SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta elektrotechniky a informatiky
Ústav telekomunikácii

Ing. Ján Vrabec

Autoreferát dizertačnej práce

MERANIE ZAŤAŽENOSTI SYSTÉMOVÝCH PROSTRIEDKOV ALGORITMAMI NA ROZPOZNÁVANIE REČI

na získanie akademického titulu doktor (philosophiae doctor, PhD.)

v doktorandskom študijnom programe: Telekomunikácie,

v študijnom odbore 5.2.15 Telekomunikácie

Bratislava, jún 2012

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia na oddelení teórie oznamovacej elektrotechniky na Ústave Telekomunikácii na Fakulte elektrotechniky a informatiky na Slovenskej technickej univerzite v Bratislave

Predkladateľ: **Ing. Ján Vrabec**
Oddelenie teórie oznamovacej elektrotechniky
Ústav telekomunikácii
FEI STU
Ilkovičova 3
812 19 Bratislava 1

Školiteľ: **doc. Ing. Gregor Rozinaj PhD.**
Oddelenie teórie oznamovacej elektrotechniky
Ústav telekomunikácii
FEI STU
Ilkovičova 3
812 19 Bratislava 1

Oponenti: .....
.....

.....
.....

Autoreferát bol rozoslaný: .....

Obhajoba dizertačnej práce sa koná ..... 2012 o .....h na Fakulte elektrotechniky a informatiky na Slovenskej technickej univerzite v Bratislave, Ilkovičova 3, 812 19 Bratislava 1.

.....

doc. RNDr. Gabriel Juhás, PhD
dekan

Obsah

1 Úvod 1

2 Ciele 1

3 Metodika na meranie záťaže aplikáciami pre multimodálny interface 2

4 Metodológie na elimináciu rušení pri meraní 5

5 Vyhodnocovacia metóda, ktorá vypočíta a zobrazí zaťaženie daným algoritmom 7

6 Overenie metodológie na elimináciu rušení pri meraní aplikácie 9

7 Dosiahnuté výsledky 14

8 Použitá literatúra 14

9 Vlastne publikácie vzťahujúce sa k dizertácii 17

10 Ohlasy 18

11 Resumé 19

1 ÚVOD

V súčasnej dobe vedci a výskumníci navrhujú stále zložitejšie algoritmy na spracovanie obrazu, reči ako aj simulácie. Pri ich návrhu a testovaní sa pracuje na výkonných pracovných staniciach, ktoré majú prakticky neobmedzený výpočtový výkon. Ak však má byť algoritmus používaný v reálnom prostredí sú na neho kladené vysoké požiadavky a preto je nevyhnutne vedieť jeho systémové požiadavky.

Každý algoritmus alebo simulácia má svoje jedinečné hardvérové nároky. Ak sú primalé alebo nedostatočné spomaľuje sa vykonaná operácia. V prípade aplikácií, ktoré bežia v reálnom čase nastáva problém z oneskorenia alebo nepresnosti výpočtov. Ak je hardvér nadhodnotený je výkon postačujúci, ale sú zbytočné vynaložené finančné prostriedky. Pri prenosných aplikáciách musíme počítať aj s menšou výdržou na batérie a potrebou väčšieho chladenia. Meranie potrebného výkonu je možné zmerať na aplikáciách alebo algoritmoch, ktoré bežia na báze ktoréhokoľvek OS.

Testovanie možno rozdeliť do štyroch časti: príprava prostredia, testovanie, filtrácia hodnôt a analýza výsledkov. Testovanie bude prebiehať tak, že sa na vzorovom počítači zmeria spomalenie systému bez bežiaceho algoritmu a následne sa zmeria záťaž systému s bežiacim algoritmom. V jednoduchosti môžeme povedať, že sa tieto hodnoty odčítajú a vypočíta sa spotreba hardwarových prostriedkov. Pre správne meranie je dôležité dodržať presne postupnosti a minimalizovať ovplyvnenie merania externými vplyvmi. Neexistujú žiadne štandardy alebo postupy ako správne merať výkon algoritmov tak, aby nedochádzalo ku nepresnostiam spôsobeným meraním. V tejto práci sú navrhnuté správne meracie postupy a riešenia, ktoré sa využívajú na meranie vplyvov algoritmov na systém.

Pre rozpoznávanie a syntézu reči sa používajú výkonovo náročné algoritmy. Pre požiadavky multi-modálneho rozhrania musia tieto algoritmy pracovať v reálnom čase. Sú používané na systémoch ktoré majú obmedzené výpočtové prostriedky a z tohto dôvodu, je veľmi dôležité poznať ich systémové zaťaženie. Dizertačná práca je zameraná na zhrnutie metodológie merania záťaže systémových zdrojov v aplikácii multi-modálneho rozhrania a implementáciu algoritmov, ktoré umožnia zanalyzovať namerané výsledky. Cieľom tejto prace je navrhnúť a zlepšiť postupy merania systémovej záťaže. Samozrejmosťou je overenie navrhnutých zlepšení v reálnom prostredí.

2 CIELE

- **Návrh metodiky na meranie záťaže aplikáciami pre multimodálny interface**

  Cieľom je navrhnúť metodiku, ktorá bude použitá pri meraní záťaže aplikáciami pre multi-modálny interface, modifikovať súčasné prípadne navrhnúť nové postupy merania systémovej záťaže, zanalyzovať vplyv jednotlivých parametrov zaťaženia na rôzne druhy používateľov. V tejto časti je dôležité zamerať sa na potenciálne problémy,

ktoré môžu vzniknúť pri meraní a poskytnúť riešenia vo forme odporúčaní. Doplnkovou úlohou je navrhnúť aplikáciu, ktorá dokáže zmerať systémovú záťaž. Táto aplikácia by mala byť schopná merať systémové zaťaženie alebo výkon aplikácie staticky ako aj dynamicky v reálnych podmienkach.

- **Návrh a overenie metodológie na elimináciu rušení pri meraní**

  Meranie systémovej záťaže môže byť náchylné na chyby v meraní ako každé iné meranie. Tieto chyby môžu vniesť do výsledkov nepresnosť. Cieľom je navrhnúť postupy resp. metodológiu, ktorá by eliminovala prípadne minimalizovala tento jav. V tomto bode by mal byť presne zdokumentovaný postup resp. metodológia, ktorá čitateľa navedie ako správne a presne merať systémovú záťaž. Ďalším krokom je overenie tejto metodológie s reálnymi aplikáciami na rozpoznávanie reči. Prípadne je možné zmerať aj viacero verzií jednej aplikácie, resp. ukázať či majú rôzne parametre aplikácie na rozpoznávanie reči vplyv na systémovú záťaž.

- **Návrh a implementácia vyhodnocovacej metódy, ktorá vypočíta a zobrazí zaťaženie daným algoritmom**

  Je potrebné navrhnúť spôsob, akým sa budú analyzovať, vyhodnocovať a prezentovať dáta, ktoré sa získali meraním systémovej záťaže aplikácií alebo algoritmov. Pri vyhodnocovaní dát bude dôležité nájsť spôsob alebo algoritmus akým predísť prípadnej chybe merania, ktorá sa môže preniesť a nedokázalo sa jej zabrániť v predchádzajúcom bode. Bude potrebné využiť štatistické metódy a navrhnúť algoritmus, ktorý automaticky očistí namerané dáta a v prípade nedostatku meraní na to užívateľa upozorní. Ak už sú k dispozícii merania s minimálnou odchýlkou, je nutné navrhnúť spôsob akým sa zobrazia namerané hodnoty s prípadnou odchýlkou v prehľadnej grafickej forme.

Výstupom tejto práce by mala byť prehľadná metodológia a nástroje na meranie systémovej záťaže algoritmami používanými v multi-modálnych rozhraniach.

3 METODIKA NA MERANIE ZÁŤAŽE APLIKÁCIAMI PRE MULTIMODÁLNY INTERFACE

Výkon aplikácie a záťaž počítača je viacdimenzionálny problém. Najlepšou možnosťou ako sa správne pozerať na výkon a záťaž je pochopiť scenár, ktorý testujeme a akým spôsobom sa hardvér a operačný systém integrujú s testovanou aplikáciou. V tejto časti je opísaných päť dôležitých výkonovo-záťažových parametrov v kontexte záťaže testovanou aplikáciou.

Sústredením sa iba na jeden parameter výkonu aplikácie alebo záťaže systému, vytvoríme veľmi zjednodušený pohľad. Zameraním sa na jeden parameter, by programátori optimalizovali aplikáciu alebo algoritmus pre parameter v teste, na úkor ostatných rovnako dôležitých parametrov (AMTSO, 2010). Jediným spôsobom ako zamedziť takémuto správaniu je merať

všetky aspekty zaťaženia a urobiť kompromisy vedome, skôr než implicitne. Pre správne meranie systému musíme postupovať podľa metodológie a merať správne ukazovatele.

Cieľom je merať čo najväčšie množstvo rozdielnych parametrov. Každý jeden by mal poskytovať časť obrazu záťaže testovanou aplikáciou.

Existuje široké spektrum aspektov spotreby systémových zdrojov, ktoré môžu byť zaujímavé pre reálne testovanie záťaže. Použitie všetkých testov na každý algoritmus alebo aplikáciu je nemožné. Je potrebné vybrať správnu testovaciu sadu pre danú situáciu. Výsledky by čo najlepšie mali pokryť používateľské potreby.

Testy pre špecifické algoritmy

Niektoré testy sú špecifické pre algoritmy alebo aplikácie, pripadne pre dané prostredie, v ktorom prebieha meranie. Viacero aspektov algoritmov, ich kompozícia a chovanie môže byť merané a porovnávané, ale môžu mať obmedzenú výpovednú hodnotu. Ak je nutné danú aplikáciu nainštalovať, zmerať inštalační čas alebo veľkosť inštalácie nemusí mať zmysel pri stacionárnych systémoch, ale pri mobilných operačných systémoch tu už význam má. Existuje viacero testov ako napríklad naštartovanie grafického používateľského rozhrania alebo reakčná doba rozhrania ktoré sú zaujímavé pre úzku skupinu používateľov alebo vývojárov.

Všeobecné použiteľné testy

Pre použitie reálnych testov, ktoré sa dajú aplikovať na skoro všetky algoritmy je ich nutné prispôsobiť použitiu cieľovej skupiny a merať len tie ukazovatele, ktoré odrážajú používateľské vzory a záujmy.

Pri porovnávaní rôznych druhov merania systémového zaťaženia systému je nutné brať do úvahy rôznorodé aspekty aplikácií, ktoré meriame. Zo subjektívneho hľadiska používateľa je dôležité, aby nepociťoval vplyv aplikácie na jeho prácu, čo však nemusí mať súvis s veľkosťou spotrebovanej pamäte alebo procesorového času. Z technického hľadiska je dôležité mať malú spotrebu pamäte, diskového priestoru alebo CPU cyklov, avšak keď aplikácie vynikajúco zvládajú syntetické testy, nemusia reflektovať reálnu záťaž na systém, ktorú pociťuje užívateľ.

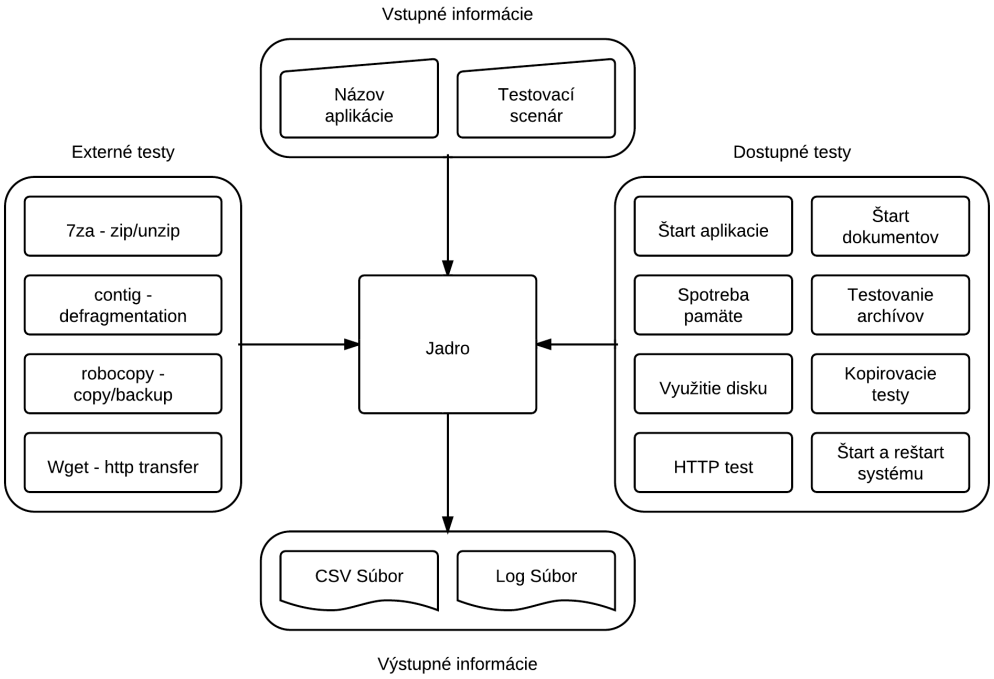
Dôležité je merať parametre, ktoré ovplyvňujú ako užívateľ vníma rýchlosť odozvy systému. Parametre, ktoré nemajú žiaden vplyv na výkon alebo spotrebu systémových zdrojov nie je potrebné merať.

Zmerať zaťaženie, ktoré sa čo najviac blíži k tomu čo pociťuje užívateľ nie je možné jedným testom. Je preto dôležité vytvoriť súbor meraní, ktoré sú vážené podľa danej užívateľskej skupiny. Z Tabuľka 1 - Dôležitosť testov pre užívateľské skupiny je možné rozpoznať, že každý druh testu je dôležitý pre inú skupinu respektíve oblasť užívateľov. Uvádzané váhy v tabuľke, boli štatisticky odhadnuté pre dané skupiny používateľov. Vychádzalo sa pri tom zo zastúpenia týchto parametrov v reálnom používaní počítača.

V mojom článku, ktorý bol spomínaný v predchádzajúcej kapitole som navrhol podobne rozdelenie užívateľov do váh (Vrabec & Harley, 2010). Potvrdenie mojich záverov bola aj citácia v článku na zahraničnej konferencii, kde píše že je to najlepší postup aký možno zvoliť (Košinár, Malcho, Marko, & Harley, 2010).

	Priemerný používateľ	Surfista	Hráč	Grafik	Podnikový zákazník	Mobilný používateľ
Čas spustenie aplikácii	100%	80%	100%	100%	100%	100%
Čas otvorenia lokálnych dokumentov	100%	50%	100%	100%	100%	100%
Kopírovanie, presun a mazanie súborov	50%	50%	50%	50%	50%	50%
Archivácia	20%	10%	20%	20%	20%	20%
Inštalácia	20%	20%	20%	20%	20%	20%
Sieťové Testy	100%	100%	100%	100%	100%	100%
Kódovanie a enkrypcia	20%	50%	60%	60%	20%	20%
Veľkosť spotrebovanej pamäte	40%	40%	70%	40%	80%	40%
Štart, reštart a vypínanie počítača	30%	30%	30%	30%	10%	30%
Procesy	0%	0%	0%	0%	50%	0%
Simulácia systému	10%	10%	10%	10%	10%	10%
Simulovaná interakcia s programami	100%	60%	100%	100%	40%	100%
Grafické testy	20%	20%	100%	20%	20%	20%
Testy procesora	10%	10%	10%	10%	100%	10%
Hardvérové testy	10%	10%	10%	10%	100%	10%

Tabuľka 1 - Dôležitosť testov pre užívateľské skupiny



Obrázok 1 - Systémová schéma meracieho program TCExecutor.

Na meranie parametrov systémovej záťaže sme navrhli a zrealizovali merací program TCExecutor, ktorý bol naprogramovaný na meranie záťaže antivírusových programov a v tejto práci má slúžiť ako demonštrácia správnosti merania systémovej záťaže aplikáciami. Cieľom dizertačnej prace nie je naprogramovať meraciu aplikáciu, ale len prezentovať vplyv aplikácii na systémovú záťaž. Systémová schéma meracieho programu je na obrázku 1.

4 METODOLÓGIE NA ELIMINÁCIU RUŠENÍ PRI MERANÍ

Všetky testy by mali byt konštruované s cieľom dosiahnuť najvyššiu možnú presnosť. Najlepšie ako to možno dosiahnuť je eliminovať možné rušenia. Ďalším krokom sú opakované merania, ktoré nám umožnia vypočítať priemernú hodnotu s minimálnou odchýlkou.

Pri meraní systémového zaťaženia môžu vzniknúť odchýlky pri meraní. Je dôležité, aby každé meranie bolo zopakované viackrát. Pri jednom meraní môžu vzniknúť nepresnosti, ktoré zapríčiňuje operačný systém počítača z dôvodu jeho dynamického chovania. Zopakovaním merania počas totožných podmienok sa minimalizujú odchýlky a následným priemerom sa zaistí presnejší výsledok.

Ďalšou možnosťou ako zvýšiť presnosť je eliminácia extrémnych hodnôt, resp. odstránenie najmenších a najväčších hodnôt. Týmto spôsobom ostane priemerná hodnota s minimálnou odchýlkou. V ďalšej časti rozdelíme testovanie do štyroch časti: príprava prostredia, testovanie, filtrácia hodnôt a analýza výsledkov. Na obrázku 2, sa nachádza diagram testovania.

Príprava prostredia

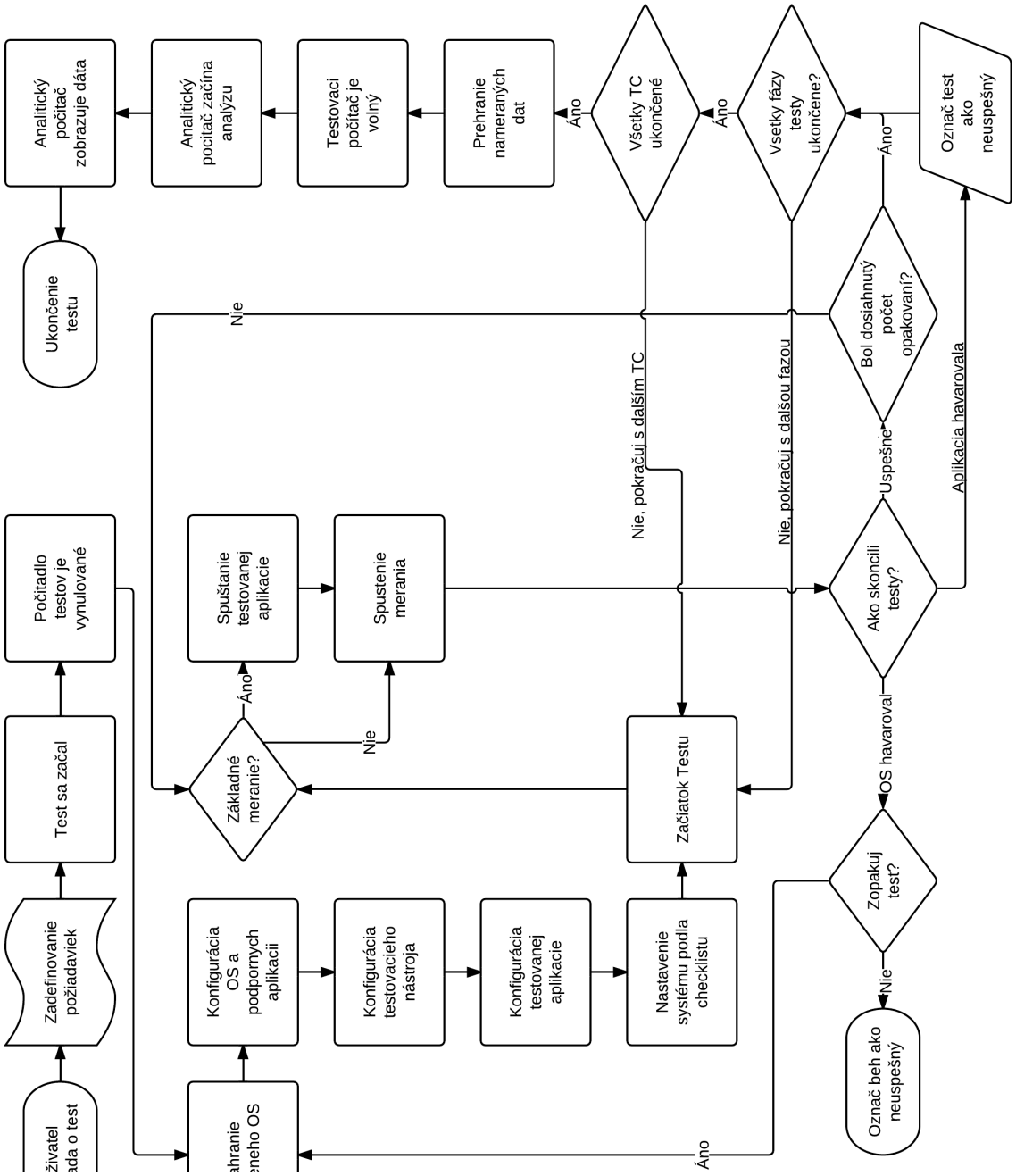
Proces sa začína keď tester nadefinuje požiadavky testu. Podľa toho aký operačný systém si tester vyberie sa podľa postupu nižšie nahrá príslušný obraz OS do testovacieho počítača. Každý nový test sa začína prepísaním obrazu OS na počítači z dôvodu, aby sa test začal v známom stave. Ak sú potrebné ďalšie programy, alebo ovládače sú doinštalované manuálne. Následne sa nakonfiguruje testovací scenár podľa požiadaviek. Posledný krok pred začiatkom testu je uistenie sa testera, že testovaná aplikácia beží podľa zadaných parametrov.

Testovanie

Testovanie je uskutočnené vo viacerých fázach, aby sa minimalizoval počet potrebných merní a následne je analýza výsledkov je prenosená na iný počítač. Ak by sme tento postup nevykonali, zvyšujeme možnosť pozorovacieho efektu. Je to jav, keď tester analyzuje výsledky testu, zatiaľ čo test beží na tom istom zariadení (Jansen, Taksa, & Spink, 2008). Napríklad: ak tester prenáša dáta počas testu a súčasné beží test, môže meranie spomaliť, čím zanesie do merania chybu. Prvý prechod testu je zahrievacie kolo. Zahriatie znamená, že aplikácia, ktorá je využívaná na testovanie ešte nie je nahraná v pamäti a počas tohto procesu sa zavedie do pamäte, pri opätovnom spustení tejto aplikácie je štart rýchlejší. Toto zrýchlenie funguje len do najbližšieho reštartu.

Chyby a filtrácia hodnôt

Ak počas testu, testovaná aplikácia havaruje, tento stav sa zaznamená do chybového záznamu a test sa začne odznova. Avšak ak počas testu spadne operačný systém počítača alebo testovacia aplikácia, počítač bude prehraný pôvodným obrazom, lebo stav operačného systému nemožno zaručiť. Ak sa takýto stav opakuje je testovací cyklus zastavený a vyhodnotí sa, čo spôsobilo nestabilitu. Keď sa všetky testovacie súbory dokončia, tester ukončí test a prenesie výsledky na analýzu. Testovací počítač je pripravený pre nový test.



Obrázok 2 - Diagram testovania

Analýza výsledkov

Meranie sa skladá z rôznych testov, ktorý každý je spúšťaný minimálne desať krát a pomocou štatistických nástrojov sú netypické alebo extrémne hodnoty vylúčené alebo vyznačené na neskoršiu analýzu. Používateľ môže merať testovanú aplikáciu v rôznych situáciách a na širokom spektre hardvéru a preto je jednom z našich cieľov zaistiť hladkú a predvídateľnú každú reakciu.

5 VYHODNOCOVACIA METÓDA, KTORÁ VYPOČÍTA A ZOBRAZÍ ZAŤAŽENIE DANÝM ALGORITMOM

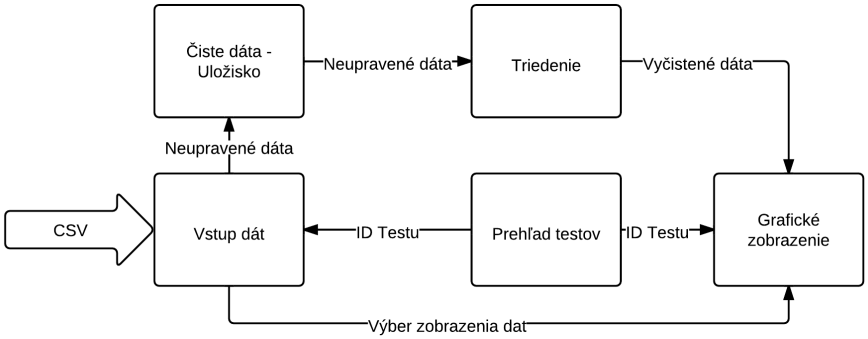
Vyhodnocovanie výsledných dát merania je rozdelené na dva kroky - zbieranie a analýza. Prehratím analýzy na dedikovaný počítač môžu začať testovacie počítače na ďalšom teste a oveľa výkonnejšie počítače serverovej triedy môžu byť použité na spracovanie analýzy omnoho rýchlejšie. Čím skôr je dokončená analýza, tým efektívnejšie sa môžu identifikovať problémové oblasti spomalenia a programátor môže zmeniť aplikáciu alebo algoritmus.

Pre analýzu výsledkov bol vytvorený nástroj, ktorý načíta dáta, vykoná analýzu výsledkov a zobrazí ich v grafickej forme. Tento nástroj pracuje na počítači s výkonným procesorom, ktorý je určený výhradné na spracovávanie výsledkov merania. Výstupom každého merania je CSV súbor, ktorý sa prenesie do analytického počítača a v ňom sa spracuje. Podľa počtu testov môže spracovanie trvať desiatky minúť. Následne sa hodnoty jednotlivých testov premietnu do grafov.

V nasledujúcej časti je popísaný automatický nástroj na spracovanie výsledkov merania. Nástroj je vytvorený v tabuľkovom procesore Microsoft Excel 2010, programovacím jazykom Visual Basic. Táto platforma bola zvolená z dôvodu efektívnej práce s dátami a kompatibility. Tabuľkový procesor je rozdelený na tieto oblasti:

- Vstup dát
- Čisté dáta
- Triedenie
- Grafické zobrazenie
- Prehľad testov

Postup prenosu dát je zobrazený na obrázok 3.

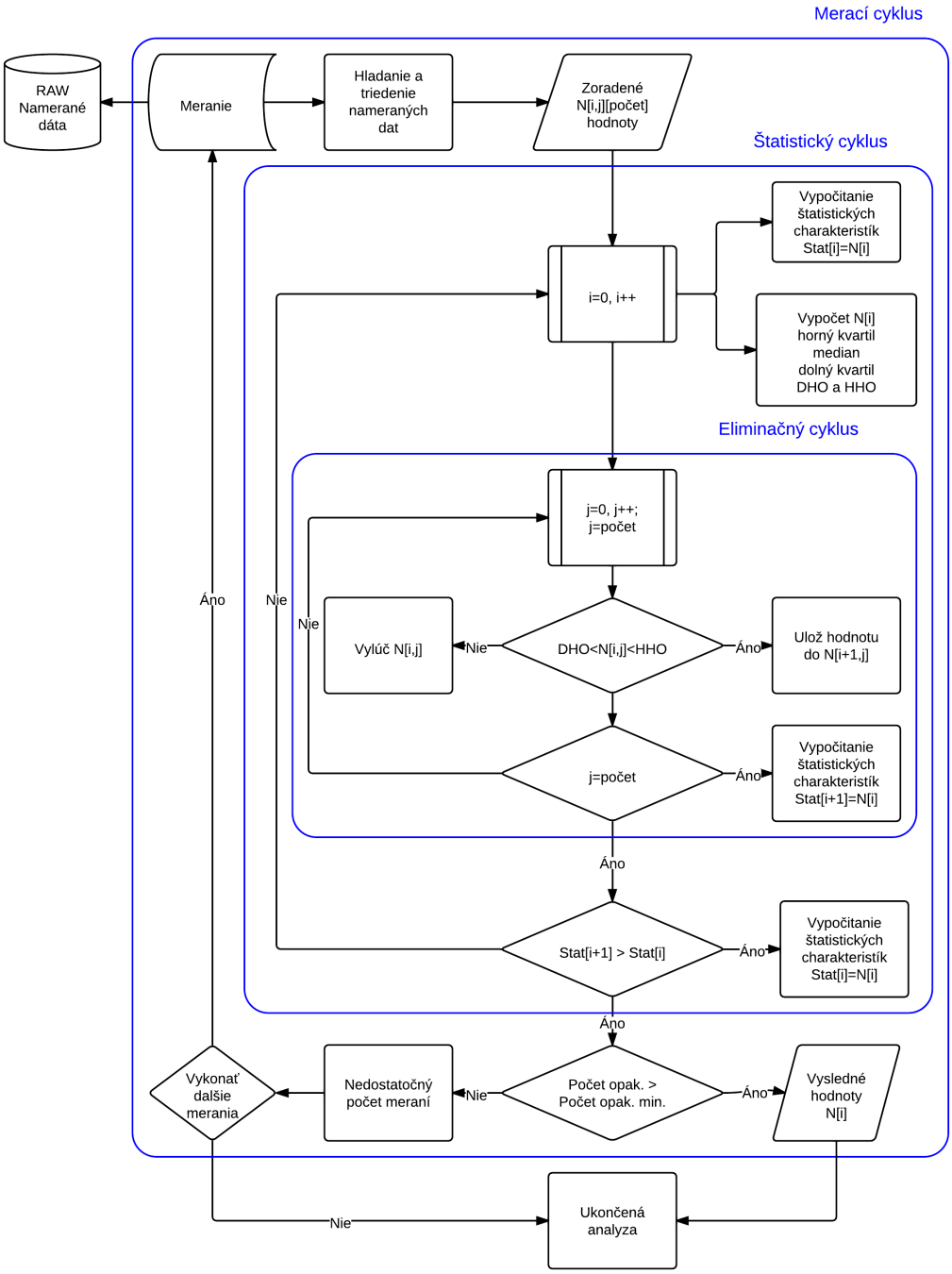


Obrázok 3 - Postup analýzy a zobrazenia dát

Ďalej je popísaný algoritmus, ktorý automaticky analyzuje namerané hodnoty a odstráni prípadne chyby merania. Schéma algoritmu je na obrázku 3. Algoritmus je rozdelený na tri cykly, prvý je merací cyklus, v ňom je vnorený druhý štatisticky cyklus a v ňom prebieha samotný tretí eliminačný cyklus.

Merací cyklus

Merací cyklus začína vstupom nameraných dát systémového zaťaženia a skončí keď je ukončená analýza. Jeho principiálna schéma sa nachádza na obrázku 4. Dáta sú v čistej, nespracovanej forme, pred ďalším spracovaním, aby nedošlo k náhodnej zmene dát, sú úložné v databáze RAW výsledkov. Keďže sú dáta uložené v poradí ako boli namerané, pred ďalším spracovaním je nutné ich roztriediť podľa meraní. Tento proces automaticky zatriedi jednotlivé hodnoty k príslušným meraniam. Výsledkom procesu je matica opakovaní jedného parametra merania N[*i,j*], premenná *i* je iterácia vylučovacej metódy a premenná *j* predstavuje identifikátor opakovania jedného merania. Číslo počet je celkový počet opakovaní jedného typu meraného



Obrázok 4 - Schéma algoritmu

parametra. Následne začína štatistický cyklus, ktorého výstupom je rozhodnutie či je počet opakovaní merania dostatočný.

Štatistický cyklus

Štatistický cyklus na obrázku 4 slúži na výpočet štatistických charakteristík z súboru meraní N[i] a vstupných hodnôt pre eliminačný cyklus. Po eliminačnom cykle je rozhodovacia funkcia, či je chyba merania po eliminácii chýb meraní menšia ako pred elimináciou alebo rovnaká.

- Chyba je menšia, v tomto prípade sa pokračuje s ďalšou iteráciou eliminačného cyklu
- Chyba je rovnaká, vypočíta štatistické charakteristiky z N[i] a ukončí štatistický cyklus

Eliminačný cyklus

Tento cyklus má za úlohu vylúčiť chybové merania. Cyklus začína porovnávať každé meranie zo súboru s pred vypočítanými kvartilmi. Ak je hodnota merania medzi dolnou a hornou hranicou odchýlky, uloží túto hodnotu do N[i,j+1]. Ak nie je v tomto intervale, tak túto hodnotu vylúči. Keď algoritmus spracuje celý súbor meraní s dĺžkou *počet*, vypočíta štatistické charakteristiky a opúšťa cyklus. Ak algoritmus spracuje súbor meraní s dĺžkou menšou ako *počet* pokračuje s ďalšou hodnotou.

6 OVERENIE METODOLÓGIE NA ELIMINÁCIU RUŠENÍ PRI MERANÍ APLIKÁCIE

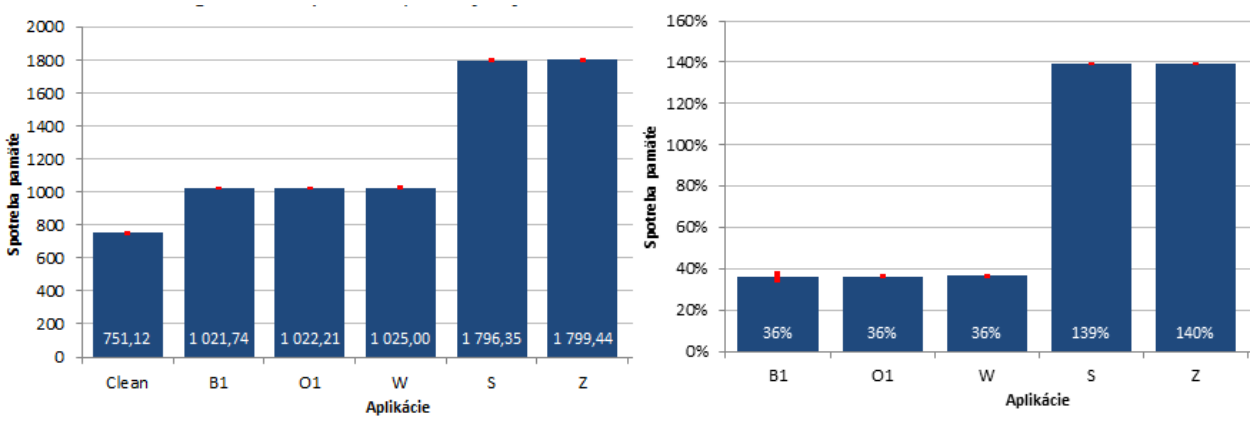
Na otestovanie merania bola vybraná aplikácia, ktorá pracuje s rečou. Cieľom merania je zistiť vplyv aplikácií, ktoré rozpoznávajú reč na systémové prostriedky počítača. Aplikácia rozpoznáva slovenskú reč na počítači. Cieľom merania je zistiť, aký vplyv má prebiehajúce rozpoznávanie na spotrebu systémových zdrojov. Aplikácia spustí rozpoznávač slovenskej reči, ktorý analyzuje hlas na mikrofóne a rozpoznáva rôzne slová (Kőrösi, Vojtko, & Rozinaj, 2010). Na porovnanie veľkosti zaťaženia rôznymi súbormi znakov boli vybrané rôzne korpusy.

Spotreba systémovej pamäte

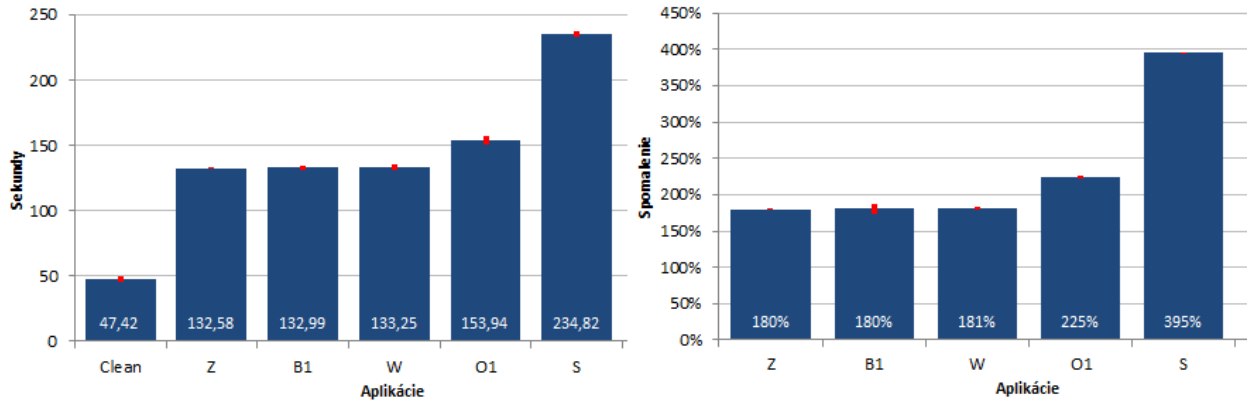
Na obrázku 5 sa nachádza spotreba systémovej pamäte počítača aplikáciou s rôznymi korpusmi. Podľa nameraných výsledkov je možné konštatovať, že najviac zaťažuje systémovú pamäť rozpoznávač, ktorý rozpoznáva slovenskú reč podľa vstupného súboru s viacerými slovami.

Štart počítača

V prípade, že sa aplikácia na rozpoznávanie slovenskej reči spúšťa pri štarte počítača, z dôvodu aby kontinuálne rozpoznávala reč, v prípade korpusu S sa spustenie počítača oneskorí až o 400%, obrázok 6.



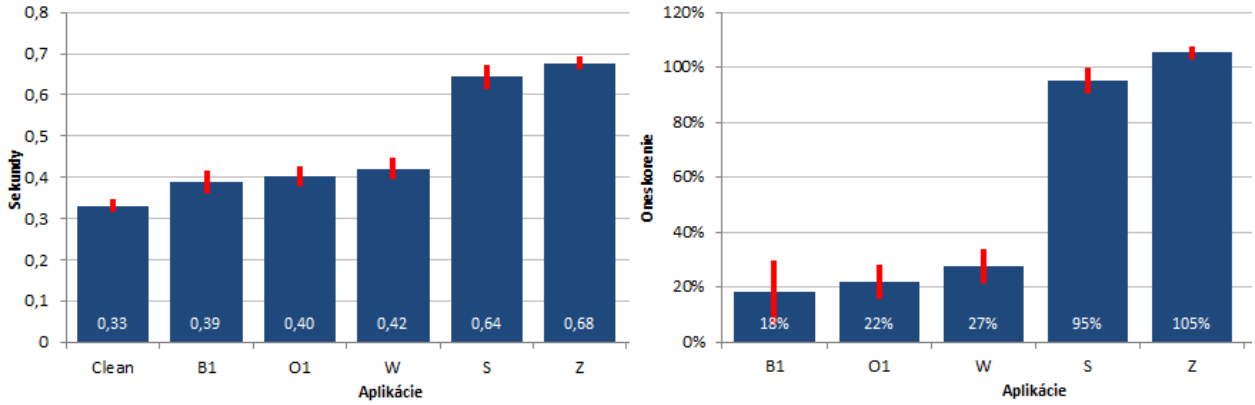
Obrázok 5 - Spotreba systémovej pamäte aplikáciou, uvádzaná v megabajtoch a percentách



Obrázok 6 - Oneskorenie štartu počítača, v absolútnych hodnotách ako aj v percentách

Otváranie dokumentov

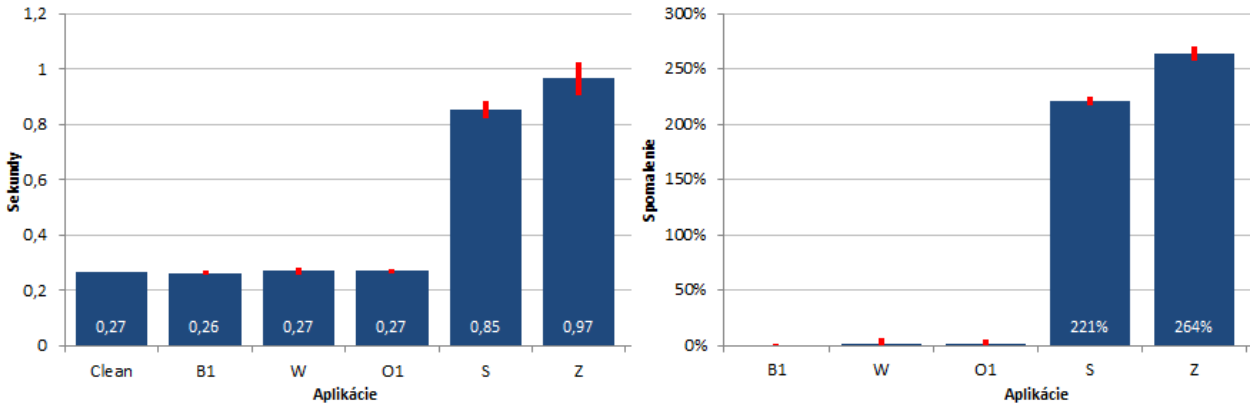
Ak pracuje rozpoznávač slovenského jazyka na pozadí ma vplyv aj na rýchlosť otvárania dokumentov. Zvlášť sa spomalenie prejavuje s korpusom S a Z, obrázok 7.



Obrázok 7 - Oneskorenie prvého otvorenia dokumentu v absolútnych číslach a percentách

Spúšťanie internetového prehliadača

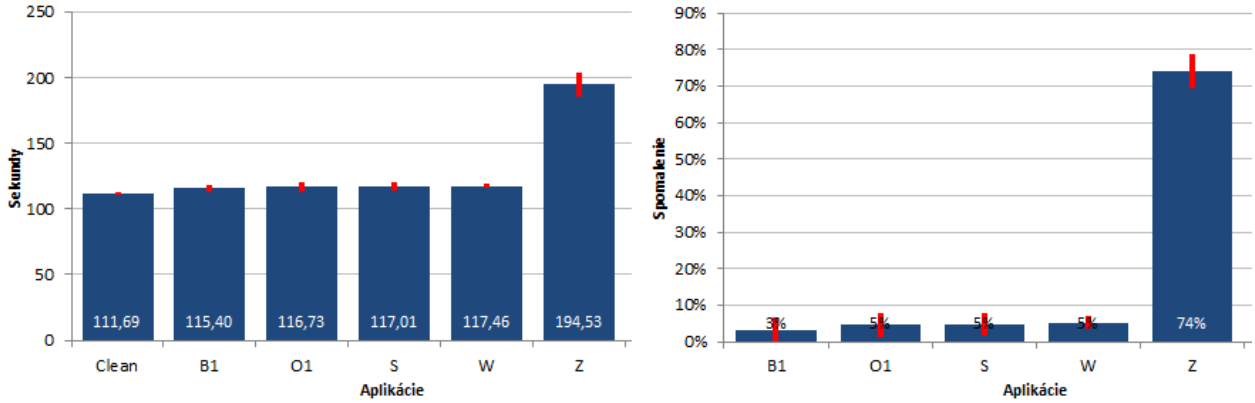
Zaujímavým zistením je, že rozpoznávač slovenskej reči s korpusom B1, W a O1 nemajú vplyv na rýchlosť spúšťania internetového prehliadača, naopak korpus S a Z rýchlosť štart internetového prehliadača spomalia až o 221% respektíve 264%, obrázok 8.



Obrázok 8 - Oneskorenie prvého štartu prehliadača internetu

Spomalenie kopírovania multimediálnych súborov

Spustená aplikácia na rozpoznávanie slovenskej reči s korpusom Z má vplyv aj na rýchlosť kopírovania multimediálnych súborov, spomalenie je až 74%, obrázok 9.



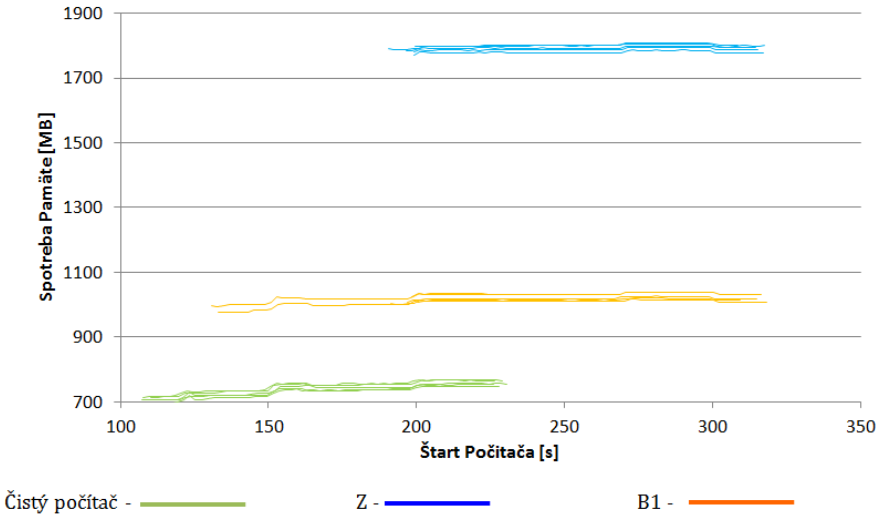
Obrázok 9 - Spomalenie kopírovania multimediálnych súborov

Dynamická spotreba pamäte a CPU pri štarte počítača

V predchádzajúcej časti boli prezentované statické merania, ktoré predstavujú priemerné systémové zaťaženie. V prípade, ak je nutné presne analyzovať vzťah spotreby pamäte alebo využitia CPU na udalosti v systéme, napríklad pri štarte počítača meranej aplikácie alebo pri určitých činnostiach aplikácie, je veľmi vhodné merať dynamicky.

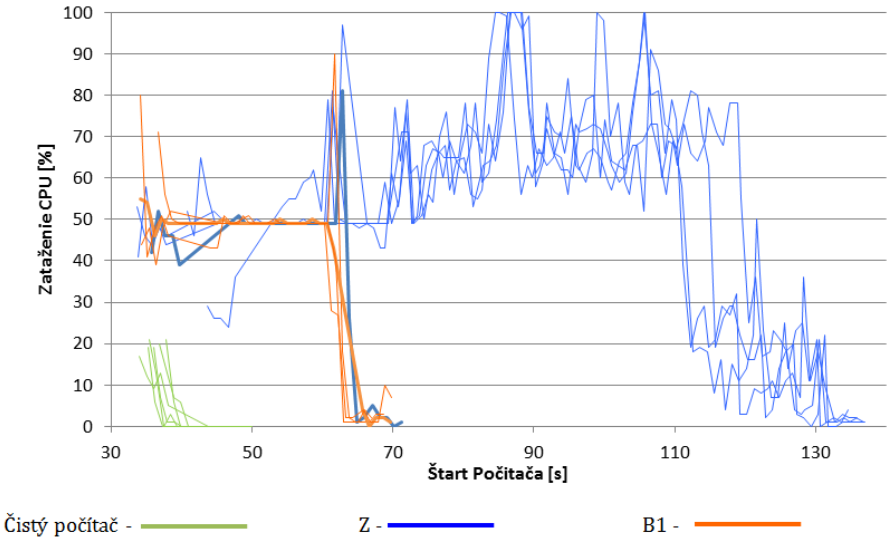
Záťaž meranej aplikácie má vplyv na niekoľko parametrov systému. Na grafe (obrázok 10) je zobrazená spotreba pamäte aplikáciou s rôznymi korpusmi, je to závislosť času od spustenia

počítača ku spotrebe pamäte operačného systému OS. Na grafe je možné rozpoznať približný čas, kedy naštartoval čistý systém a systém s meranou aplikáciou a priemernú spotrebu pamäte. Priemerné hodnoty spotreby pamäte sú počas štartu stabilné a nedochádza k výkyvom.



Obrázok 10 – Dynamická spotreba pamäte

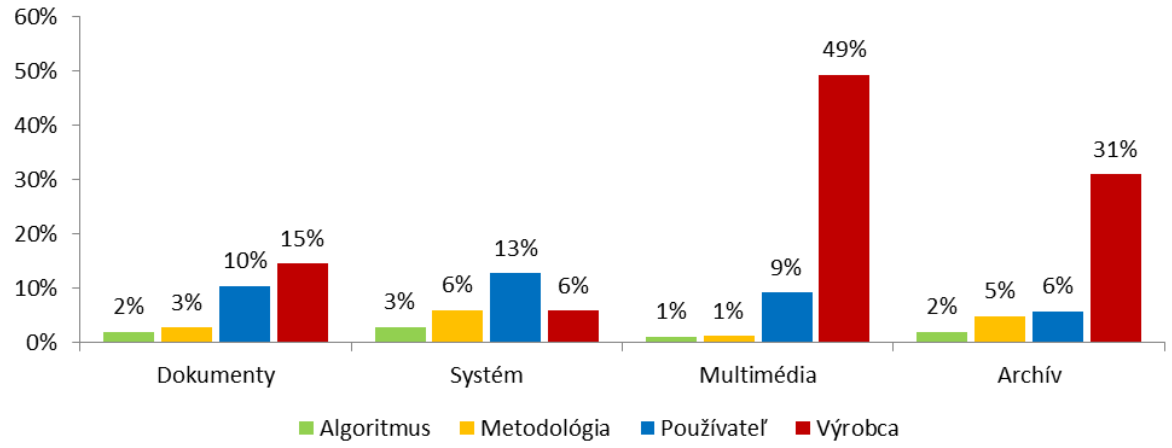
Na ďalšom grafe (obrázok 11) je zobrazené aktuálne zaťaženie CPU aplikáciou s rôznymi korpusmi ako závislosť času od spustenia operačného systému ku zaťaženiu CPU vyjadrené v percentách. Na grafe je možné rozpoznať približný čas, kedy naštartoval čistý systém, ktorý bol celkovo stabilný, avšak systém s meranou aplikáciou vykazuje známky nepravidelnej záťaže CPU.



Obrázok 11 – Dynamická záťaž CPU

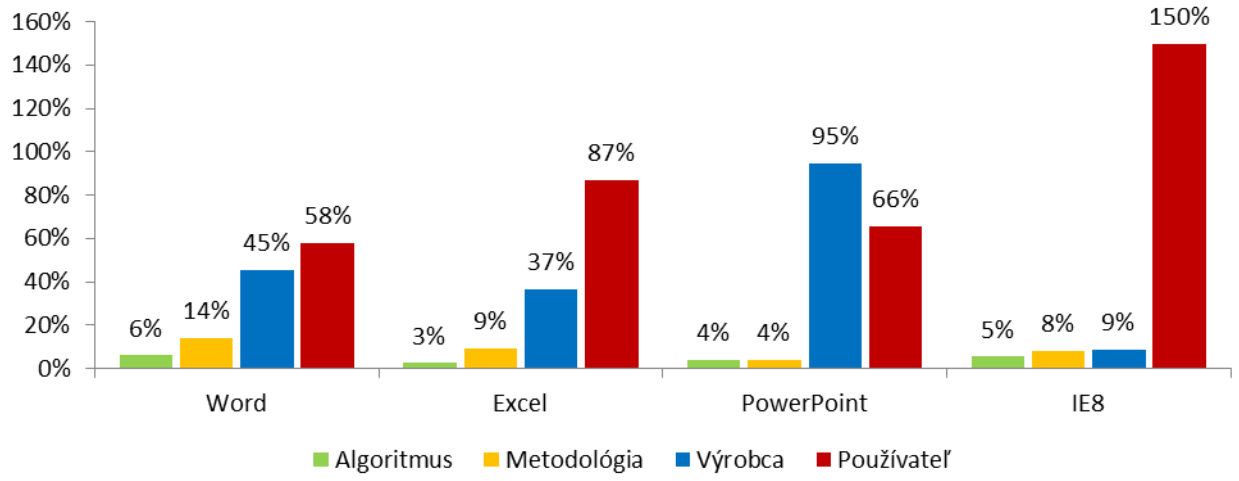
Overenie vplyvu metodiky na chybovosť merania

V tejto časti je overená navrhnutá metodológia, z pohľadu zmenšenia chybovosti merania. Na grafoch (Obrázok 12 a 13) je možné vidieť štyri hodnoty, ktoré boli namerané na tom istom hardvéry - meranie používaného systému, systému v nastaveniach od výrobcu, systém nastavený podľa metodológie a chybovosť merania po úprave algoritmom.



Obrázok 12 - Pomer štandardnej odchýlky pri meraní rýchlosti kopírovania

Používateľský systém je prostredie, ktoré reprezentuje priemerný systém, ktorý je používaný užívateľom. Na tomto systéme sú nainštalované a bežia rôzne programy, ktoré náhodne zaťažujú systém. Systém v nastaveniach od výrobcu je prostredie, ktoré reprezentuje systém tak ako je dodané od výrobcu hardvéru. Na počítači sú predinštalované aplikácie, ktoré náhodne zaťažujú systém a samotný operačný systém tiež vykazuje známky náhodného zaťaženia. Systém, ktorý bol nastavený podľa popísanej metodológie je ten istý operačný systém s bežnými aplikáciami potrebnými na vykonanie testu. Ako možno vidieť, vplyv úprav na systém má výrazný vplyv na zníženie chybovosti merania. Následne spracovaním výsledkov algoritmom na odstránenie nepresnosti je možné dosiahnuť akceptovateľnú chybovosť.



Obrázok 13 - Pomer štandardnej odchýlky pri opakovanom spúšťaní aplikácie

7 DOSIAHNUTÉ VÝSLEDKY

- Bola navrhnutá a overená metodika na meranie záťaže aplikáciami pre multimodálny interface a súčasne modifikované postupy merania systémovej záťaže. Boli navrhnuté odporúčania pre testovanie, ktoré predídu problémom, ktoré vznikajú pri meraní.
- Bol navrhnutý systém na klasifikáciu používateľov a nájdenie optimálnych váh k jednotlivým parametrom zaťaženia.
- Bola navrhnutá aplikácia, vrátane užívateľského prostredia, ktorá dokáže zmerať systémovú záťaž. Táto aplikácia meria systémové zaťaženie a výkon aplikácie staticky ako aj dynamicky v reálnych podmienkach.
- Bola navrhnutá metodológia na elimináciu rušení pri meraní. Táto metodológia, užívateľa navedie ako správne a presne merať systémovú záťaž. Metodológia bola overená pri meraní reálnych aplikácií na rozpoznávanie reči. Zvolená aplikácia bola zmeraná s viacerými databázami a bolo dokázané, že rôzne parametre aplikácie na rozpoznávanie reči majú vplyv na systémovú záťaž.
- Bol navrhnutý a implementovaný algoritmus, ktorý za pomoci štatistických metód automaticky identifikuje a eliminuje vzniknuté odchýlky pri meraní a v prípade, že presiahnu stanovenú hraničnú hodnotu odporučí osobe vykonávajúcej testy, aby vylúčila a zopakovala meranie.
- Bol navrhnutý a implementovaný vyhodnocovací algoritmus, ktorý spracuje namerané dáta vo forme systémového zaťaženia meranej aplikácie alebo algoritmu. Výsledné hodnoty sú prezentované v prehľadnej grafickej forme na finančných grafoch, ktoré popri výsledných hodnotách zobrazujú aj odchýlky v meraní.

8 POUŽITÁ LITERATÚRA

Clementi, A. (1. január 2008). *Performance Tests*. Cit. 2012. október 01. Dostupné na Internete: AV-Comparatives: : http://www.av-comparatives.org/en/comparativesreviews/performance-tests

Clonezilla. (30. január 2012). *About*. Cit. 01. febuár 2012. Dostupné na Internete: Clonezilla: http://clonezilla.org/

Croarkin, C., & Tobias, P. (2004). *e-Handbook of Statistical Methods*. Cit. 01. 01 2012. Dostupné na Internete: NIST/SEMATECH: http://www.itl.nist.gov/div898/handbook/

AMTSO. (01. január 2012). *About us*. Cit. 01. február 2012. Dostupné na Internete: Anti-Malware Testing Standards Organization: http://www.amtso.org/about-us.html

AMTSO. (02. jún 2010). AMTSO Whole Product Testing Guidelines. Helsinki, Finland.

Anti-Malware Testing Standards Organization. (03. 11 2008). *The Fundamental Principles of Testing.* Cit. 01. 01 2012. Dostupné na Internete: AMTSO - Documents: http://www.amtso.org/documents.html

AV-Test. (01. jún 2011). *Product Review and Certification Report*. Cit. 01. január 2012. Dostupné na Internete: AV-Test: http://www.av-test.org/no_cache/en/tests/test-reports/test-reports/?tx_avtestreports_pi1%5Breport_no%5D=112256

Šulhin, T. (01. september 2010). Test Executor Guide. Bratislava, Slovakia.

Baquiran, M., & Wren, D. (01. november 2011). *2012 Consumer Security Products Performance.* Cit. 01. január 2012. Dostupné na Internete: Passmark: http://www.passmark.com/ftp/antivirus_12-performance-testing-ed2.pdf

Devore, J. L. (2000). *Probability and statistics for engineering and the sciences. .* Duxbury.

Future Mark. (01. január 2011). *PC Mark 7*. Cit. 01. január 2012. Dostupné na Internete: Future Mark: http://www.pcmark.com/benchmarks/pcmark7/

FutureMark. (01. január 2012). *Company Info*. Cit. 01. január 2012. Dostupné na Internete: FutureMark: http://www.futuremark.com/companyinfo/aboutus/

FutureMark. (01. január 2012). *PowerMark*. Cit. 01. január 2012. Dostupné na Internete: FutureMark: http://www.futuremark.com/benchmarks/powermark/

FutureMark. (01. január 2010). *Vantage*. Cit. 01. január 2012. Dostupné na Internete: PCMark: http://www.futuremark.com/benchmarks/pcmarkvantage/introduction/

Hawes, J. (25. máj 2010). AMTSO Performance testing guidelines. London, United Kingdom.

HBB-NEXT Consortium. (4. Október 2011). *HBB-NEXT - Next Generation Hybrid Media*. (H.-N. Consortium, Producent) Cit. 1. Január 2012. Dostupné na Internete: About Project: http://www.hbb-next.eu/

Jacobs, L. J., & Dinman, J. D. (dátum neznámy). *Systematic Analysis of Bicistronic Reporter Assay Data*. Cit. 01. január 2012. Dostupné na Internete: Department of Cell Biology & Molecular Genetics: http://dinmanlab.umd.edu/statistics/tutorial.html

James, J. (2012). *Statistics: Lecture Notes*. Cit. 01. január 2012. Dostupné na Internete: Richland Community College: http://people.richland.edu/james/lecture/m170/ch08-int.html

Jansen, B. J., Taksa, I., & Spink, A. (29. júl 2008). *Research and Methodological Foundations of Transaction Log Analysis.* Cit. 01. január 2012. Dostupné na Internete: College of Information sciences and technology: http://faculty.ist.psu.edu/jjansen/academic/jansen_theoretical_foundations.pdf

Kaspersky, E. (30. september 2011). *Benchmarking Without Weightings: Like a Burger Without a Bun.* Cit. 01. január 2012. Dostupné na Internete: Nota Bene: http://eugene.kaspersky.com/2011/09/30/benchmarking-without-weightings-like-a-burger-without-a-bun/

Kaspersky, E. (18. október 2011). *Nota Bene*. Cit. 01. január 2012. Dostupné na Internete: The Holy Grail of AV Testing, and Why It Will Never Be Found: http://eugene.kaspersky.com/2011/10/18/the-holy-grail-of-av-testing-and-why-it-will-never-be-found/

Kevin, C. (05. febuár 2006). *Hardware Testing and Benchmarking Methodology*. Cit. 01. január 2012. Dostupné na Internete: Donutey: http://donutey.com/hardwaretesting.php

Košinár, P., Malcho, J., Marko, R., & Harley, D. (2010). AV TESTING EXPOSED. *Virus Bulletin Conference* (s. 281-287). Vancouver: Virus Bulletin.

Kupper, R. R., & Hafner, K. B. (1989). How Appropriate Are Popular Sample Size Formulas? *American Statistician, 43* , 101-105.

Kőrösi, J., Vojtko, J., & Rozinaj, G. (2010). Statistic evaluation of various speech parameters for phonemes in slovak language. *52nd international symposium ELMAR-2010*, (s. 375-378). Zadar, Croatia.

Marx, A. (01. marec 2010). *Test Reports*. Cit. 01. január 2012. Dostupné na Internete: AV-Test.org: http://www.av-test.org/en/tests/test-reports/

Mendenhall, W., & Sincich, T. (2008). *Statistics for the engineering and computer sciences.* New York: Macmillan Publ. Co.

Microsoft. (16. február 2012). *Internet Explorer Performance Lab: reliably measuring browser performance*. Cit. 01. marec 2012. Dostupné na Internete: MSDN Blogs: http://blogs.msdn.com/b/b8/archive/2012/02/16/internet-explorer-performance-lab-reliably-measuring-browser-performance.aspx

Microsoft. (22. jún 2010). *Measuring Browser Performance with the Windows Performance Tools*. Cit. 01. január 2012. Dostupné na Internete: MSDN Blogs: http://blogs.msdn.com/b/ie/archive/2010/06/21/measuring-browser-performance-with-the-windows-performance-tools.aspx

NovaBench. (01. január 2012). *Main*. Cit. 01. január 2012. Dostupné na Internete: NovaBench: http://novabench.com/

Passmark. (01. január 2011). *About Passmark*. Cit. 01. január 2012. Dostupné na Internete: Passmark: http://www.passmark.com/about/index.htm

PassMark. (02. febuár 2012). *PerofromanceTest*. Cit. 01. marec 2012. Dostupné na Internete: PassMark: http://www.passmark.com/products/pt.htm

Primate Labs. (01. január 2010). *Geekbench*. Cit. 01. január 2012. Dostupné na Internete: Primate Labs: http://www.primatelabs.ca/geekbench/

SiSoftware. (01. máj 2011). *Main*. Cit. 01. január 2012. Dostupné na Internete: SiSoftware: http://www.sisoftware.co.uk/

Sojková, Z. (30. september 2008). *Studijné materialy - Deskriptívne charakteristiky.* Cit. 01. január 2012. Dostupné na Internete: Katedra štatistiky a operačného výskumu, Fakulta Ekonomiky a Manažmentu, Slovenska Poľnohospodarska Univerzita v Nitre: http://www.fem.uniag.sk/ACADEMIC/depart/ksov/predmety/statistika/subory/charakteristiky.pdf

Vrabec, J. (25. január 2010). *Generalist Anti-Malware Product Testing*. Cit. 01. január 2012. Dostupné na Internete: ESET Threat Blog: http://blog.eset.com/2010/01/25/generalist-anti-malware-product-testing

Vrabec, J., & Harley, D. (2010). Real Performance? *The 19th Annual EICAR Conference.* Paris: Eicar.

Warner, O. (01. január 2006). *What Really Slows Windows Down*. Cit. 01. január 2012. Dostupné na Internete: The PC Spy: http://thepcspy.com/read/what_really_slows_windows_down/

Zotac. (15. január 2012). *Zotac Product Webpage*. (Zotac, Producent) Cit. 01. febuár 2012. Dostupné na Internete: ZOTAC ZBOX ID36 Blu-ray 3D PLUS: http://www.zotacusa.com/zbox-id36-blu-ray-3d-plus.html

9 VLASTNE PUBLIKÁCIE VZŤAHUJÚCE SA K DIZERTÁCII

ABC Kapitoly vo vedeckých monografiách vydané v zahraničných vydavateľstvách

Rozinaj, G., Vrabec, J., Čepko, J., & Talafová, R. (2008). Terminals for the Smart Information Retrieval. In I. K. IBRAHIM, Handbook of Research on Mobile Multimedia : Vol.1.,2. (Zv. 2, s. 1 - 274). Hershey: Information Science Reference.

ADE - Vedecké práce v zahraničných nekarentovaných časopisoch

Talafova, R., Rozinaj, G., Čepko, J., & Vrabec, J. (2008). Multimedia SMS Reading in Mobile Phone. International Journal of Mathematics and Computers in Simulation Vol. 1, Iss. 1., 12 - 17.

ADF Vedecké práce v domácich nekarentovaných časopisoch

Vrabec, J., & Rozinaj, G. (2007). Multichannel Beamforming for Speech Recognition. EE časopis pre elektrotechniku a energetiku Roč. 13, mimoriadne čislo, (pp. 267 - 269). Trenčin.

AFC Publikované príspevky na zahraničných vedeckých konferenciách

Vrabec, J., & Harley, D. (2010). Real Performance? The 19th Annual EICAR Conference. Paris: Eicar.

Vrabec, J., Rozinaj, G., & Talafová, R. (2008). IQ Kiosk in Metropolitan Information Systems. Proceedings Elmar-2008 : 50th International Symposium ELMAR-2008 (pp. 1 - 638). Zadar: Croatian Society Electronics in Marine.

Vrabec, J., Rozinaj, G., & Vojtko, J. (2007). MABox - Multimodal Microphone Array Algorithm Development System. 2007 IWSSIP & EC-SIPMCS. 14th International Workshop on Systems, Signals & Image Processing and EURASIP Conference Focused on Speech & Image Processing, Multimedia Communications & Services (pp. 281 - 283). Maribor: University of Maribor.

Vrabec, J., & Rozinaj, G. (2007). IQ Kiosk - Multimedia Intelligent Terminal. Proceedings Elmar-2007 : 49th International Symposium Elmar-2007 focused on Mobile Multimedia (pp. 203 - 206). Zadar: Croatian Society Electronics in Marine.

AFD Publikované príspevky na domácich vedeckých konferenciách.

Vrabec, J., Rozinaj, G., & Černochová, L. (2009). Evaluation of project IQ Kiosk and future work. Redžúr : 3rd International Workshop on Speech and Signal Processing (s. 79 - 81). Bratislava: STU v Bratislave FEI.

Čepko, J., Talafová, R., & Vrabec, J. (2008). Indexing Join Costs for Faster Unit Selection Synthesis. IWSSIP 2008. 15th International Workshop on Systems, Signals and Image Processing (s. 503 - 506). Bratislava: STU v Bratislave.

Janata, J., & Vrabec, J. (2008). Application of Beamforrming Methods in Speech Analysis (Delay & Sum Beamformer). Redžúr 2008 : 2nd International Workshop on Speech and Signal Processing (pp. 9 - 12). Bratislava: STU v Bratislave FEI.

Rozinaj, G., & Vrabec, J. (2007). Human Computer Interaction for Data Retrieval from Internet. Proceedings of Redžúr : 1st International Workshop on Speech and Signal Processing (pp. 59 - 62). Bratislava: FEI STU.

FAI Redakčné a zostavovateľské práce knižného charakteru (bibliografie, encyklopédie, katalógy, slovníky, zborníky...)

Rozinaj, G., Vrabec, J., & Vojtko, J. (2007). Proceedings of Redžúr : 1st International Workshop on Speech and Signal Processing. Bratislava: FEI STU.

Rozinaj, G., & Vrabec, J. (2008). Redžúr 2008 : 2nd International Workshop on Speech and Signal Processing. Bratislava: STU v Bratislave FEI.

Rozinaj, G., Vrabec, J., & Körösi, J. (2009). Proceedings Redžúr 2009 : 3rd International Workshop on Speech and Signal Processing. Bratislava: STU v Bratislave FEI.

Rozinaj, G., Čepko, J., Trúchly, P., Vrabec, J., & Vojtko, J. (2008). IWSSIP 2008 - 15th International Workshop on Systems, Signals and Image Processing. Bratislava: STU v Bratislave.

Bažant, I., Čepko, J., Hájek, J., Hrad, J., Nevosad, M., Podhradský, P., . . . Zeman, T. (2007). Fundamentals of Teleworking for Disadvantaged Persons. Nitra: Agrogenofond.

Bažant, I., Čepko, J., Hájek, J., Hrad, J., Nevosad, M., Podhradský, P., . . . Zeman, T. (2007). Základy teleworkingu pre znevýhodnené osoby. Nitra: Agrogenofond.

10 OHLASY

Vrabec, J., & Harley, D. (2010). Real Performance? *The 19th Annual EICAR Conference.* Paris: Eicar**. Bol citovaný v článkoch** Košinár, P., Malcho, J., Marko, R., & Harley, D. (2010). AV TESTING EXPOSED. *Virus Bulletin Conference* pp. 281-287 Vancouver: Virus Bulletin. **a** L´evesque, F. L., Davis, C. R., Fernandez, J., M., Chiasson, S., Somayaji, A. (2012). Methodology for a Field Study of Anti-Malware Software. Workshop on Usable Security (USEC 2012), Divi Flamingo Beach Resort, Bonaire

Talafova, R., Rozinaj, G., Čepko, J., & Vrabec, J. (2008). Multimedia SMS Reading in Mobile Phone. International Journal of Mathematics and Computers in Simulation Vol. 1, Iss. 1., 12 - 17. **Bol citovaný v článkoch** Burileanu, D., Surmei, M., Negrescu, C., Catalin, Unguren, C., Dervis, A., (2010). Multi-Access Network-Based TTS Architectures, In Romanian jurnal of information, science and technology Volume 13, pp. 310–320 **a** Surmei, M., Burileanu, D., Negrescu, C., Ungurean, C., Dervis, A. (2009). Real-time architectures for a network-based text-to-speech service implementation. In Speech Technology and Human-Computer Dialogue, 2009. SpeD '09. pp. 1-6

Rozinaj, G., Vrabec, J., Čepko, J., & Talafová, R. (2008). Terminals for the Smart Information Retrieval. In I. K. IBRAHIM, Handbook of Research on Mobile Multimedia : Vol.1.,2. (Zv. 2, s. 1 - 274). Hershey: Information Science Reference. **Bol citovaný v článku** Kondelova, A. , Toth, J., Guzmicky, P. (2011). Simulation of prosody contours with embeded signal generator In proceedings of 18th International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 1- 4

Čepko, J., Talafová, R., & Vrabec, J. (2008). Indexing Join Costs for Faster Unit Selection Synthesis. IWSSIP 2008. 15th International Workshop on Systems, Signals and Image Processing pp. 503-506. Bratislava: STU v Bratislave. **Bol citovany v článkoch** Chea, L. Y., (2009). Implementation of simulated annealing in unit selection for malay text-to-speech system, In. A dissertation submitted in fulfillment of the requirements for the award of the degree of Master of Science (Mathematics) on Faculty of Science Universiti Teknologi Malaysia **a** Tan, T.,S., Chea, L., Y., Zainuddin, Z.,M., Shaikh, S.,H.,S., & Chai, H., Y., (2011). Unit selection for Malay text-to-speech system using segmental context and simulated annealing, Scientific Research and Essays Vol. 6(20), pp. 4281-4289 **a** Kastelan, I., Katona, M. ; Papp, I. ; Davidovic, M. ; Resetar, I. (2011), An Integrated Audio and Video Communication System for Digital Television Sets, 2nd Eastern European Regional Conference on the Engineering of Computer Based Systems (ECBS-EERC), pp. 78- 84

11 RESUMÉ

The aim of this thesis is to analyze the performance impact of speech processing algorithm in multimodal interfaces. Every algorithm has unique performance requirements and if the performance impact is underestimated the operation of applications or algorithms is delayed or causes errors, on the other hand if the performance resources of hardware are overestimated, the result is higher costs and shorter battery lifetime. For this reason it is crucial to exactly measure the required performance of an algorithm and correct any problems and optimize during early stages of development, to ensure that it works within the desired tolerances. For precise measurements it is necessary to strictly follow the methodology and minimize external influence. The analysis of an algorithm's impact on resource usage for audio and video processing poses quite new topic for research.

In the first part of the thesis we focus on measuring the algorithm's impact on system resources and analyze the existing approaches to this issue. In the next part we propose methodology for measuring system performance of applications and algorithm in multimodal interfaces and proposes an approach how to minimize interference during measurements. After that, we propose and implement an evaluation method, which calculates and generates graphs showing the algorithm's impact on system performance. In closing we validate that the methodology and algorithms were correctly designed and implemented by measurements on model algorithms.