблица III.3. Промени (в %) на използваните памет и процесор

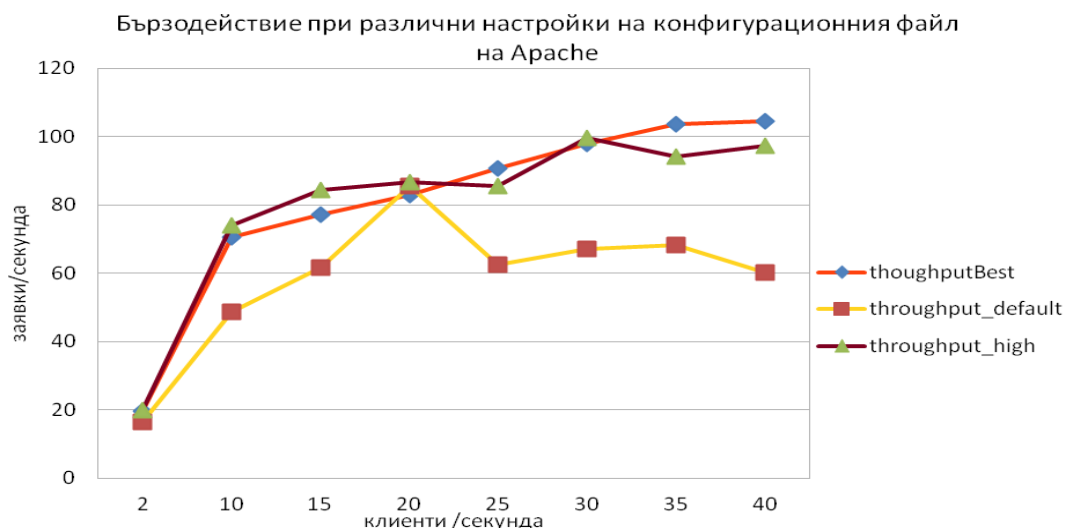
Клиенти	cpu_best, %	mem_best, %	cpu_high, %	mem_high, %
2	34,123	95,129	43,4142	88,845
10	39,974	94,84	43,324	89,271
15	37,989	91,729	36,188	93,521
20	39,934	94,284	42,788	93,209
25	40,072	92,164	37,689	84,995
30	40,319	91,923	40,313	85,19
35	40,559	90,401	44,864	84,963
40	46,41	82,648	42,476	89,137

Ефективността на направените промени в конфигурационния файл на Apache се изследва и чрез показателите за производителността на системата – бързодействие и време за отговор. Резултатите представят измененията в тези два показателя при три случая: с настройки по подразбиране (default) и два варианта за преконфигурирани настройки на конфигурационния файл на Apache (bestSettings и highSettings). Установява се, че за настройки по подразбиране времето за отговор (RT_default) е по-голямо, отколкото за двата случая с преконфигурирани настройки. Когато сървърът работи с настройки по подразбиране, времето за обслужване на потребителските заявки е по-голямо. Предложените два варианта за преконфигурирани настройки на параметрите на директивите на Apache предоставят по-бързо обслужване на потребителските задачи.



Фиг. III.20. Изменение на времето за отговор при различни настройки на конфигурационния файл

Също така бе установено, че бързодействието с настройките по подразбиране е по-малко в сравнение с двата случая на преконфигурираните настройки. Бързодействието с настройки highSettingse по-добро, когато се обслужват по-малко от 20 клиента едновременно.



Фиг. III.21. Бързодействие при различни настройки на конфигурационния файл.

Създаването на управление за автоматично преконфигуриране на Apacheцели по-бързо обработване на заявките в зависимост от броя на потребителите. От получените резултати може да се направи извода, че програмата за автоматично преконфигуриране на Apache трябва да използва настройките, наречени highSettings, когато софтуерният приложение се използва до 20 потребителя и настройките bestSettings, когато потребителите са повече от 20. В програмния код за преконфигуриране на Apache се използват Linux програмата PСи езика AWK.

Преконфигурирането на уеб сървър Apache се реализира чрез измерване на параметъра %сри. В програмата за управление той измерва колко процесорно време се използва от процеса и въз основа на това се преконфигурира конфигурационния файл на Apache. Този параметър се проверява в цикъла while, след което се избира кои

конфигурационни файлове да се използват оглед неговата стойност. Представеният програмен код осъществява автоматично преконфигуриране на уеб сървъра Apache на потребителско ниво в зависимост от броя потребители на софтуерното приложение. По този начин се прилага по-ефективно обслужване на потребители на разработеното приложение.

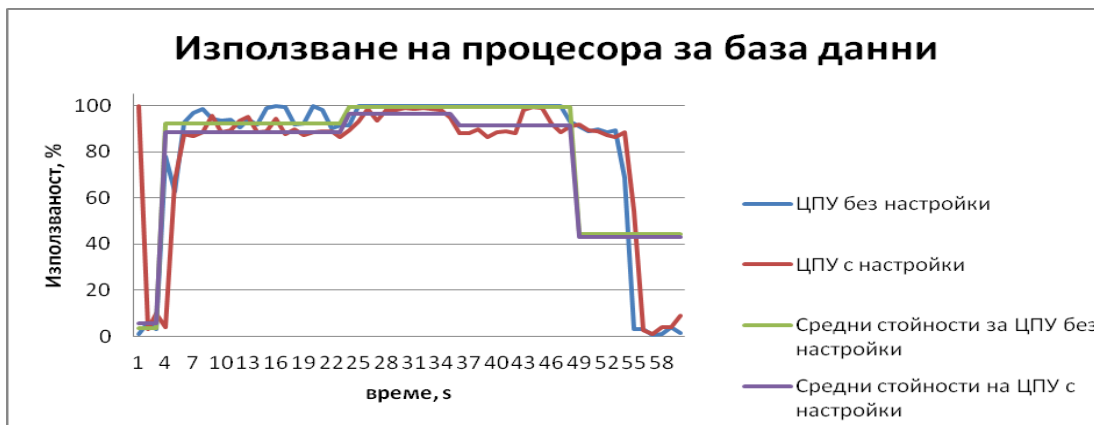
Управляващата програма е от типа:

```
#!/bin/bash
#!/bin/sed
while :
do
for i in $( ps aux | grep opt/lampp/bin/httpd
|awk '{s+=$3} END {print s}' );
do
if [ $i \> 3.9 ]
then
cp httpd-mpm_2.conf httpd-mpm.conf
cp httpd-default_2.conf httpd-default.conf
sudo /opt/lampp/lampp reloadapache
echo "Load new settings!"
elif [ $i \< 1.00 ]
then
cp httpd-mpm_1.conf httpd-mpm.conf
cp httpd-default_1.conf httpd-default.conf
sudo /opt/lampp/lampp reloadapache
echo "Load settings by default!"
fi
done
sleep 20
done
```

В дисертационния труд са представени и основни параметри за подобрене управлението на файловата система – swappiness и file-max. Разгледани са и допълнителни механизми, като например обновяване на хардуерната конфигурация, подобряване настройките на браузъра и кеширане на данни от страна на браузъра.

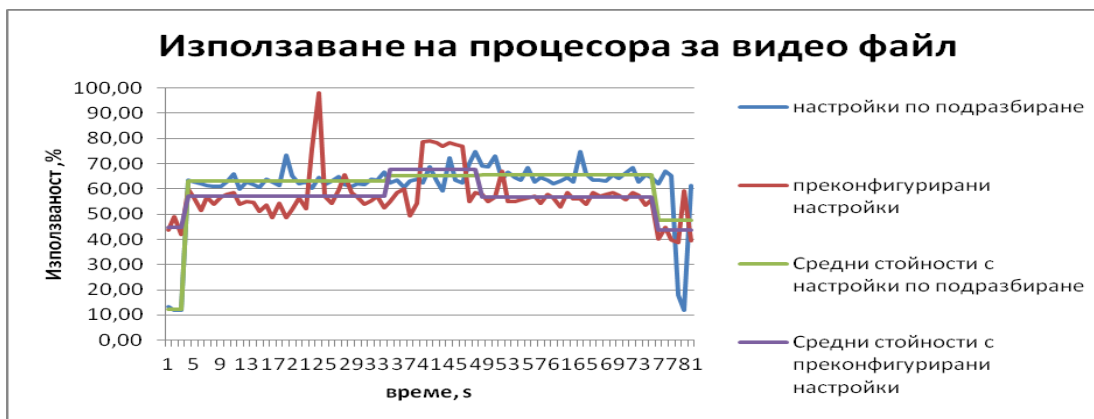
Една от основните задачи на дисертационния труд е да сравнени ефективността на реализираните преконфигурационни настройки. За да се представят по-добре и по-ясно резултатите от подобренията за системата се изследва работа на компютърната система при обслужване на заявки към база данни с нов сценарий на изпращане. Входният сигнал е във вид на импулс, генериран от програмата за симулиране на реални клиенти.

От изследването става ясно, че осреднените стойности за използване на процесора, с настройки по подразбиране е 3,53 %, а с преконфигурирани настройки на системата то е 5,68 %. По този начин се определя началната точка на работа на процесора, при която започва изпращането на заявки. В този момент чрез генератора на входен сигнал се симулира едновременно изпращане на заявки от три клиента към системата. В този период на изпращане на заявки средното използване на процесора е 92,44% при настройки по подразбиране и 88,54% при преконфигурирани настройки. Изпращането на заявки с такава големина трае 20 секунди, след което се стартира симулиране на изпращане на заявки от пет клиента едновременно, като в този период средното използване на процесора при настройки по подразбиране е 99,28%, а с подобрени настройки е 96,39%. Изследван е и трети интервал, в който отново се изпращат заявки от три клиента едновременно, като при него използването на процесора с настройките по подразбиране е 99,50% и 91,24% с подобрените настройки. Използването на процесора в трите интервала намалява съответно с 3,90%, 2,89% и 8,26%.



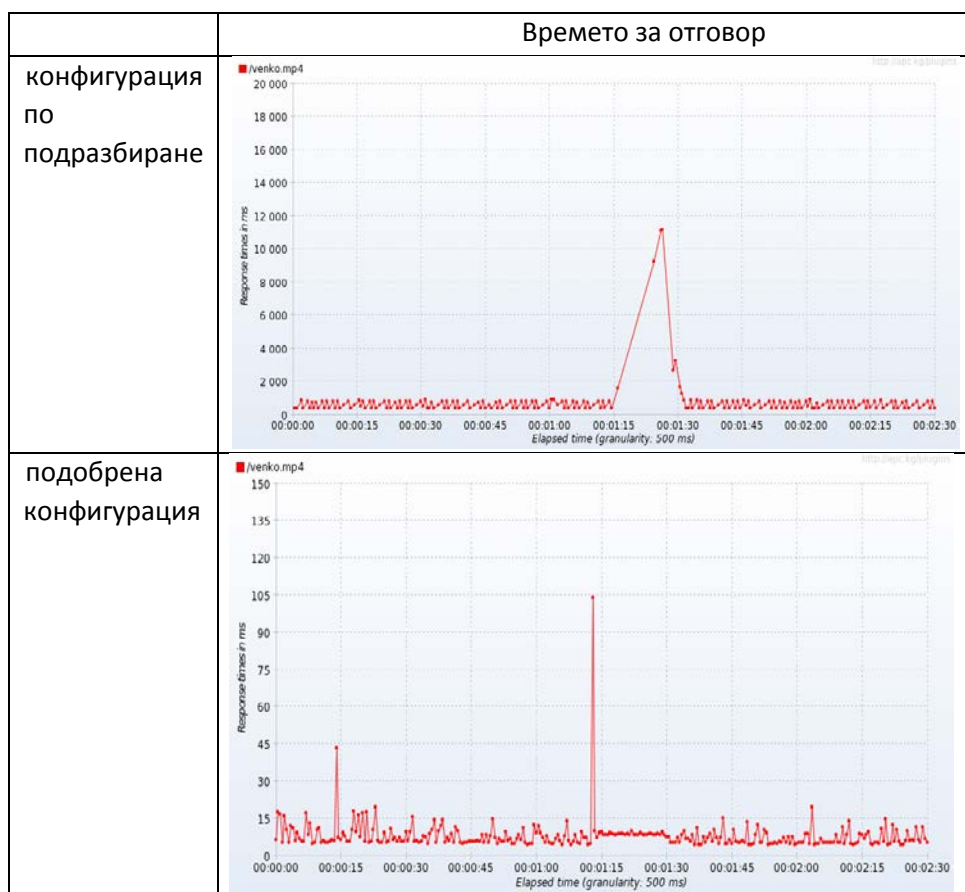
Фиг. III.23. Сравнение на използването на процесора при работа с база данни

Направените настройки на конфигурационния файл са приложени и за сравнение с резултатите за видео файл. Средното използване на процесора при начални условия, когато изследването се извършва с настройки по подразбиране е 12,35%, а при преконфигурирани настройки е 44,82%. Изпращането на заявки се осъществява по следния сценарий: в начало се изпращат заявки от един клиент, след което се увеличава генерирането на заявки от 10 клиента едновременно, и последващо изпращане на заявки отново от един клиент. При такъв сценарий на изпращане на заявките може да се направи заключението, че средното използване на процесора в първия интервал на изпращане на заявки е намаляло с 38,20%, във втория – с 29,91%, а при следващото изпращане на заявка от един клиент намаляло с 41,27%.



Фиг. III.24. Сравнение на използването на процесора при работа с видео файл

Подобренията от приложените настройки на конфигурационния файл на Apache и TCP протокола се потвърждават и от графиките за времето за отговор. То намалява около 10 пъти при нововъведената конфигурация, което представлява значителен принос на изследването.



Фиг. III.25. Време за отговор при изследване на видео файл

От двете представени сравнения на резултатите – за база данни и за видео файл става ясно, че преконфигурирането на TCP протокола и на конфигурационния файл на Apache водят до подобрене в използването на процесора и до намаляване на времето за отговор на системата. Потози начин се доказва, че предложените в дисертационния труд промени в настройките на уеб сървъра Apache и комуникационния протокол TCP са ефективно решение за създаденото софтуерно приложение, което поддържа база данни с научни статии и видео файлове от семинари, конференции и обсъждания.

В заключение, глава трета на дисертационния труд демонстрира приложимостта на теория на автоматичното управление с цел повишаване ефективността на програмни системи. Експерименталните резултати показват, че предложените промени в TCP комуникационния протокол и в конфигурационния файл на Apache, водят до подобрене в показателите за качество на производителността – натоварване на процесора, натоварване на паметта и време за отговор.

Глава 4.Повишаване на производителността на компютърна система чрез прилагане на методи от теория на управлението

В последната глава на дисертационния труд е представено прилагането на методите за идентификация за конкретна компютърна система. По този начин се цели да се

определят динамичните характеристики на системата чрез показатели на производителността ѝ. Представените характеристики се апроксимират и се изграждат модели на системата. Тяхната ефективност се изследва чрез метода на контролните карти, за да се определи приложимостта им. Реализираните софтуерни подобрения в изследваната система са моделирани чрез нелинейни модели, поради тяхната гъвкавост относно сложни явления и процеси. Получените модели са симулирани в симулационна среда на MATLAB, за да се изследва поведението на системата при включване на пропорционално-интегрален (ПИ) регулатор за управление на натоварването на процесора.

В първата част на главата са разгледани методите за идентификация, които позволяват да се изградят модели на динамични системи чрез експериментални изследвания. Идентификация на системата е научна област, изследваща изграждането на математически модели на динамични системи чрез наблюденията на входно-изходни данни. Един от начините за идентифициране на обекта за управление е чрез подаване на единично стъпаловидно входно въздействие към него при нулеви начални условия, а реакцията на обекта е неговата преходна характеристика (функция). Получаването на преходната характеристика се осъществява чрез провеждане на експеримент. Целта е събиране на недостъпна и неясна информация за обекта и въз основа на статистически методи да се извърши моделирането и оптимизирането му. При снемане на преходни характеристики експериментално, т.е. реализиране на наблюдение, данните много често са зашумени от влиянието на неконтролируеми въздействия, което изисква осъществяването на експериментално снемане на данните няколко пъти (4-6 пъти).

При моделиране на система за автоматично управление чрез експериментални изследвания за получаване на техните преходни характеристики се използват таблици за записване на получените данни. Целта на тяхното прилагане е нагледност на резултатите и улесняване на последващата обработка на данните.

В таблица IV.1 са показани първите 15 измерени данни за натоварването на процесора ($C_{pi\%1-4}$), изчислените преходни характеристики на натоварването на процесора (Y_{1-4}) и осреднената преходна характеристика при подаване на единично въздействие от 10 потребителя. Изискването, което трябва да се съобрази, е обектът да е в установено състояние. При обекти, при които началният момент е различен от нула, като нашия, се използва следната зависимост [Гарип-07]

$$y_{ij}^0 = y_{ij} - y_{j0}, i=0, 1..(N-1) j=1, 2..n \quad (3)$$

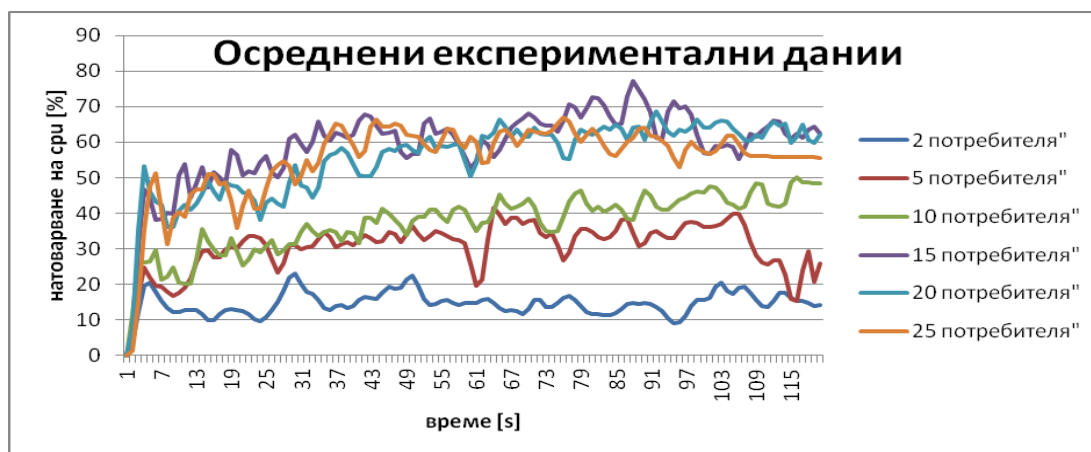
Така преходната функция се премества в началото на координатната система и започва от 0. Резултатите от прилагането на тези действия са показани в колони Y1-4. на таблица IV.1. В последната колона се изчислява средната стойност от четирите експериментално проведени опита.

Този начин на обработване на експерименталните данни се използва за получаване на осреднени преходни характеристики за натоварването на процесора и за бързодействието (получените данни от системата) при обслужването на различен брой клиенти (2, 5, 10, 15, 20, 25).

Табл.IV.1 Получаване на осреднена експериментална преходна характеристика

T[s]	Измерени данни				При нулеви начални условия				$\bar{Y}$
	Cpu% ₁	Cpu% ₂	Cpu% ₃	Cpu% ₄	Y ₁	Y ₂	Y ₃	Y ₄	
1	19,6420	56,6896	62,4040	67,9263	0,0000	0,0000	0,0000	0,0000	0,0000
2	21,8174	59,9757	61,0201	67,6415	2,1754	5,0404	1,6464	1,2349	2,5243
3	27,2572	56,1521	61,0108	67,7781	7,6152	14,6565	5,9758	6,8977	8,7863
4	33,4232	48,7413	62,8116	67,7385	13,7812	24,8439	8,1972	12,5882	14,8526
5	38,8170	43,8372	63,5706	66,5161	19,1750	32,9478	9,7298	15,5006	19,3383
6	43,8703	42,8724	61,2725	62,6580	24,2283	36,2339	13,6123	19,3758	23,3625
7	49,0946	43,6702	57,4372	57,4163	29,4526	32,4102	21,0318	27,9603	27,7137
8	54,6025	44,2807	55,6166	55,6947	34,9605	24,9995	29,9253	37,4323	31,8294
9	59,9135	44,6167	58,2741	60,0223	40,2715	20,0954	35,9234	42,2437	34,6335
10	62,7727	45,7113	64,4806	66,5499	43,1307	19,1306	36,9471	42,7508	35,4898
11	61,8238	47,0324	70,0408	69,6327	42,1818	19,9284	35,5633	42,4660	35,0348
12	59,2710	47,4592	71,4321	68,3844	39,6290	20,5389	35,5539	42,6026	34,5811
13	56,8640	48,4573	69,3436	66,4653	37,2220	20,8748	37,3547	42,5630	34,5036
14	53,4603	52,3111	67,1831	66,7838	33,8184	21,9694	38,1138	41,3406	33,8105
15	48,6432	58,0175	66,7838	68,1800	29,0012	23,2906	35,8156	37,4825	31,3975

От показаните на фиг.IV.2 осреднени преходни характеристики се вижда, че те се разделят на две групи – когато изследваната системата обслужва по-малко или повече от 15 потребителя. За да се изведат обобщени преходни характеристики е необходимо получените данни да имат близки стойности. Въз основа на това изискване се изгражда модел, който ще определи зависимостта между стойностите по оста x и тези по оста y. Полученият модел за натоварването на процесора на изследваната в този дисертационен труд система се използва за прилагане на допълнителни настройки, чрез които да се постигне по-добро обслужване на потребителите на системата. Натоварването на система в голяма степен зависи от броя потребители на системата. Затова разделянето на модела на две нива е подходящо решение. От друга страна, честата промяната на настройките на системата изисква допълнителни изчисления и хардуерен ресурс, а по този начин се ограничават изчисленията и се намалява използвания ресурс за настройка.



Фиг. IV.2 Осреднени експериментални данни за натоварването на процесора

Тенденцията в данните най-добре се описва с алгебричен полином от трета степен. Редът на полинома, който апроксимира данните се определя от броя на флуктуациите в данните или от минимумите и максимумите на кривата. Той описва модела с висок коефициент детерминация и е по-лесно реализируем от полиноми от по-висока степен. Осъществената апроксимация на експерименталните данни с полином от трета степен за по-малък брой потребители е:

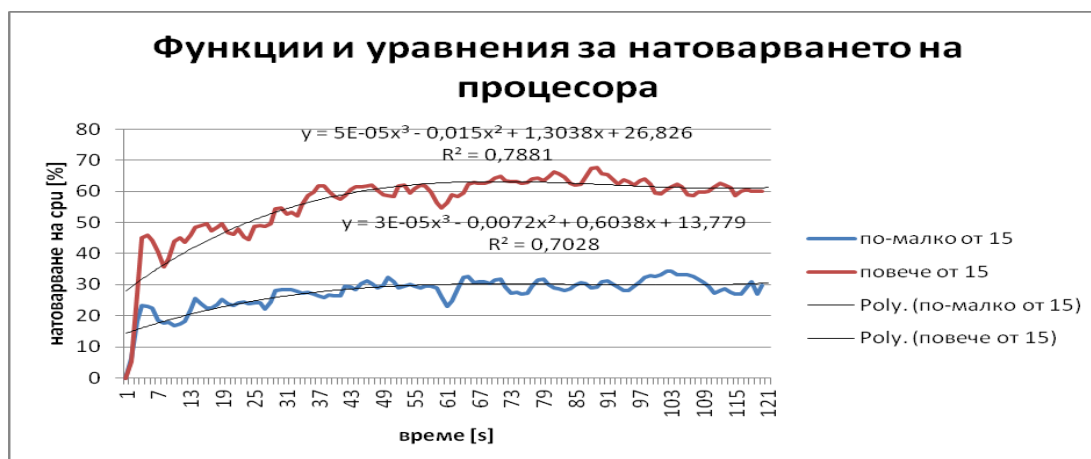
$$y = 0,0003x^3 - 0,0072x^2 + 0,6038x + 13,779 \quad (8)$$

Коефициентът на детерминация на получения модел е $R^2 = 0,7028$. Това означава, че 70,28% от данните за натоварването на процесора се изменят във времето според установения полином от трети ред при по-малко от 15 потребителя на разработеното софтуерно приложение.

Апроксимация на експерименталните данни с полином от трета степен за повече от 15 потребителя се описва с:

$$y = 0,00005x^3 - 0,015x^2 + 1,3038x + 26,826 \quad (9)$$

Коефициентът на детерминация е $R^2 = 0,7881$. Това означава, че 78,81% от данните за натоварването на процесора се изменят във времето според установения полином от трети ред при повече от 15 потребителя на разработеното приложение.

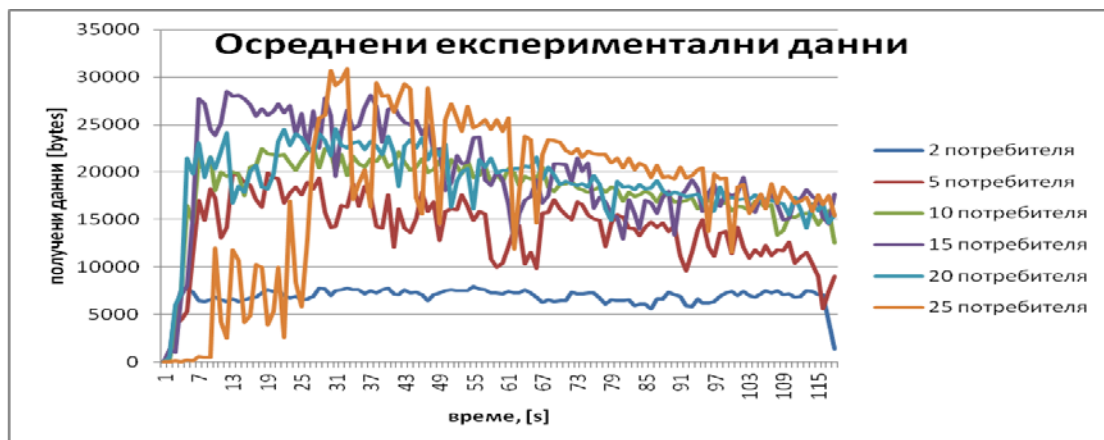


Фиг. IV.3. Функции и уравнения за натоварването на процесора

Системата бе изследвана допълнително чрез анализиране на данните от втори параметър, за да се установи зависимостта на количеството получени и изпратени данни от системата, и броят потребители на системата. Целта е да се изгради модел на неговото поведение в зависимост от броя потребители и да се получи по-ясна представа за бързодействието ѝ.

Осреднените преходни характеристики за този параметър се наблюдават с по-голяма колебливост и това явление най-ясно проличава при експериментите с 25 потребителя. Представените експериментални преходни характеристики за бързодействието на системата имат по-близки стойности, когато потребителите са повече от 15. При по-малък брой потребители стойностите им се отдалечават. При определянето на обобщена преходна характеристика е необходимо получените експериментални данни да имат

близки стойности. Въз основа на резултатите за получените осреднени експериментални данни от програмата Jmeter и посоченото условие за близки стойности, отново се изгражда модел на две нива.



Фиг. IV.4. Осреднени експериментални данни за бързодействието на системата

Моделите на бързодействието на системата се описват с алгебричен полином от четвърти ред поради по-големия брой на флуктуациите в данните. Полиномът, който описва модела за получените данни за повече от 15 потребителя е:

$$y = -0,0012x^4 + 0,362x^3 - 38,702x^2 + 1599,9x + 1539,6 \quad (10)$$

Коефициентът на детерминация е $R^2 = 0,8225$, което означава, че 82,25% от данните за бързодействието на системата се изменят във времето според установения полином от четвърти ред при повече от 15 потребителя на разработеното приложение.

Полиномът, който описва модела за получените за по-малко от 15 потребителя е:

$$y = -0,0012x^4 + 0,3317x^3 - 30,362x^2 + 1048,9x + 4112,5 \quad (11)$$

Коефициентът на детерминация е $R^2 = 0,733$, което означава, че 73,3% от данните за бързодействието на системата се изменят във времето според установения полином от четвърти ред за по-малко от 15 потребителя на разработеното приложение.



Фиг.IV.5. Функции и уравнения на бързодействието на системата

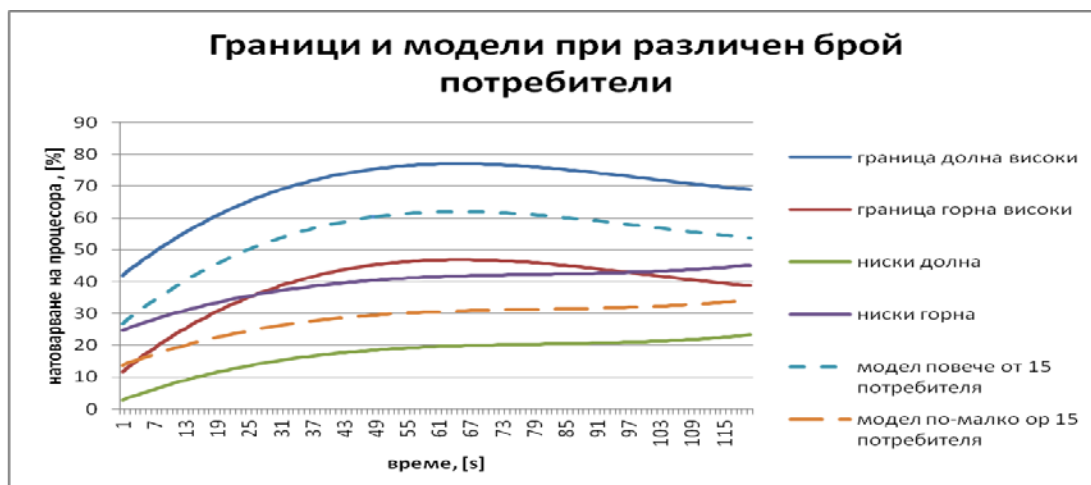
При измерване показателите на натоварване на процесора и бързодействие при подаване на единично входно въздействие във вид на брой потребители на уеб сайт, бе установено, че реакцията на изследвания обект се променя при увеличаване на броя на потребители на софтуерната система. За описанието на подобни обекти, които си изменят преходните характеристики при промяна на входните въздействия, се използват системи с твърда адаптация.

Поради тази причина във втората част на четвърта глава от дисертационния труд са разгледани системите с твърда адаптация. При тях изменението на параметрите на регулатора се осъществява въз основа на модела на обекта. Чрез измерване на работните натоварвания на променливите на обекта на процеса или въздействащите смущения се установява промяната на модела. По този начин може много бързо системата да реагира на променените натоварване на процесора и бързодействие на системата. Системите с твърда адаптация могат много добре да се приложат и за компютърни системи, понеже при изменение на броя потребители се променят и характеристиките на обекта – в случая уеб сървър Apache. Идентификацията се осъществява чрез измерване натоварването на процесора, а настройките на регулатора се реализират чрез промяна на конфигурационния файл на Apache. Използваната операционна система Linux и уеб сървър Apache имат много добре направени настройки за саморегулиране по подразбиране. По този начин може много бързо системата да реагира при променено натоварване на процесора и бързодействие на системата. За прилагане на системи с твърда адаптация е необходимо да се изчислят показателите за качеството на измерваните параметри с цел определяне на състоянията, при които тя трябва да бъде прилагана.

За изчисляване показателите за качество в третата част на главата се използва методът на контролните карти. Чрез него се наблюдават текущите стойности на качествените показатели на даден продукт. Той позволява бързо вземане на решение с цел коригиране и избягване на отклонения от желаните стойности. При метода на контролните карти се използва правилото на трите сигми, т.е. стойности на показателя на качеството x имат нормален закон за разпределение и попадат в интервала $m_x \pm 3\sigma$ с вероятност 0,9973, където m_x е математическото очакване на показателя, а σ стандартното му отклонение.

Тук приемаме, че показателите за нашата изследвана система са натоварването на процесора и постъпилите данни от потребителите, т.е. бързодействието на обслужване. Техните данни се изчисляват в интервал $\pm\sigma$, поради близостта на получените данни и възможността те да бъдат обработвани с по-малка вероятност. Изчислени са стандартни отклонения за двата модела при повече и при по-малко от 15 потребителя. Тези стойности се използват за определяне на граници, в които да се приложат изчислените модели.

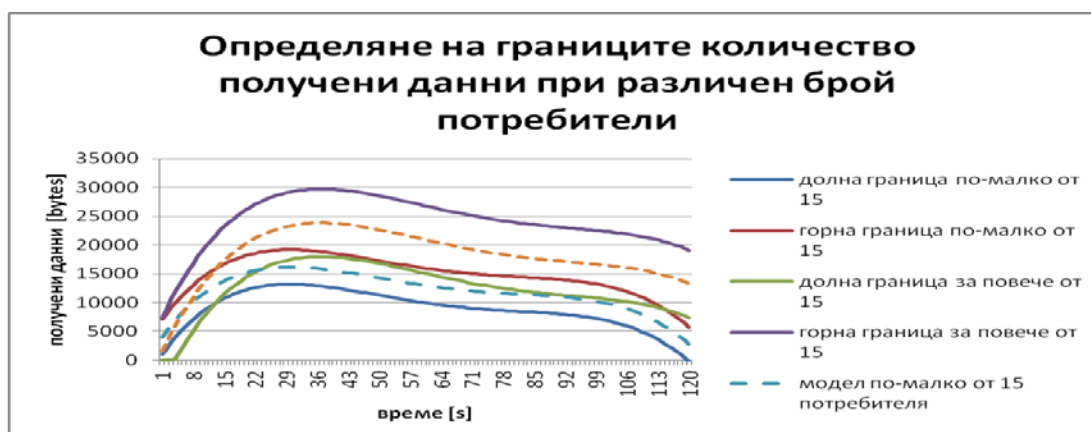
На фиг.IV.7. са показани графично функциите, изчислени за моделите за натоварването на процесора и за бързодействието на система, съвместно с изчислените стандартни отклонения на експерименталните данни. Това позволява да се определят границите на действията на подобренията на конфигурационните настройки на системата.



Фиг.IV.7. Функции на моделите и границите на показателя на натоварване на процесора.

Вижда се, че долната граница на модела за повече от 15 потребителя в два момента пресича горната граница на модела за по-малко от 15 потребителя. Освен това се наблюдава един малък интервал от пространството, който не се покрива от нито един от двата модела. За да се отстранят тези недостатъци на моделите може да се приложат допълнителни изчисления като например осредняване стойностите на двете граници.

Относно показателя бързодействие на системата бе установено, че определената долна граница за повече от 15 потребителя е по-малка от горната граница за по-малък брой потребители. Затова е необходимо или да се намали областта, в която да се проследява количеството постъпили данни от потребителите на системата, или да се използва само горната граница за потребители по-малко от 15. Трето възможно решение е да се изчисли средна стойност за двете граници.



Фиг.IV.8 Функции на моделите и границите на бързодействие на системата.

В последната част на главата е представен нелинеен анализ на приложеното управление на програмната система. За да се представи реализираното софтуерно управление чрез прилагане принципите за автоматично управление се използва инструмента на MATLAB – Systemidentificationtools, който позволява чрез MATLAB функции симулационни блокове да се изградят математически модели. Той е инструмент, който има широко приложение за идентифицирането на времеви и честотни

входно-изходни данни, непрекъснати и дискретни предавателни функции и др. Освен това позволява да се представи динамиката на нелинейни системи чрез изчисляване на Хамерщайн-Винер модели или нелинейни ARX модели с вълновидни мрежи, дървовидни части и сигмоидни мрежови нелинейности. Те предоставят по-голяма гъвкавост за установяване на сложни явления.

Поради факта, че изключително голям брой процеси в компютърните системи са нелинейни, използването на нелинейна ARX структура за тяхното моделиране е изключително лесно, понеже се определят единствено редовете на полиномите A, B и времезакъснението p_k за системата. В ARX нелинейната структура използването на нелинеен оценител с вълновидна мрежа често предоставя задоволителни резултати и се използва като модел за бърза оценка. Редът и закъснението на нелинейните ARX модели са целочислени и положителни стойности, въвеждани чрез параметрите p_a , p_b , p_k , където p_a определя броя изходни стойности за предвиждане на текущата стойностна модела, p_b - определя броя на входните стойности, а с p_k се определя закъснението от стойности.

Нелинейните ARX модели трябва да се разглеждат като разширение на линейните ARX модели на системи с един вход и един изход (SISO ARX).

Нелинейният ARX модел използва по-гъвкава нелинейна функция за описание на изследваната система

$$y_p(t)=f(y(t-1),y(t-2),y(t-3),...,u(t),u(t-1),u(t-2),...) \quad (15)$$

където f е нелинейна функция. Входните данни за f са регресорите на модела. В структурата на нелинейните модели се избира нелинейна функция за нелинейния оценител. При нелинейните ARX модели регресорите могат да бъдат входните и изходните закъснения или по-сложни техни зависимости.

Нелинейните ARX модели се представят чрез регресории нелинеен оценител, който има линейна и нелинейна функция.

Нелинейните оценители представляват нелинейна функция като сума от последователни нелинейни елементи. Оценяващият алгоритъм във вълновидната мрежа определя броя на използваните елементите автоматично. Той използва следната формула за определяне на функцията $g(x)$:

$$g(x) = \sum_{k=1}^n \alpha_k k(\beta_k(x - y_k)) \quad (17)$$

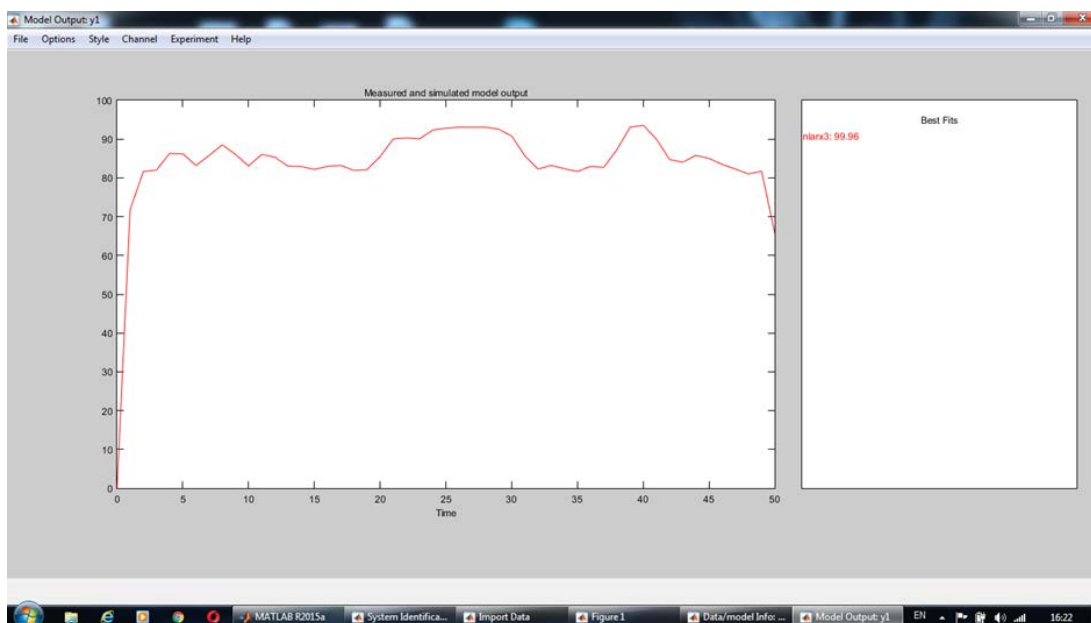
където $k(s)$ е вълновидна функцията. Определянето на регресорите на модела се осъществява чрез задаване на степенния ред на полинома и закъснението на модела, които създават поредицата от стандартни регресори на модела.

Използването на нелинейните ARX модели цели да се представят модели на подобренията, които са реализирани чрез софтуерните приложения: TCP комуникационния протокол, конфигурационния файл на уеб сървъра Apache програмата за автоматичното им преконфигуриране, всички работещи с операционната система Linux.

В създадените в хода на дисертационното изследване модела входни стойности са използвани данните от натоварването на процесора от програмната система, преди да се извършат промените в гореспоменатите приложения, а изходните – след извършените

промени. Данните, използвани за изграждане на моделите, са получени при различен брой потребители на разработеното софтуерно приложение за електронно разпространение на научно списание, когато се използва видео файл или база данни.

За целите на настоящия дисертационен труд първо е изследвано поведението на използваната програмна система с нелинеен модел за три и пет потребители на база данни на разработеното софтуерно приложение за натоварване на процесора при обслужването им. На фиг.IV.10 са представени експерименталните данни и изградения нелинеен ARX модел с вълновидна мрежа за оценяване на нелинейността, след като са приложени настройките към изследваната система. Моделът е изграден чрез инструмента за идентификация Identification tool на MATLAB. Достоверността на модела е 99,96%. Избраните степенни редове на полиномите са $na=3$, $nb=3$, $nk=1$.



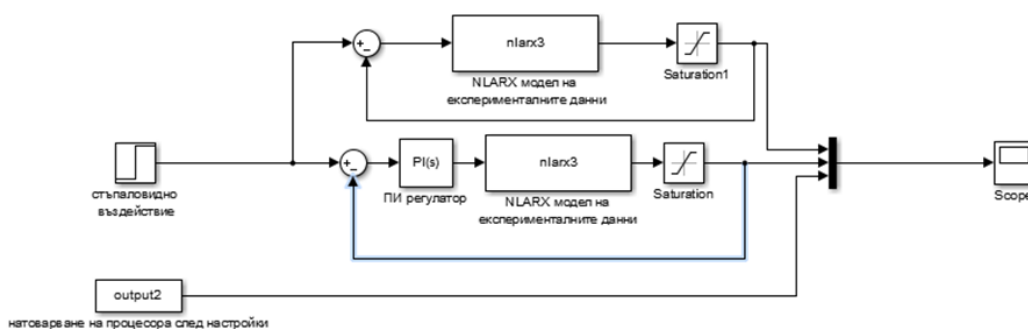
Фиг.IV.10.Нелинеен ARX модел на изследваната система при три и пет потребителя набаза данните

Полученият модел е изследван в симулационна среда на MATLAB. На фиг.IV.11 е показана симулационната схема, чрез която се проучват резултатите, получени от приложеното софтуерно управление. Чрез нея се изследва приложимостта на линейни регулатори в използваната програмна система. Задачата, която решава регулаторът, е да се поддържа максимално близко управляваната променлива (натоварването на процесора) до зададената променлива (стъпаловидното въздействие). Това се постига чрез сравняване на двете стойности и се подава на изхода на регулатора манипулираната стойност. В зависимост от желаните изходни стойности, които се изисква да се постигнат след действието на регулатора, се използват различни начин за управление.

За да се симулира управление в изследваната система се използва ПИ регулатор. Пропорционално-интегралният регулатор позволява бърза реакция и компенсиране на смущенията в установен режим.

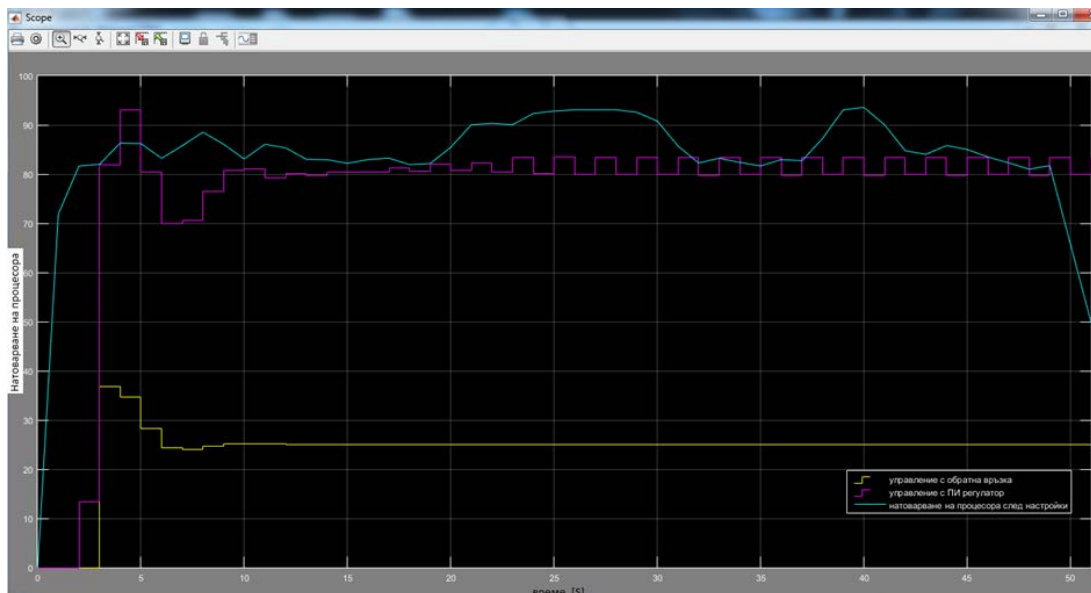
На симулационната схема (фиг.IV.11) (output2) е изходният сигнал от експериментално получените данни от натоварването на процесора след извършените

настройки в използваните програми за управление на системата. Те се извличат от текстови файл. Симулирано е управление в затворена система за автоматично управление с пропорционално-интегрален регулатор и е представено негово влияние за програмна система, която е моделирана с нелинеен ARX. Настройката на ПИ регулатора е процес, при който се определят стойностите на П и И компонентите на регулатора за постигане на желаното качество на процеса с цел да се отговори на поставените изисквания. За определяне на тяхните стойности се използва инструментаPIDTuner, който подпомага интерактивното настройване на регулатора за постигане на желаните настройки. Той позволява лесно да се определят стойностите на компонентите на регулатора за постигане на желаното качество на изходния сигнал.



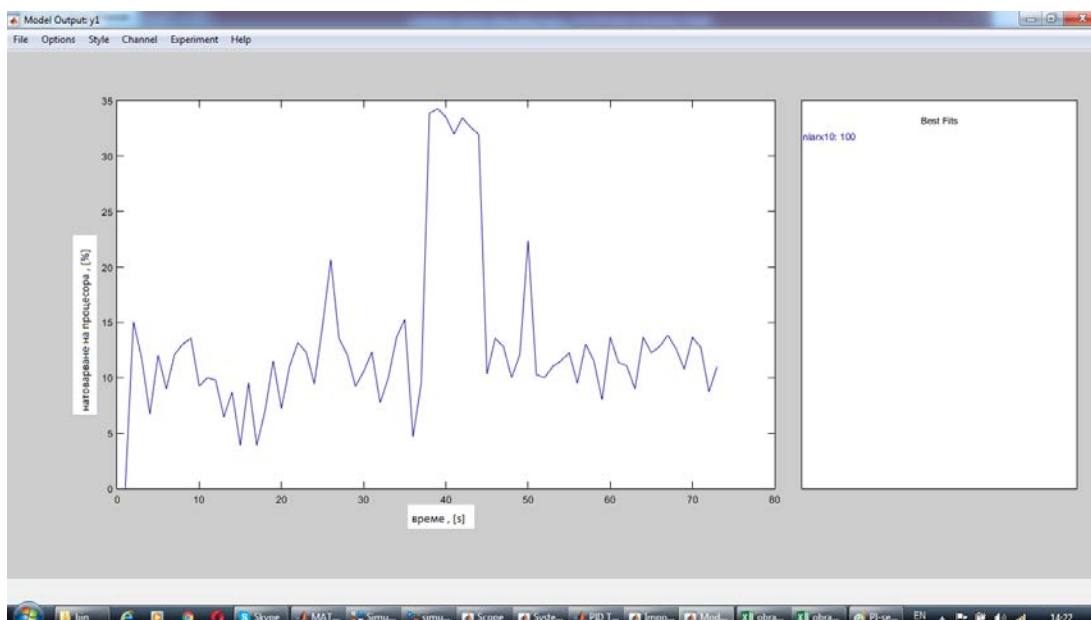
Фиг.IV.11. Симулационен модел на изследваната система

На фиг.IV.12 са показани получените резултати от симулационния модел. Зелената линия отразява данните, получени след прилагане на подобрените настройки на системата, а жълтата линия представя управление само с обратна връзка без регулатор. Лилавата линия показва прилагането на ПИ регулатор в система с обратна връзка с пропорционална съставка $P=1,2201$ и интегрална съставка $I=0.53072$. Стойностите на ПИ регулатора са получени чрез инструмента за настройване на регулатори PIDtuner. Чрез него експериментално се постига по-бърза реализация на системата и допълнително изглаждане на получения изходен сигнал.



Фиг.IV.12 Получените резултати от симулационния модел при три и пет потребителя на база данните

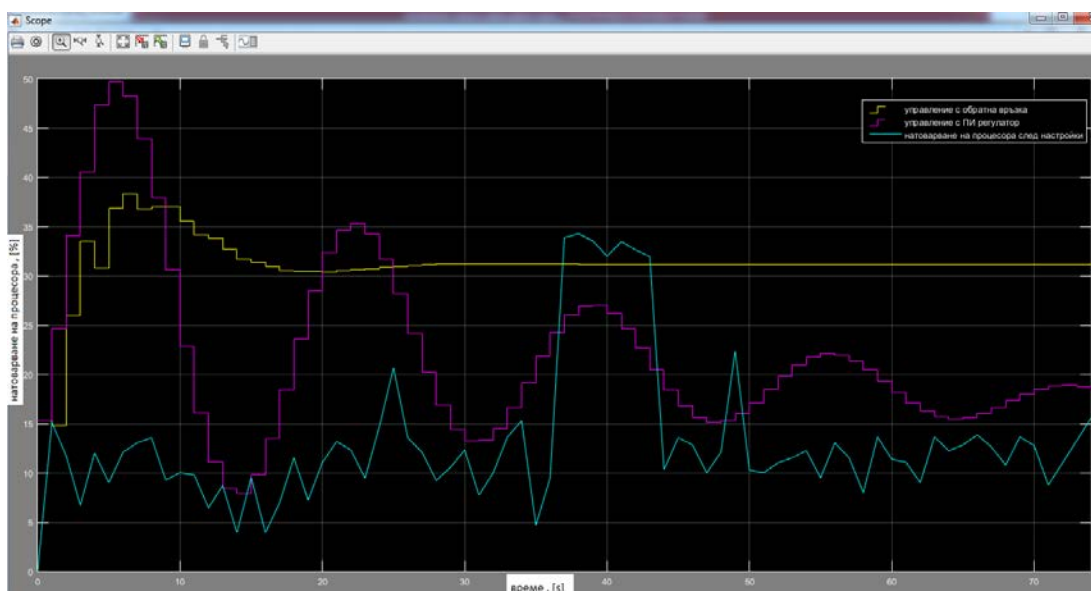
В хода на дисертационния труд бяха изследвани реализираните подобрения в системата чрез нелинеен ARX модел и при използване на видео файл от потребители на софтуерното приложение. Моделът е изграден чрез инструмента за идентификация Identification tool на MATLAB. Той описва натоварването на процесора от програмната система с достоверност 100%. Избраните степенни редове на полиномите са $na=6$, $nb=4$, $nk=1$. Измерените изходни данни и симулирания нелинеен модел, получен чрез MATLAB, са показани на фиг. IV.15.



фиг.IV.15. Нелинеен arx модел при един и десет потребителя на видео файл

Полученият модел е изследван чрез симулационната схема показана по-горе (на фиг.IV.11). Чрез нея се симулира прилагането на регулатор за натоварването на процесора

при обслужване на видео файлове. Резултатите от реализирана симулация са представени на фиг.IV.16. На нея зелената линия показва данните, получени след прилагане на настройките на системата, а жълтата линия представя управление само с обратна връзка без регулатор. Лилавата линия показва прилагането на ПИ регулатор в система с обратна връзка с пропорционална съставка $P=-0,04992$ и интегрална съставка $I=-0.048602$. Стойностите на ПИ регулатора са получени чрез инструмента за настройване на регулатори PID tuner. Чрез него експериментално се постига по-бърза настройка на системата и допълнително изглаждане на получения изходен сигнал.



Фиг.IV.16. Получени резултати от симулационната схема за видео файл при един и десет потребителя

От направените симулационни изследвания може да се изведе заключението, че използването на ПИ регулатор за управление на показателя за производителност – натоварването на процесор, който се моделира с нелинеен ARX модел, ще намали натоварването на процесора при увеличаване на броя потребители на софтуерното приложение.

В обобщение, четвърта глава на дисертационния труд представено прилагането на методи за идентификация за конкретна компютърна система. По този начин се целеше да се определят динамичните ѝ характеристики чрез показатели на производителността. Получените стойности на характеристиките бяха апроксимирани за изграждането на полиномни модели на системата. Тяхната ефективност се оценява чрез метода на контролните карти, за да се определи приложимостта им. Изчислени бяха и средни стойности за натоварването на процесора в зависимост от броя потребители на софтуерното приложение и бе определена квадратична функция, която позволява да се адаптират подходящи настройки.

Реализираните софтуерни подобрения в изследваната система бяха моделирани чрез нелинейни модели, поради тяхната гъвкавост относно сложни явления и процеси. Получените модели бяха симулирани в симулационна среда на MATLAB, за да се

изследва поведението на системата при включване на пропорционално-интегрален (ПИ) регулатор за управление на натоварването на процесора. Така се демонстрира, че неговото използване ще подобри работа на изследваната система при обслужването на по-голямо брой потребители на софтуерното приложение.

Четвърта глава на дисертационния труд също така демонстрира приложимостта на нелинейни ARX модели, за да представи реализираното софтуерно подобрение и да се разгледа възможността за допълнително управление чрез ПИ регулатори. От експерименталната работа може да се заключи, че при потребление на видео файлове и база данни е подходящо да се реализира ПИ управление за изследваната система.

Приноси на дисертационния труд

1. Разработен е функционален модел чрез прилагане на формалния апарат на теорията на автоматичното управление за компютърна система.
2. Реализирано е управление в изследваната компютърна система като се извършва синтез на параметрите на TCP комуникационния протокол, уеб сървър Apache и операционната система.
3. Експериментално са определени динамичните характеристики на компютърната система при управление на уеб сървър Apache и са изведени техните математически зависимости, описващи нейната работа.
4. На базата на стандартни показатели за качество на компютърната система е изследвана ефективността на приложените настройки за системата.
5. Предложен е подход за настройка на регулатори с пропорционално-интегрално действие – с цел постигане на по-добро качество на управление в затворения контур.

Апробация на резултатите

Резултати, получени в хода на работата по дисертационния труд, са оповестени и дискутирани по време на следните конференции, семинари и работни срещи:

- Работна среща (Workshop) по проект BG051PO001-3.3.06.0048 "Изграждане и развитие на млади висококвалифицирани изследователи за ефективно прилагане на биомедицински изследвания за подобряване качеството на живот" 26.04.2013г., ИЕМПАМ-БАН, бл. 25.
- International Conference Computer Systems and Technologies, CompSysTech'13, 28-29, June 2013, Rouse, Bulgaria
- Първа национална тематична школа и борса за научни идеи в областта на информационните и комуникационните технологии, 27-28 юни, 2013, гр. Русе.
- Международна конференция „Автоматика и информатика”, 03-07 октомври 2013, гр. София
- VII Национална студентска научно-техническа конференция, 23-27 септември 2011г., гр. Созопол

Резултати, включени в дисертационния труд са обсъждани на работни семинари на секция „Йерархични системи“ към ИИКТ-БАН и катедра „Информатика“ при МГУ „Св. Иван Рилски“.

Списък на публикациите по темата на дисертационния труд

1	[DimiT-13]	Dimitrov S., T.Stoilov. Loading test of the Apache HTTP Server by videofile and usage measurements of the hardware components. Proceedings of the International Conference Computer Systems and Technologies, CompSysTech'13, 28-29 June 2013, Rouse, ACM Conference Proceeding Series, pp.59-66
2	[Димит-13]	Димитров, Ст., Тест за натоварване на Apache HTTP Server чрез различни видове файлове и измерване използването на хардуерните компоненти, Първа национална тематична школа и борса за научни идеи в областта на информационните и комуникационните технологии, 27-28 юни, 2013, Русе, ISSN 1314-9024, стр. 212-217
3	[Димит-13a]	Димитров, Ст., Информационна система за електронен абонамент на научно списание, Международна конференция „Автоматика и информатика“ 2013 3-7 Октомври, София, ISSN 1313-1850 2013 стр. I-185 - I-188
4	[Димит-11]	Димитров, Ст., Разглеждане на Apache HTTP Server като система за автоматично регулиране и извършване на тестове с Jmeter , VII Национална студентска научно-техническа конференция 23-27.09.2011г., гр.Созопол, ISSN 1314-0442, 2011, стр. 263-269

Библиография

	[ApaSF-16]	Apache Software Foundation, Apache Jmeter – User's Manual, Building a web testplan, 2016, http://jmeter.apache.org/usermanual/build-web-test-plan
	[ArnoA-94]	Arnold O. A., Computer Performance Analysis with Mathematica, Academic Press, 1994. \$1.1 Introduction, pg 1.
	[Avoya-11]	Hovhannes Avoyan, (2011), 25 Apache Performance Tuning Tips, http://www.monitis.com/blog/2011/07/05/25-Apache-performance-tuning-tips/
	[BoGMT-05]	G. Bolch, S. Greiner, H. de Meer and K. S. Trivedi, Queueing networks and Markov chains: Modeling and Performance Evaluation with Computer Science Applications, Wiley-Interscience, 2005.
	[DGHPT-02]	Diao Y., Gandhi N., Hellerstein J.L., Parekh S., Tilbury D., Using MIMO feedback control to enforce policies for interrelated metrics with application to the Apache Web server. IEEE/IFIP Network Operations and Management, April 2002
	[Garre-09]	Russ Garrett, (2009), Linux Kernal Tuning, http://russ.garrett.co.uk/2009/01/01/linux-kernel-tuning/
	[GheGF-13]	Gheorghe G. F., Web Application Performance Testing using Apache JMeter2 March 2013, http://grosan.co.uk/web-application-performance-

		testing-using-Apache-jmeter-part-2-monitor-server-resources/
	[Gregg-13]	Gregg B., Systems Performance: Enterprise and the Cloud 1st Edition, Prentice Hall, 2013, ISBN-10: 0133390098
	[HTCAG-10]	High Throughput Computing Administration Guide, (2010) https://computing.llnl.gov/linux/slurm/high_throughput.html
	[MDVHR-01]	R. D. Van der Mei, R. Hariharan and P. Reeser, "Web Server Performance Modeling, "Telecommunication Systems, Vol. 16, No. 3-4, 2001, pp. 361-378. doi:10.1023/A:1016667027983
	[Moura-00]	G. Mourani, (2000), Securing and Optimizing Linux RedHat Edition -A Hands on Guide, chapter 6 Linux General Optimzation, http://tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap6sec75.html
	[OracC-00]	Oracle Company, Oracle9i Application Server Oracle HTTP Server powered by Apache Performance Guide, 2000, https://docs.oracle.com/cd/A95427_01/httpperf/listener.htm
	[SoftP-15]	SoftPanorama, Linux TCP Performance Tuning, 20 october 2015, http://www.softpanorama.org/Commercial_linuxes/Performance_tuning/TCP_performance_tuning.shtml
	[TricE-11]	Trichkova, E., "PHPbasedsite-templatefordistancelearninginmedicaldisciplines", ProceedingsofE-Learning'11 conference - E-LearningandtheKnowledgeSociety, 25-26 August 2011, Bucharest, Romania, pp. 87-92, ISBN 978-606-505-459-2
	[WigeN-13]	Wiger N, Linux Network Tuning for 2013, 6 Apr 2013, http://www.nateware.com/linux-network-tuning-for-2013.html
	[Гарип-07]	Гарипов, Е., Идентификация на системи: I част (Идентификациячрез непрекъснати модели) (трето допълнено и преработено издание).ТУ - София, 2007, 133 стр
	[СапуГ-07]	(Георги Сапунджиев, Информационни системи в индустрията ISBN 954-438-481-2)