



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS**

Remigijus Kutas

**SISTEMINĖS APLINKOS OPTIMIZAVIMAS VMA „MOODLE“
NAŠUMUI BEI GREITAVEIKAI**

Baigiamasis magistro projektas

Vadovas
Doc. D. Rutkauskienė

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

**SISTEMINĖS APLINKOS OPTIMIZAVIMAS VMA „MOODLE“
NAŠUMUI BEI GREITAVEIKAI**

Baigiamasis magistro projektas

Nuotolinio mokymosi informacinės technologijos (621E14002)

Vadovas

(parašas) Doc. D. Rutkauskienė
(data)

Recenzentas

(parašas) Lekt. R. Kubiliūnas
(data)

Projektą atliko

(parašas) Remigijus Kutas
(data) 2015 05 18

KAUNAS, 2015



KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

(Fakultetas)

Remigijus Kutas

(Studento vardas, pavardė)

Nuotolinio mokymosi informacinės technologijos (621E14002)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „SISTEMINĖS APLINKOS OPTIMIZAVIMAS VMA „MOODLE“ NAŠUMUI
BEI GREITAVEIKAI“
AKADEMINIO SAŽININGUMO DEKLARACIJA

2015 m. gegužės 18 d.
Kaunas

Patvirtinu, kad mano, **Remigijaus Kuto**, baigiamasis projektas tema „SISTEMINĖS APLINKOS OPTIMIZAVIMAS VMA „MOODLE“ NAŠUMUI BEI GREITAVEIKAI“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

TURINYS

Santrauka	7
Summary.....	8
Iliustracijos	9
Lentelės.....	11
Pagrindinės sąvokos	12
Įvadas.....	13
1. Veiksniai, didinantys VMA „Moodle“ apkrovą	15
2. Žiniatinklio programų spartinimo metodų ir priemonių apžvalga.....	17
2.1 Techninė įranga.....	19
2.2 Operacinė sistema	19
2.3 HTTP protokolas – galimybės spartinti jo darbą	20
2.4 Nuolatinių jungčių (<i>keep-Alive</i>) naudojimas.....	21
2.5 HTTP užklausų mažinimas	22
2.5.1 Image Maps naudojimas	22
2.5.2 CSS Sprites naudojimas.....	22
2.5.3 Stilių bei JavaScript failų apjungimas	23
2.5.4 Paralelinis komponentų siuntimas	23
2.5.5 Expires antraščių naudojimas	25
2.6 Informacijos glaudinimas GZIP formatu	25
2.7 ETAG konfigūravimas	26
2.8 Peradresavimu vengimas.....	27
2.9 Naudotojo sąsajos optimizavimo galimybės.....	27
2.9.1 Stiliaus failų aprašymas	27
2.9.2 JavaScript failų aprašymas	27
2.9.3 JavaScript ir CSS kodo perkėlimas į atskirus failus	27
2.9.4 JavaScript minimizavimas	27
2.9.5 CSS minimizavimas.....	28

2.10	PHP interpretatoriaus darbo spartinimas	28
2.10.1	PHP mašininio kodo spartinimas.....	28
2.11	PHP aplinkos kintamųjų spartinimas	30
2.12	Žiniatinklio serveris ir kiti paskirstymo sprendimai.....	30
2.12.1	Apache	31
2.12.2	Nginx	31
2.12.3	Turinio paskirstymo tinklas	31
2.12.4	Tarpiniai (proxy) serveriai.....	32
2.12.5	Klasteriai.....	32
2.13	Duomenų bazių valdymo sistema (DBVS)	32
3.	Našumo bei greitaveikos vertinimo metodika bei įrankiai	34
3.1	Testavimo metodika.....	34
3.1.1	Testavimo aplinkos identifikavimas	34
3.1.2	Pakankamos greitaveikos lygio identifikavimas	35
3.1.3	Testų planavimas	35
3.1.4	Testavimo aplinkos konfigūravimas.....	35
3.1.5	Testo parengimas	35
3.1.6	Testo įvykdymas.....	35
3.1.7	Rezultatų analizė, ataskaitos.....	35
3.2	Greitaveikos vertinimo įrankiai	35
3.2.1	AB.....	36
3.2.2	Siege	37
4.	Našumą bei greitaveiką lemiančių veiksnių tyrimas	38
4.1	Eksperimentų atlikimo tvarka bei testavimo metodika.....	38
4.2	Testavimo aplinka bei jos konfigūracija	38
4.3	Techninė įranga.....	39
4.3.1	Procesoriaus bei operatyviosios atminties įtaka	39
4.3.2	Diskinės posistemės bei spartinančiųjų saugyklų įtaka.....	40

4.4	Operacinė sistema	43
4.5	Žiniatinklio serveris	45
4.6	Duomenų bazių valdymo sistema	47
4.7	Srauto paskirstymo priemonių bei SSL šifravimo įtaka	50
4.8	PHP programų stekas	54
4.9	„Moodle“ savybės	55
4.9.1	„Moodle“ versijų įtaka.....	55
4.9.2	Turinio kiekio įtaka.....	57
4.9.3	„Moodle“ klasterio ir jo komponentų skaičiaus įtaka.....	59
4.10	„Moodle“ našumo įvertinimas.....	62
4.11	Eksperimentinio tyrimo apibendrinimas	66
	Išvados	69
	Literatūra	70
	Priedai	72

Kutas, R. Sisteminės aplinkos optimizavimas VMA „Moodle“ našumui bei greitaveikai. Informacinių technologijų magistras / vadovas Doc. D. Rutkauskienė; Kauno technologijos universitetas, Informatikos fakultetas.

Kaunas, 2015. 71 p.

SANTRAUKA

Darbe nagrinėjama VMA „Moodle“ naudotojų užklausų trukmės priklausomybė nuo techninės bei sisteminės programinės įrangos konfigūracijos. Šio darbo tikslas yra testuoti ir patikrinti VMA „Moodle“ našumo bei greitaveikos pokyčius, taikant skirtingus sisteminės aplinkos komponentus ir jų konfigūravimo parametrus, ir pasiūlyti efektyvią VMA „Moodle“ naudotojų užklausų trukmės mažinimo metodiką.

Skyriuje „Veiksniai, didinantys VMA „Moodle“ apkrovą“ nagrinėjamas poreikis užtikrinti pakankamą VMA „Moodle“ našumą bei greitaveiką. Analizuojami veiksniai, didinantys „Moodle“ aplinkos apkrovą: naudotojų skaičiaus augimas, apkrovos pikai.

Skyriuje „Žiniatinklio programų spartinimo metodų ir priemonių apžvalga“ atliekama literatūros analizė. Analizės tikslas – nustatyti sisteminės įrangos komponentus ir jų konfigūravimo parametrus, turinčius įtaką VMA „Moodle“ našumui bei greitaveikai. Skyriuje taip pat išanalizuotos našumo ir greitaveikos testavimo metodikos, parinkta metodika VMA „Moodle“ testuoti.

Eksperimentinis tyrimas aprašomas skyriuje „Našumą bei greitaveiką lemiančių veiksnių tyrimas“. Tyrimo eigoje atlikti bandymai, kurių tikslas – nustatyti įvairių sisteminės aplinkos komponentų įtaką VMA „Moodle“ našumui bei greitaveikai. Eksperimentinio tyrimo rezultatai parodė, kad techninės įrangos pajėgumas – pagrindinis veiksnys, lemiantis VMA „Moodle“ našumą ir greitaveiką. „Moodle“ aptarnaujančios sistemos procesoriaus pajėgumo didinimas arba kelių sistemų apjungimas į klasterį užtikrina didžiausią „Moodle“ našumo bei greitaveikos augimą. Palyginti su procesoriaus našumu, diskinės posistemės greitis turi santykinai mažą įtaką „Moodle“ našumui bei greitaveikai. Pasiekus pakankamą operatyviosios atminties kiekį, tolesnis jos apimtys didinimas „Moodle“ greitaveikai įtakos nedaro. Nagrinėjant sisteminę programinę įrangą ir jos įtaką „Moodle“ greitaveikai bei našumui nustatyta, kad didžiausią įtaką daro DBVS programinės įrangos parinkimas ir jos konfigūravimo modifikacijos.

Darbo rezultatai atskleidė efektyvius „Moodle“ našumo bei greitaveikos didinimo būdus. Jais remiantis parengtos praktikoje pritaikomos rekomendacijos „Moodle“ našumui ir greitaveikai užtikrinti.

SUMMARY

The optimization of system environment for the performance of Moodle LMS

This paper presents Virtual Learning Environment Moodle user's request time dependence on server hardware and system software configuration. The aim of this work is to test and evaluate VLE Moodle speed and performance changes using different system environment components and their configuration parameters. The aim is to suggest VLE Moodle user's request time reduction methodology.

As described in chapter "Factors that increase the VLE Moodle load", here the need to ensure sufficient VLE speed and performance is analysed. Factors which boost Moodle environment work which are analysed include user growth and system load during the peak period.

The chapter "Overview of the World Wide Web software speeding up methods and measures" presents theoretical approach based on scientific literature analysis. The aim of the analysis is to establish new parameters of system software and configuration measures which may impact VLE Moodle speed and performance. This chapter also comprises analysed speed and performance testing methodologies and chosen methodology for VLE Moodle testing.

Experimental research is described in chapter „Investigation of the main factors in underlying speed and performance“. The aim of performed tests is to determine the influence of various system environment components for VLE Moodle speed and performance. The results obtained from experimental research showed that technical equipment capability is the main factor in ensuring VLE Moodle speed and performance. Moodle system CPU enhancement in performance or system's interconnection into a cluster ensures the highest Moodle speed and performance. In comparison with processor's performance, the speed of sub-systems has little impact on Moodle speed and performance. When the sufficient amount of data memory is reached, the further growth of the memory does not affect Moodle speed and performance. When analysing system software and its influence on Moodle speed and performance it was found that the greatest impact has the choice of relational database software and its configuration modifications.

The results of the research revealed effective ways of increasing Moodle speed and performance. Based on the results, recommendations which can be implemented in practice are prepared to maintain and ensure Moodle speed and performance.

ILIUSTRACIJOS

1 pav.	Veiksniai, didinantys VMA apkrovą.....	15
2 pav.	HTTP užklausų skaičiaus kaita studentų žinių patikros metu.....	16
3 pav.	Parodytų puslapių skaičiaus kaita aukštosios mokyklos mokymosi semestro eigoje	16
4 pav.	PHP programos komponentai (Armando Padilla, 2010).....	18
5 pav.	Piešinukai apjungti į CSS Sprite	23
6 pav.	Puslapio komponentai parsiončiami po du.....	24
7 pav.	Keturi komponentai parsiončiami paraleliai	24
8 pav.	PHP kodo vykdymo seka	29
9 pav.	PHP kodo vykdymo seka naudojant mašininio kodo spartinimą.....	29
10 pav.	Greitaveikos vertinimo etapai	34
11 pav.	Eksperimento komponentų schema.....	39
12 pav.	vCPU ir RAM kiekio didinimo įtaka	40
13 pav.	Eksperimento komponentų schema.....	41
14 pav.	Diskinės posistemės bei spartinančiųjų saugyklų įtaka.....	42
15 pav.	Eksperimento komponentų schema.....	44
16 pav.	Operacinės sistemos įtaka	45
17 pav.	Eksperimento komponentų schema.....	46
18 pav.	WWW serverio programinės įrangos įtaka	47
19 pav.	A ir B eksperimento komponentų schemas variantai.....	48
20 pav.	C, D ir E eksperimento komponentų schemas variantai	48
21 pav.	DBVS įtaka.....	50
22 pav.	A eksperimento komponentų schema.....	51
23 pav.	B eksperimento komponentų schema.....	51
24 pav.	C eksperimento komponentų schema.....	52
25 pav.	D eksperimento komponentų schema.....	52
26 pav.	Srauto paskirstymo priemonių bei SSL šifravimo įtaka.....	53
27 pav.	Eksperimento komponentų schema.....	54
28 pav.	PHP interpretatoriaus versijų įtaka.....	55
29 pav.	Eksperimento komponentų schema.....	56
30 pav.	„Moodle“ programinio kodo įtaka	57
31 pav.	Eksperimento komponentų schema.....	57
32 pav.	Turinio kiekio įtaka	58
33 pav.	Eksperimento komponentų schema.....	60

34 pav.	„Moodle“ klasterio bei jo komponentų skaičiaus įtaka.....	61
35 pav.	Eksperimento komponentų schema.....	62
36 pav.	„Moodle“ našumo įvertinimas.....	63
37 pav.	CPU apkrova A testavimo aplinkos variante	64
38 pav.	Aktyvios RAM kiekis KB A testavimo aplinkos variante	64
39 pav.	Disko apkrova KBps A testavimo aplinkos variante.....	65
40 pav.	Duomenų srautai tinkle KBps A testavimo aplinkos variante	65
41 pav.	Testų metu fiksuoti maksimalūs našumo bei greitaveikos padidėjimai, susiję su techninės įrangos konfigūracijos pokyčiais	66
42 pav.	Testų metu fiksuoti maksimalūs našumo bei greitaveikos padidėjimai, susiję su programinės įrangos konfigūracijos pokyčiais	67
43 pav.	Testų metu fiksuotas maksimalus našumo bei greitaveikos neigiamas pokytis, susijęs su „Moodle“ versija bei turiniu	68

LENTELĖS

1 lentelė.	Skirtingų apimčių mokymosi kursai.....	38
2 lentelė.	Testavimo aplinkos konfigūracija.....	39
3 lentelė.	vCPU ir RAM kiekio didinimo įtaka.....	40
4 lentelė.	Testavimo aplinkos konfigūracija.....	41
5 lentelė.	Diskinės posistemės bei spartinančiųjų saugyklų įtaka.....	42
6 lentelė.	Testavimo aplinkos konfigūracija.....	43
7 lentelė.	Operacinės sistemos įtaka.....	44
8 lentelė.	Testavimo aplinkos konfigūracija.....	46
9 lentelė.	WWW serverio programinės įrangos įtaka.....	47
10 lentelė.	Testavimo aplinkos konfigūracija.....	49
11 lentelė.	DBVS įtaka.....	49
12 lentelė.	Testavimo aplinkos konfigūracija.....	52
13 lentelė.	Srauto paskirstymo priemonių bei SSL šifravimo įtaka.....	53
14 lentelė.	Testavimo aplinkos konfigūracija.....	54
15 lentelė.	PHP interpretatoriaus versijų įtaka.....	55
16 lentelė.	Testavimo aplinkos konfigūracija.....	56
17 lentelė.	„Moodle“ programinio kodo įtaka.....	56
18 lentelė.	Testavimo aplinkos konfigūracija.....	58
19 lentelė.	Turinio kiekio įtaka.....	58
20 lentelė.	Testavimo aplinkos konfigūracija.....	60
21 lentelė.	„Moodle“ klasterio bei jo komponentų skaičiaus įtaka.....	61
22 lentelė.	Testavimo aplinkos konfigūracija.....	63
23 lentelė.	„Moodle“ našumo įvertinimas.....	63

PAGRINDINĖS SĄVOKOS

HTTP (trumpinys angl. *HyperText Transfer Protocol*) – protokolas naudojamas informacijai pasiekti žiniatinklyje.

HTML (trumpinys angl. *Hyper text Markup Language*) - tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete.

CPU (trumpinys angl. *Central processing unit*) - tai loginis įtaisas, apdirbantis duomenų srautą.

PHP (trumpinys angl. *Hypertext Preprocessor*) - plačiai paplitusi dinaminė interpretuojama programavimo kalba, specialiai pritaikyta interneto svetainių kūrimui.

„**Moodle**“ (trumpinys angl. *Modular Object Oriented Dynamic Learning Environment*) - atvirojo kodo virtualiojo mokymosi aplinka, suprojektuota padėti pedagogams organizuoti mokymosi kursus internete.

MySQL - reliacinių duomenų bazių valdymo sistema, palaikanti daugelį naudotojų, dirbanti SQL kalbos pagrindu.

Mašininis kodas - mašininės kalbos instrukcijos, kuriomis apibrėžiamos paruoštos vykdymui operacijos.

VMA (trumpinys **V**irtualioji **M**okymosi **A**plinka) – programinė įranga leidžianti suburti besimokančiuosius į virtualiąją klasę ir aprūpinti juos įvairiomis mokymosi galimybėmis.

WWW (trumpinys angl. *World Wide Web*) – žiniatinklis, saitynas, pasaulinis tinklas.

GNU (trumpinys angl. *General Public License arba GPL*) - laisvosios programinės įrangos licencija, pradžioje sukurta GNU projektui. Šiuo metu tai viena populiariausių atvirojo kodo licencijų.

GZIP – failo tipas talpinantis suspaustą informaciją arba informacijos suspaudimo tipas.

ĮVADAS

Vienas svarbiausių veiksnių, pastaraisiais dešimtmečiais lemiančių esminius pokyčius formuojant studijų aplinką švietimo institucijose, yra intensyvus informacinių komunikacinių technologijų (IKT) diegimas. Tai lemia tiek bendros įvairių gyvenimo sričių kompiuterizavimo tendencijos, tiek mokymosi institucijų poreikiai modernizuoti studijų aplinką. Todėl šiandien daugelis Lietuvos švietimo institucijų turi ne tik kompiuterių tinklus, kompiuterizuotas auditorijas ar bibliotekas, bet ir virtualiąsias mokymosi aplinkas (VMA).

VMA taikymas šiandien būdingas ne tik nuotoliniam mokymuisi, bet dažnai yra svarbus ir tradiciniame mokyme ar studijose. Lietuvos švietimo institucijose daugelis dėstomų dalykų bent iš dalies naudoja įvairias virtualiųjų mokymosi aplinkų galimybes: medžiagos talpinimą, žinių patikrinimo priemones. Šiandienos švietimo institucijose VMA naudotojų skaičius nuolatos auga. Atvirų švietimo iniciatyvų dėka institucijos naudotojų skaičius dažnai viršija institucijoje besimokančiųjų skaičių. Nuolat augantys aktyvių naudotojų skaičiai, mokymosi objektų kiekiai bei mokymosi medžiagos apimtys kelia naujus iššūkius institucijų serverinei infrastruktūrai bei VMA serverius prižiūrintiems administratoriams.

„Moodle“ – viena populiariausių virtualiųjų mokymosi aplinkų Lietuvoje, suprojektuota organizuoti mokymosi kursus tinkle ir atitinkanti kursų valdymo sistemų grupę. Tačiau sistema taip greitai plėtojama, kad jau dabar ją galima vadinti mokymosi valdymo sistema. Dėl naudojimo Lietuvoje populiarumo, VMA „Moodle“ našumas bei greitimeika buvo pasirinkti pagrindiniu šio tyrimo objektu. Tyrimo eigoje atlikta įvairių sisteminės aplinkos komponentų įtakos VMA „Moodle“ našumui analizė. Tyrimo darbo rezultatai atskleidė efektyviausius „Moodle“ našumo bei greitimeikos didinimo būdus. Jais remiantis parengtos sisteminės aplinkos konfigūravimo gairės, padėsiančios optimaliai išnaudoti techninius išteklius ir užtikrinti maksimalų VMA „Moodle“ darbo našumą bei greitimeiką.

Darbo tikslas ir uždaviniai

Šio darbo objektas yra VMA „Moodle“ naudotojų užklausų trukmės priklausomybė nuo techninės bei sisteminės programinės įrangos konfigūracijos.

Šio darbo tikslas yra testuoti ir patikrinti VMA „Moodle“ našumo bei greitimeikos pokyčius, taikant skirtingus įrangos komponentus ir jų konfigūravimo parametrus, ir pasiūlyti efektyvią VMA „Moodle“ naudotojų užklausų trukmės mažinimo metodiką.

Darbo uždaviniai:

1. Išanalizuoti sisteminės įrangos tipus, turinčius įtaką VMA „Moodle“ našumui bei greitimeikai.
2. Išanalizuoti našumo bei greitimeikos testavimo metodikas, parinkti metodiką VMA „Moodle“ testavimui.

3. Pasirinkta testavimo metodika testuoti VMA „Moodle“ naudotojų užklausų trukmę, naudojant skirtingus sisteminės įrangos komponentų derinius. Nustatyti naudotojų užklausų trukmę mažinančius sisteminės įrangos komponentų derinius.
4. Atlikti VMA „Moodle“ bandomuosius diegimus su pasirinktais techninės bei programinės įrangos komponentų deriniais. Atlikti šių diegimų testavimus su skirtingo dydžio naudotojų grupėmis. Išanalizuoti testavimo rezultatus ir, remiantis jais, koreguoti komponentų derinius, siekiant pagerinti greitaveiką.
5. Parinkti techninės bei programinės įrangos komponentų derinį, kuris maksimaliai sumažintų naudotojų užklausų trukmę. Parengti tipines techninės ir programinės įrangos diegimo bei konfigūravimo rekomendacijas.

Darbo struktūra

Skyriuje „Veiksniai, didinantys VMA „Moodle“ apkrovą“ nagrinėjamas poreikis užtikrinti pakankamą VMA „Moodle“ našumą bei greitaveiką. Analizuojami veiksniai, didinantys „Moodle“ aplinkos apkrovą: naudotojų skaičiaus augimas, apkrovos pikai.

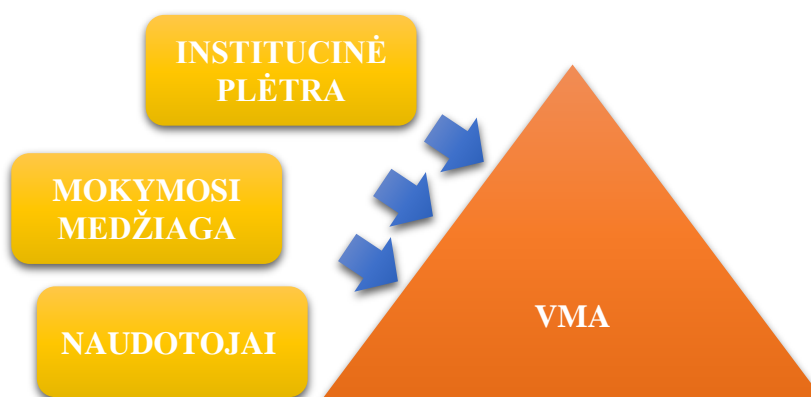
Skyriuje „Žiniatinklio programų spartinimo metodų ir priemonių apžvalga“ atliekama literatūros analizė. Analizės tikslas – nustatyti sisteminės įrangos komponentus ir jų konfigūravimo parametrus, turinčius įtaką VMA „Moodle“ našumui bei greitaveikai.

Eksperimentinis tyrimas aprašomas skyriuje „Našumą bei greitaveiką lemiančių veiksnių tyrimas“. Tyrimo tikslas – nustatyti įvairių sisteminės aplinkos komponentų įtaką VMA „Moodle“ našumui bei greitaveikai.

Remiantis tyrimo darbo rezultatais, parengtos sisteminės aplinkos konfigūravimo gairės.

1. VEIKSNIAI, DIDINANTYS VMA „MOODLE“ APKROVĄ

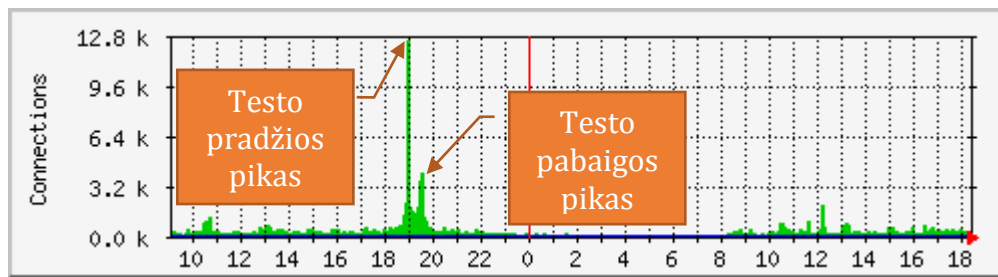
VMA „Moodle“ programinė įranga šiandien – tai ne tik nuotolinių studijų organizavimo priemonė. Tai kasdieninio naudojimo priemonė įvairiuose mokymuose. Dėl patogaus mokymo proceso organizavimo, medžiagos talpinimo, žinių tikrinimo galimybių, „Moodle“ naudojama įvairiems taikymo scenarijams platesnėje auditorijoje. „Moodle“ naudojamas jau ne tik aukštosiose mokyklose ar specializuotuose nuotolinio mokymo centruose, bet ir tokiose institucijose, kaip vidurinio lavinimo mokyklos, suaugusiųjų švietimo centrai, nacionalinės valdžios įstaigos (pvz., LR Seimas ar VMI), verslo konsultacinės bendrovės ar didelės korporacijos.



1 pav. Veiksniai, didinantys VMA apkrovą

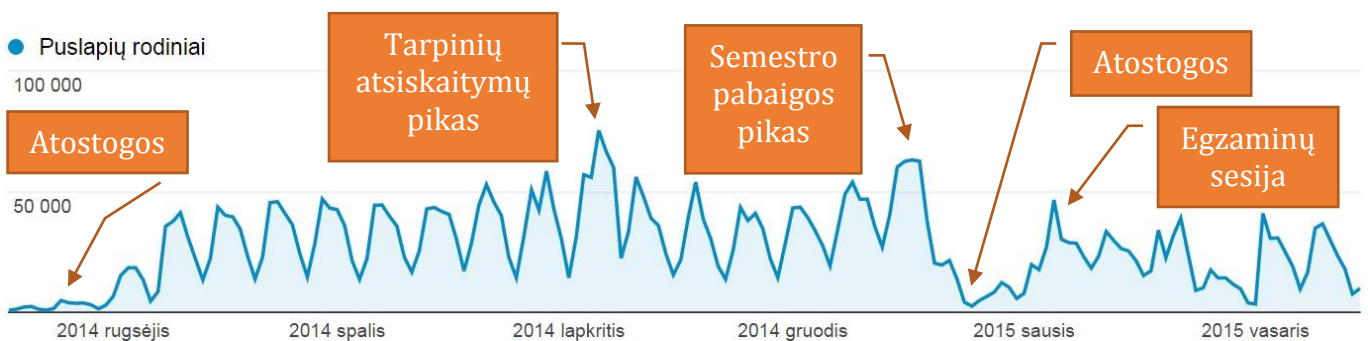
Dėl platesnės VMA naudojimo skvarbos į tradicinius mokymosi metodus jos naudojimo apimties nuolat auga (Moodle: A Low-Cost Solution for Successful e-Learning, 2015). Naudojimas auga vykstant švietimo institucijų plėtrai. Augimą skatina neformaliojo švietimo, atvirų švietimo iniciatyvų paplitimas. Išvardyti veiksniai didina naudotojų bei mokymosi objektų skaičių (Capterra, 2014; Lambda Solutions, 2013). Tai neišvengiamai didina reikalavimus techninės bei programinės įrangos galimybėms aptarnauti išaugusius naudotojų poreikius. Svarbiausi veiksniai, didinantys VMA apkrovą, pateikti 1 pav.

VMA „Moodle“ aplinkos apkrova nėra tolygi. Mokymosi proceso eigoje „Moodle“ gali patirti apkrovos padidėjimo pikus. Apkrovos pikai – tai staigūs „Moodle“ apkrovos padidėjimai, atsirandantys dėl mokymosi proceso ypatumų. Apkrovos intensyvumas piko metu nuo įprastinės apkrovos gali skirtis dešimtis kartų. 2 pav. pateiktas pavyzdys, iliustruojantis, kaip kinta HTTP užklausų skaičius, kai 300 studentų laiko žinių patikrinimo testą. Testo pradžios laikas fiksuotas nuo 19:00 iki 19:30. Testui atlikti skirta 30 min.



2 pav. HTTP užklausų skaičiaus kaita studentų žinių patikros metu

Išskirtini du apkrovos padidėjimo pikai. Pirmasis susietas su testo pradžia – testo aktyvavimo pradžios laikas fiksuotas, o jo aktyvumo periodas pakankamai trumpas. Studentai nekantriai laukia testavimo pradžios, nuolat atnaujindami VMA „Moodle“ puslapį su nuoroda į testą. Sistemai aktyvavus testą didžioji studentų dalis pradeda atlikti testą pirmąją jo aktyvavimo minutę. Pirmasis apkrovos pikas pasižymėjo 8,5 karto didesniu užklausų skaičiumi palyginti su užklausų skaičiumi testo laikymo eigoje. Antrasis pikas sukeltas fiksuotos testo trukmės – vienu metu pradėję testo laikymą jie ir baigia panašiu laiku išsekus testui skirtam laikui. Kadangi klausimų atsakymams studentai užtrunka skirtingą laiko tarpą, tai antrasis apkrovos pikas (2,5 karto) nėra toks ryškus ir labiau pasiskirstęs laike.



3 pav. Parodytų puslapių skaičiaus kaita aukštosios mokyklos mokymosi semestro eigoje

Vertinant ilgos trukmės laiko periodus, apkrovos pikai susiję su mokymosi periodais: semestro pradžia, pabaiga, egzaminų sesija. Pavyzdžiui, aukštosios mokyklos studentai intensyviau naudojami VMA tarpinių semestro atsiskaitymų, semestro pabaigos ar sesijos metu (3 pav.) (Xin Guo, Qing Shi, Danjue Zhang, 2013).

Mokymosi institucijų plėtra, VMA panaudojimo atveju, naudotojų skaičiaus, mokymosi medžiagos apimties augimas didina nuolatinę VMA tenkančią apkrovą. Todėl apkrovos piko metu atsiranda galimybė sistemos darbas sutrikimams. Įvairūs VMA darbo sutrikimai, ypač pasitaikantys žinių patikrinimo metu nepageidaujami. Todėl VMA aptarnaujančios sistemos turi būti parengtos dirbti su apkrovomis dešimtis kartų viršijančias vidutines. Tam būtina ištirti „Moodle“ našumo bei greitaveikos priklausomybę nuo techninės bei programinės įrangos.

2. ŽINIATINKLIO PROGRAMŲ SPARTINIMO METODŲ IR PRIEMONIŲ APŽVALGA

Virtualioji mokymosi aplinka „Moodle“ – modulinė PHP ir MySQL pagrindu sukurta sistema. Šiuo metu „Moodle“ sudaro apie 7000 PHP kodo failų. Nagrinėjant būdus optimizuoti „Moodle“ greitaveiką, pritaikant dideliems naudotojų srautams aptarnauti, svarbiausias dėmesys turi būti skiriamas į PHP programoms būdingas greitaveikos problemas. Todėl „Moodle“ funkcionavimui būtinas sisteminių komponentų derinys, būdingas tipinei PHP programai vykdyti. Tai techninės įrangos, operacinės sistemos, žiniatinklio (WWW) serverio, PHP interpretatoriaus bei duomenų bazių valdymo sistemos derinys. Veiksniai, darantys įtaką VMA „Moodle“ greitaveikai yra iš esmės tie patys, kaip ir kiekvienai žiniatinklio programai parašytai PHP programavimo kalba. Žiniatinklio serverio darbo optimizavime reiktų susitelkti ties veiksniais, kurie turės didžiausią reikšmę naudotojui. Pavyzdžiui, jei yra daugiau naudotojų, intensyviau naršančių statinius duomenis nei naudojančių komponentus, lemiančius duomenų bazės greitaveiką, rekomenduotina spartinti žiniatinklio serverį (Moodle performance recommendations, 2014).

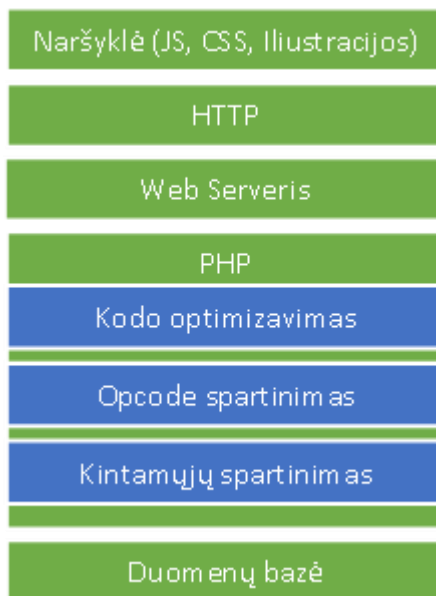
„Moodle“ architektūra sukurta atsižvelgiant į potencialų poreikį aptarnauti daug naudotojų. Didelį naudotojų skaičių aptarnaujančios svetainės paprastai atskiria žiniatinklio serverį ir duomenų bazių aptarnavimo serverį į skirtingus fizinius serverius. Taipogi, galimas VMA „Moodle“ diegimas, naudojantis apkrovos balansavimo metodus, kai naudojamas daugiau kaip vienas žiniatinklio serveris. Atskiras žiniatinklio serveris prijungiamas prie tos pačios duomenų bazės ir susietas su ta pačia „Moodle“ failų talpykla bei laikinų objektų podėliu. Bet dažniausiai VMA „Moodle“ darbo greitaveikai užtikrinti taikomas tik žiniatinklio ir DBVS serverių atskyrimas. Esant nepakankamai DBVS greitaveikai, panašiai kaip žiniatinklio serveriai, DBVS serveriai gali būti apjungiami į serverių grupę, pvz.: MySQL klasterį („Moodle“ performance recommendations, 2014).

Vienas iš pagrindinių veiksnių, lemiančių VMA „Moodle“ naudotojo užklausų trukmę yra PHP interpretatoriaus darbo greitaveika. PHP yra labai populiarus programavimo kalba, naudojama daugelyje populiarių interneto svetainių. PHP kodo interpretavimo greitis gali labai keistis, atsižvelgiant į PHP interpretatoriaus versiją, žiniatinklio serverio sisteminės aplinkos bei PHP kodo sudėtingumą. Todėl nustatant PHP greitaveikos problemas reikia identifikuoti taškus, turinčius neigiamą įtaką greitaveikai. Dažnai PHP kodo greitaveikos problemos susijusios ne su jo optimizavimu, o su duomenų baze ar tinklu. PHP kodo vykdymą spartina PHP *bytecode* podėlių, paskirstytos atminties objektų podėlių naudojimas. Šios priemonės spartina PHP programų veikimą, kaupdamos jau sukompiliuoto PHP kodo fragmentus ar duomenų objektų kopijas laikinos atminties podėlyje. PHP naudoja atminties buferį, kad sukauptų duomenis, kuriuos programa ruošia išvedimui. Šis buferis gali turėti neigiamos įtakos laikui, per kurį naudotojui atverčiamas žiniatinklio puslapis. Pasak E. Higgins, galima padaryti tam tikrus PHP

interpretatoriaus nustatymų pakeitimus, kurie privers PHP greičiau atnaujinti išvedimo buferius, ir svetainė naudotojams taps greitesnė (Eric Higgins, 2014).

Literatūros analizės tikslas - apibrėžti potencialias PHP programos greitaveikos problemas ir jų sprendimo galimybes bei išnagrinėti, ar galimi šių problemų sprendinių taikymai VMA „Moodle“ atžvilgiu. Nes nekeičiant programos turinio (išeities kodo), tik dalį greitaveikos optimizavimo sprendimų galima įgyvendinti. Tuo tarpu, net 80% žiniatinklio puslapio krovimo laiko priklauso nuo jo turinio (King, 2008). Pagrindinis dėmesys literatūroje skiriamas žiniatinklio puslapio optimizavimo problemoms naršyklės lygmenyje spręsti. Literatūroje minimas didelis skaičius, veiksnių darančių įtaką greitaveikai (Armando Padilla, 2010; King, 2008; Padilla & Hawkins, 2010; Moodle performance recommendations, 2014; Sheltren, Newton, & Catchpole, 2014; Souders, 2007). Žemiau pateikta iliustracija išvardija komponentus, sudarančius žiniatinklio programos ekosistemą bei galinčius lemti VMA „Moodle“ greitaveiką. Daugelis PHP programų įgyvendina savo sąsają su naudotoju žiniatinklio naršyklės pagalba. Sąsajai užtikrinti naudojamos HTML, JS, CSS, Flash bei kitos žiniatinklio programos. Naršyklėje atvaizduota programos sąsaja HTTP protokolu per žiniatinklio serverį komunikuoja su programos PHP kodu, kuris savo ruožtu dirba su duomenimis iš išorinio šaltinio – duomenų bazės.

Greitaveika priklauso nuo išvardytų PHP programos darbą įgyvendinančių komponentų (4 pav.). Todėl nagrinėjant problemas, susijusias su VMA „Moodle“ greitaveika bei būdus kaip tas problemas spręsti, reikia nagrinėti tipinės PHP programos komponentus.



4 pav. PHP programos komponentai (Armando Padilla, 2010)

Pagrindiniai greitaveikos spartinimo aspektai, susiję su PHP interpretavimo procesu būtų šie: PHP kodo optimizavimas, PHP mašininio kodo spartinimas, duomenų mainų su išoriniais duomenų šaltiniais spartinimas. Tikėtina, kad VMA „Moodle“ PHP kodas pakankamai gerai parašytas. Optimizuoti „Moodle“

kodą labiau nei tai padarė kūrėjai – mažai tikėtina. Tuo tarpu operacijų su PHP mašininio kodu bei duomenų objektais spartinimas talpinant juos spartinančiose atmintinėse „Moodle“ yra būtinas.

2.1 TECHNINĖ ĮRANGA

Operatyvioji atmintis. Didinant VMA „Moodle“ greitaveiką reiktų nepamiršti techninių parametrų. Greičiausias ir efektyviausias pakeitimas pagerinti „Moodle“ greitaveiką – operatyviosios atminties kiekio padidėjimas. Operatyviosios atminties didėjimas sumažins poreikį informacijos mainams su diskiniu kaupikliu ir leis sumažinti naudotojų užklausų trukmę bei padidinti lygiagrečiai aptarnaujamų užklausų kiekį. Diskinės sistemos greitis taipogi labai svarbus sklandžiam „Moodle“ darbui. Todėl diskinės sistemos našumo gerinimas pagerins „Moodle“ sistemos, ypač tokių posistemų, kaip „Moodle“ pranešimai, darbą. Jei „Moodle“ *moodledata* katalogas talpinama pakankamai lėtoje saugykloje (pavyzdžiui, NFS, SAN) rekomenduojama laikinų objektų podėlį perkelti į spartesnę failų sistemos vietą (Moodle performance recommendations, 2014).

Procesorius. HTML puslapių teikimas, paprastai, nesukelia ypatingos CPU apkrovos. Tačiau sudėtingiems dinamiškiems tinklalapiams, generuojamiems PHP interpretatoriaus, CPU greitis gali būti vienas iš lemiamų greitaveikos ribojimo veiksnių.

Tinklo jungtis. Vienas iš dažniausių „Moodle“ naudotojų užklausų trukmę didinančių veiksnių – nepakankamas serverio tinklo jungties pralaidumas. Sakykim, serveris turi 10 Mbit/s sąsają su internetu, ja galima siųsti 1 megabaitą duomenų per sekundę. Ir nors kiekvienas „Moodle“ tinklalapis bus tik 30 kb, 33 naudotojų užklausos per sekundę, viršys interneto jungties pralaidumą.

2.2 OPERACINĖ SISTEMA

Operacinė sistema (OS) – speciali programinė įranga, užtikrinanti naudotojo sąsają ir kompiuterio techninės įrangos, taikomųjų programų bei duomenų valdymą. Pradžioje „Moodle“ buvo sukurtas „Linux“ operacinei sistemai, naudojančiai Apache žiniatinklio serverį, MySQL duomenų bazes ir PHP kalbos interpretatorių. „Linux“ ir „Windows“ OS dažniausiai naudojamos VMA „Moodle“ talpinimui. „Linux“ OS vertinama kaip optimali „Moodle“ diegimui OS. „Moodle“ kūrėjai dažniausiai šia VMA bando „Debian“, „Ubuntu“, „CentOS“, „RedHat“, „Windows“ ir „Mac OS X“ operacinėse sistemose, todėl tikėtina, kad šiose OS „Moodle“ darbas vyks sklandžiausiai (moodle.org, 2015).

Spartinančioji atmintinė naudojama procesų tarpusavio komunikacijai, daugialypių procesų podėliui duomenims ir kodui saugoti (Free & open source, high-performance, distributed memory object caching system, 2014). Jei nepakankamas bendrosios atmintinės kiekis procesuose, naudojančiuose bendrąją atmintinę, pvz., duomenų bazės ryšiai, ar įvykdomas kodas, atsiras vėlinimas.

Prieiga prie duomenų, esančių standžiajame diske, gali būti nuo 50 iki 100 kartų lėtesnė negu duomenų, esančių operatyviojoje atmintyje. Failų sistemos podėliai, naudojantys operatyviąją atmintį, gali procesus paspartinti. Darbą sulėtinti gali dažnas simbolinių nuorodų naudojimas UNIX sistemose.

Tokiose operacinėse sistemose, kaip „Windows“, naujo proceso sukūrimas trunka ilgai (Padilla & Hawkins, 2010). Tai reiškia, kad CGI užklauskos nauji procesai startuos lėčiau ir užklauskos bus vykdomos ilgiau nei kitose operacinėse sistemose. Reikia vengti perpildyti žiniatinklio serverį per dideliu kiekiu nereikalingų procesų. Pavyzdžiui, jei serveris yra tikrai žiniatinkliui aptarnauti, rekomenduojama vengti tokių procesų, kaip X-„Windows“ paleidimo (Padilla & Hawkins, 2010).

2.3 HTTP PROTOKOLAS – GALIMYBĖS SPARTINTI JO DARBĄ

Prieš nagrinėjant specifines žiniatinklio programų greitaveikos problemas, reikalingos žinios apie HTTP protokolo veikimo principus ir jo poveikį greitaveikai.

HTTP (trumpinys *angl.* HyperText Transfer Protocol) – pagrindinis metodas pasiekti informaciją žiniatinklyje.

Pradinė protokolo paskirtis – pateikti standartinį būdą HTML puslapiams skelbti ir skaityti. HTTP kūrimą koordinavo World Wide Web konsorciumas ir Internet Engineering Task Force darbo grupė. Svarbiausias HTTP protokolo aprašymas yra RFC 2616, apibrėžiantis HTTP/1.1, plačiausiai naudojamą HTTP versiją (Hypertext Transfer Protocol, 2014).

HTTP yra užklauskimo - atsakymo protokolas, jungiantis klientą ir serverį. HTTP klientas, pavyzdžiui, naršyklė, inicijuoja užklauską prisijungdamas TCP/IP ryšiu prie tam tikro nutolusio kompiuterio prievado („porto“) (pagal nutylėjimą naudojamas 80). HTTP serveris laukia kliento užklauskos, pavyzdžiui, „GET/HTTP/1.1“ (užklauskiančios pradinio serverio puslapio) bei susijusio MIME informacinio pranešimo, nusakančio užklauskos prigimtį, taip pat kartais po informacinio pranešimo gali sekti papildoma informacija. Gavęs užklauską, serveris ją apdoroja ir grąžina naudotojui atsakymo eilutę („200 OK“ sėkmės atveju), ir susijusį pranešimą, kuris dažniausiai yra HTML puslapis. HTTP užklauskos:

- GET – dažniausia užklauską, reikalaujanti tam tikro resurso duotu URL adresu;
- POST – panašu į GET, bet siunčiama papildoma informacija, paprastai raktų ir reikšmių poros, nusakančios HTML formos laukų reikšmes;
- PUT – naudojama failų dėjimui į serverį;
- DELETE – naudojama trynimui, retai palaikoma;
- HEAD – panaši į GET, tačiau reikalaujama tik atsakymo antraštė, o pats atsakymo dokumentas nelaukiamas;
- TRACE – grąžina gautą užklauską, gali būti naudojama tikrinimui, kokių papildomų meta duomenų prideda tarpiniai serveriai;

- CONNECT – retai palaikoma.

Nagrinėjant žiniatinklio programų greitaveikos problemas, daugiausia dėmesio skiriama GET tipo HTTP užklausoms. GET tipo užklausa sudaro URL nuoroda, lydima antraščių. HTTP atsakymą sudaro operacijos statuso kodas, antraštės bei atsakymo turinys. Žemiau pateiktas paprastas kliento – serverio bendravimo pavyzdys:

Kliento užklausa:

```
GET /test.txt HTTP/1.1
Host: vma.elinara.lt
```

Serverio atsakymas:

```
HTTP/1.1 200 OK
Date: Mon, 16 Jun 2014 07:56:35 GMT
Server: Apache/2.2.15 (CentOS)
Last-Modified: Mon, 16 Jun 2014 07:49:47 GMT
ETag: "c0038-9-4fbef456fdb42"
Accept-Ranges: bytes
Content-Length: 9
Content-Type: text/plain; charset=UTF-8

test data
```

2.4 NUOLATINIŲ JUNGČIŲ (*KEEP-ALIVE*) NAUDOJIMAS

HTTP protokolo veikimas pagrįstas TCP protokolo veikimu. Ankstyvose HTTP protokolo realizacijose kiekvienai HTTP užklausiai buvo atidaromas nauja TCP jungtis. Tai nebuvo efektyvu, nes klientas į serverį siuntė didelį HTTP užklausų kiekį. Nuolatinės jungtys arba kitaip *Keep-Alive* atsirado HTTP /1.0 specifikacijoje tam, kad būtų išvengta didelio kiekio naujų TCP jungčių tarp serverio ir kliento kūrimo. Klientas nurodo apie gebėjimą palaikyti *Keep-Alive Connection*, antraštės pagalba. Serverio atsakyme taip pat naudojama *Connection* antraštė:

```
GET /test.txt HTTP/1.1
Host: vma.elinara.lt
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Wed, 18 Jun 2014 05:39:24 GMT
Server: Apache/2.2.15 (CentOS)
```

```
Last-Modified: Mon, 16 Jun 2014 07:49:47 GMT
ETag: "c0038-9-4fbef456fdb42"
Accept-Ranges: bytes
Content-Length: 9
Keep-Alive: timeout=3, max=300
Connection: Keep-Alive
```

VMA „Moodle“ atžvilgiu nuolatinių TCP jungčių naudojimas rekomenduojamas dėl didelio VMA „Moodle“ puslapius sudarančių komponentų skaičiaus. Kadangi nuolatinių TCP jungčių įgalinimas nereikalauja „Moodle“ modifikavimo, tikėtina sėkminga realizacija.

2.5 HTTP UŽKLAUSŲ MAŽINIMAS

VMA „Moodle“ atžvilgiu turime ganėtinai ribotas galimybes spartinti „Moodle“ naudotojo sąsajos darbą. Ši programa nėra vietinis produktas, todėl jos modifikacijų galimybės apsiriboja tik dizaino šablonu. Apie 10 – 20% žiniatinklio puslapio krovimo laiko tenka HTML dokumento parsisiuntimui, kiti 80 – 90% laiko sunaudojama puslapio komponentų parsisiuntimui (Souders, 2007). Todėl, paprasčiausias kelias sumažinti puslapio krovimo laiką yra papildomų HTTP dokumento komponentų skaičiaus sumažinimas.

Siūlymas išmesti kai kuriuos puslapio komponentus, iškelia pasirinkimo klausimą tarp puslapio krovimo laiko ir puslapio dizaino. Todėl šią rekomendaciją reikia išpildyti naudojant įvairias technikas, mažinančias HTTP užklausų skaičių, nekeičiant puslapio išvaizdos.

2.5.1 Image Maps naudojimas

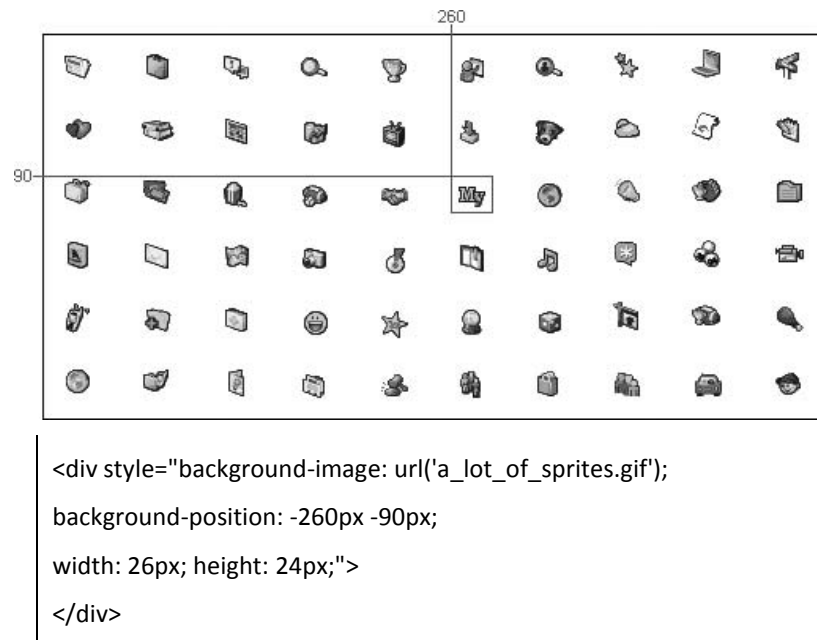
Paprasčiausia žiniatinklio navigacijos forma – teksto asociacija su URL. Vizualesnė forma – paveiksluko asociacija su URL. Taip įgyvendinami žiniatinklio puslapių mygtukai ar meniu. Jei taip panaudojama serija paveikslėlių, sukuriama daug puslapio komponentų, kuriems parsiusiti bus inicijuojamos HTTP užklausos ir ilginamas puslapio krovimo laikas. Paprasčiausias problemos sprendimo būdas – naudoti *image maps*. Tada vienas piešinukas asocijuojamas su keletu URL, taip sumažinamas puslapio komponentų skaičius ir sumažinamas puslapio generuojamų HTTP užklausų skaičius.

Image maps panaudojimas „Moodle“ gali būti problematinis, kadangi navigacija „Moodle“ aplinkoje įgyvendinta tekstinėmis nuorodomis, navigacijos komponentai – piešinukais ir, paprastai, būna išsklaidyti po puslapio plotą.

2.5.2 CSS Sprites naudojimas

Taip kaip *image maps*, *CSS Sprites* leidžia apjungti iliustracijas, tik daro tai daug lanksčiau. Pavyzdžiui, galima puslapio navigacijoje naudojamus piešinėlius sutalpinti į vieną failą. Ir paskui reikiamoje vietoje CSS naudoti su pasirinktu elementu, palaikančiu fono piešinius. Pvz.: SPAN arba DIV

HTML elementai. Elementui parenkama reikalinga dalis iš bendro piešinio naudojant CSS *background-position* savybę (5 pav.).



5 pav. Piešinukai apjungti į CSS Sprite

VMA „Moodle“ atžvilgiu, šis metodas gali būti naudojamas kuriant temą. Taip galima realizuoti įvairias puslapio dizaine naudojamas piktogramas. Piktogramas apjungus į vieną piešinį, galima atsisakyti daugelio HTTP užklausų.

2.5.3 Stilių bei JavaScript failų apjungimas

JavaScript ir CSS šiandien plačiai naudojami puslapių komponentai. Reikia apjungti skirtingus JavaScript bei CSS failus. Taip bus generuojama mažiau HTTP užklausų ir mažinamas puslapio krovimo laikas. Šios technikos naudojimo perspektyvos VMA „Moodle“ atžvilgiu nėra aiškios ir apribotos dizaino šablono modifikavimo galimybėmis, todėl toliau šiame darbe nebus analizuojamos.

2.5.4 Paralelinis komponentų siuntimas

Pasak Sounders, žiniatinklio puslapį sudarančių komponentų skaičius labiausiai lemia puslapio krovimosi laiką (Souders, 2007). Kiekvienas komponentas, nerastas spartinančioje atmintinėje, generuoja papildomą HTTP užklausą. HTTP/1.1 specifikacija numato, kad vienu metu iš to pačio domeno gali būti parsisuntinėjami du puslapio komponentai (6 pav.). Daugelis puslapių savo komponentus siunčia iš vieno domeno.



6 pav. Puslapio komponentai parsiončiami po du

Komponentų paskirstymas keliuose domenuose padidintų vienu metu parsiončiamų komponentų skaičių. Paraleliai parsiončiamų komponentų skaičiaus padidinimas sutrumpina puslapio krovimo laiką. Jeigu komponentai tolygiai paskirstomi per du domenų, komponentų užkrovimo laikas sutrumpėja dvigubai (7 pav.) (Souders, 2007). Naudotojas gali keisti savo naršyklės parametrus ir taip priversti parsiončiamą vienu metu daugiau nei du komponentus, bet daug geriau naudoti CNAME DNS paslaugos įrašus ir taip išskaidyti puslapio komponentus į keletą domenų. Didelis kiekis paraleliai parsiončiamų puslapio komponentų bus įtakojamas tinklo pralaidumu bei CPU pajėgumas, todėl bandymas puslapio komponentus atsiųsti paraleliai, nebūtinai sutrumpins puslapio atvaizdavimo laiką. Todėl, daugelyje šaltinių rekomenduojama puslapio komponentus paskirstyti tarp dviejų domenų naudojantis CNAME įrašais.



7 pav. Keturi komponentai parsiončiami paraleliai

„Moodle“ atžvilgiu šios metodikos panaudojimas kol kas netaikomas. Bet yra galimybė atlikti eksperimentinius bandymus modifikuojant grafinių failų URL. Modifikavimui atlikti galima panaudoti Apache serverio *mod_rewrite* direktyvas, pakeičiant domeno pavadinimo dalį nuorodose į grafinius turinio elementus.

2.5.5 Expires antraščių naudojimas

Naršyklės bei spartinantieji (*proxy*) serveriai naudoja spartinančiąją atmintį mažindami HTTP užklausų skaičių ir taip sumažindami WWW puslapio krovimo laiką. WWW serveris naudoja *Expires* antraštę kliento informavimui, apie laiką, kaip ilgai jis gali naudoti spartinančioje atmintinėje laikomą puslapio resursus:

```
HTTP/1.1 200 OK
Content-Type: application/x-javascript
Last-Modified: Wed, 22 Feb 2014 04:15:54 GMT
Expires: Wed, 05 Oct 2014 19:16:20 GMT
```

Kai naršyklė randa *Expires* antraštę, ji išsaugo komponentą ir jo galiojimo laiką spartinančioje atmintinėje. Kitą kartą prireikus to paties komponento, jei jo galiojimo laikas dar nepasibaigęs, naršyklė nebepateikia HTTP užklausos į žiniatinklio serverį, bet pasiima reikiamą puslapio komponentą iš spartinančiosios atmintinės.

Expires antraštę geriausiai apibrėžti *ExpiresDefault* direktyva. Dažnai *Expires* antraštė nustatoma iliustracijoms, bet nereikėtų tuo apsiriboti ir rekomenduojama nustatinėti *Expires* antraštę statiniams VMA „Moodle“ komponentams: JS, CSS (Souders, 2007).

2.6 INFORMACIJOS GLAUDINIMAS GZIP FORMATU

Taip pat, kaip bandoma sumažinti failų dydžius, taip pat galima sumažinti HTTP atsakymo dydį suglaudinant jį GZIP formatu. Pradedant HTTP/1.1, naršyklė gali identifikuoti gebėjimą apdoroti suglaudintą informaciją naudojant HTTP užklausos *Accept-Encoding* antraštę (Souders, 2007):

```
Accept-Encoding: gzip, deflate
```

Žiniatinklio serveris, radęs užklausoje *Accept-Encoding* antraštę, gali suspausti atsakymo turinį kliento nurodytu suglaudavimo formatu. Apie turinio suglaudimą žiniatinklio serveris klientui nurodo *Content-Encoding* antrašte:

```
GET / HTTP/1.1
host: vma.elinara.lt
Accept-Encoding: gzip, deflate
```

```
HTTP/1.1 200 OK
Date: Mon, 16 Jun 2014 11:21:04 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.5.7 ZendServer/6.3.0
Expires: Mon, 20 Aug 1969 09:23:00 GMT
Content-Encoding: gzip
Content-Length: 5458
Content-Type: text/html; charset=utf-8
í<Ér7gó+àí'µRjñXVk BÔT *QU`e&²deÐxq|M:ïasCø_óÜ²dKíäÜÑ-Lào_°¼¼|üòÑÉëWOÈTGáÑÆþpyè
```

GZIP šiuo metu pats populiariausias ir efektyviausias suspaudimo metodas (Souders, 2007). Tai yra nemokamas ir neribojamas jokiais patentais formatas, sukurtas GNU projekto rėmuose ir standartizuotas RFC 1952. Kitas galimas informacijos glaudinimo formatas yra *deflate*, bet jis ženkliai mažiau efektyvus ir populiarus. Naršyklės palaikančios *deflate* formatą palaiko ir GZIP, bet yra keletas naršyklių kurios palaiko tik GZIP. Todėl HTTP atsakymų turinį rekomenduojama glaudinti GZIP formatu. Informacijos glaudinimą rekomenduojama naudoti spaudžiant HTML, JavaScript ir CSS failus, bet nerekomenduojama glaudinti paveikslukų ir PDF failų nes juos sudaro jau suglaudinta informacija.

2.7 ETAG KONFIGŪRAVIMAS

Vientisumo žymės (*Entity tags, ETags*) mechanizmas, kurio pagalba naršyklės identifikuoja spartinančioje atmintyje talpinamus objektus. Kai naršyklė parsisiunčia puslapio komponentą, jis - patalpinamas spartinančioje atmintinėje. Kito to pačio puslapio krovimo metu „šviežias“ puslapio komponentas bus kraunamas iš spartinančiosios atmintinės. Kai spartinančioje atmintinėje komponentas „pasensta“, naršyklė tikrina jo tinkamumą naudojimui, siųsdama serveriui sąlyginę GET užklausa. Tam, yra būtina papildoma HTTP užklausa, bet jei komponentas tinkamas naudojimui vietoj 200 tipo atsakymo, talpinančio visą komponentą, serveris grąžina daug mažesnę 304 tipo atsakymą „*Not Modified*“ (nemodifikuota). Tam kad patikrintų, ar tikrai naršyklės turimas komponentas atitinka serverio turimą puslapio komponentą, tikrinama failo modifikavimo data arba vientisumo žymė ETag.

Problemos su ETag, susijusios su jų generavimo prigimtimi. Pvz.: Apache 1.3 ir 2.x ETag generuoja naudodama OS failo atributą `inode-size-timestamp`. Todėl jei sistemą aptarnauja keli žiniatinklio serveriai, ETag parametras bus generuojamas skirtingai, ir žiniatinklio serveris tikrindamas komponento tinkamumą, greičiausiai, grąžins jį pakartotiniam parsisiuntimui. Lygiai tos pačios problemos susidaro naudojant spartinančiuosius serverius – žiniatinklio serverio sugeneruotas ETag nesutampa spartinančiajame serveryje išsaugoto failo ETag (Souders, 2007).

Etag naudojimas dažnai iššaukia perteklines HTTP užklausas, todėl rekomenduojama jų naudojimo atsisakyti.

2.8 PERADRESAVIMU VENĖIMAS

Peradresavimas, paprastai, naudojimas pakeisti vieną URL adresą kitu. Peradresavimai galimi keleto tipų, bet dažniausiai pasitaiko 301 ir 302 tipai. Peradresavimai dažniausiai realizuojami HTML dokumentams, bet gali būti atlikti ir puslapių komponentams (piešinukams, JavaScript failams ir t.t.) Nauja svetainės versija, naudotojų stebėsenos procesai, URL supaprastinamas patrauklumui naudotojui didinti, tai keletas priežasčių iššaukiančių peradresavimų naudojimą (Souders, 2007).

2.9 NAUDOTOJO SAŠAJOS OPTIMIZAVIMO GALIMYBĖS

2.9.1 Stilias failų aprašymas

Progresinis vaizdo generavimas vyksta naršyklei užkrovus stiliaus failus iki bus užkrauti JavaScript failai. Todėl, norint maksimaliai išnaudoti progresinio vaizdo generavimo naudą, stilių aprašymų failų užkrovimo nuorodos talpinamos puslapio HEAD dalyje.

2.9.2 JavaScript failų aprašymas

Jei naudojami stiliai, progresinis atvaizdavimas blokuojamas iki visiško jų užkrovimo. Todėl reikalinga stiliaus failus aprašyti puslapio HEAD skiltyje.

2.9.3 JavaScript ir CSS kodo perkėlimas į atskirus failus

Norint atskirti statinį turinį nuo dinaminio, rekomenduojama atskirti JavaScript ir CSS failus nuo HTML dokumentų į atskirus failus.

2.9.4 JavaScript minimizavimas

Minimizavimas – perteklinių simbolių šalinimo iš kodo praktika, siekiant sumažinti jo dydį bei sutrumpinti krovimo laiką. Kai kodas minimizuojamas, iš jo pašalinami komentarai, nereikalingi tarpai, eilučių pabaigos ir panašiai. JavaScript atveju, tai ženkliai sumažina pradinį JavaScript failo dydį taip pagreitinant jo parsisiuntimą.

„Moodle“ JavaScript failai nėra minimizuoti. JavaScript failų minimizavimas, „Moodle“ atveju, gali turėti įtakos greitimeikai dėl JavaScript failų gausos bei sudėtingumo. Tiesa, reikia nepamiršti, kad kaskart atnaujinus „Moodle“ versiją JavaScript failų minimizavimo procesą reikės atlikti iš naujo.

Yra agresyvesnis JavaScript failų mažinimo procesas nei minimizavimas – „Obfuscation“ (apgadinimas). Šio proceso metu iš JavaScript failo šalinama ne tik perteklinė informacija, bet kartu

parametrų bei metodų pavadinimai keičiami trumpesniais. Taip pasiekiamas dar mažesnis JavaScript failo dydis. Šis procesas įneša nepatogumo palaikymo, klaidų taisymo procesams, bet kadangi mes į „Moodle“ žiūrime tik iš naudotojo perspektyvos, tai kodo tobulinimo klausimus galime ignoruoti. GZIP naudojimas dar papildomai sumažina JavaScript failų failus. Dydžių skirtumas tarp minimizuotų ir *obfuscation* proceso pagalba sumažintų failų sumažėja naudojant Gzip (Souders, 2007).

2.9.5 CSS minimizavimas

Siunčiamos informacijos sumažėjimas, minimizuojant CSS failus, ženkliai mažesnis lyginant su JavaScript, nes CSS turi mažiau komentarų bei tuščių tarpų, o jo vidinė struktūra ženkliai kompaktiškesnė. Todėl didžiausias potencialas mažinant CSS failų dydį – identiškų klasių apjungimas, nenaudojamų klasių naikinimas. Šių rekomendacijų reikia laikytis, rengiant VMA „Moodle“ dizaino šablonus.

2.10 PHP INTERPRETATORIAUS DARBO SPARTINIMAS

VMA „Moodle“ programinis kodas parašytas PHP kalba. PHP – plačiai paplitusi dinaminė interpretuojama programavimo kalba (*Hypertext Preprocessor*), sukurta 1995 m. ir specialiai pritaikyta interneto svetainių kūrimui. PHP sintaksė panaši į daugelį struktūrinių kalbų, ypač į C bei Perl. PHP kalbos interpretatorius yra atviro kodo programinė įranga. Programuoti PHP kalba nesudėtinga, nes ji gana lanksti – veikia daugumoje operacinių sistemų, palaiko nemažai reliacinių duomenų bazių bei veikia su dauguma žiniatinklio serverių – CGI, FastCGI, ISAPI ir kitais protokolais.

PHP kalbos kodas vykdomas interpretavimo būdu. Tai pagrindinis PHP trūkumas, neigiamai veikiantis „Moodle“ našumą bei greitaveiką. Nes kiekvienas PHP kalba parašytas programinio kodo failas yra kompiliuojamas prieš jam įvykdant. Interpretavimas papildomai apkrauna serverio procesorių bei prailgina PHP kodo vykdymo trukmę.

Šiandien „Moodle“ veikimui minimali PHP interpretatoriaus versija yra 5.4.4. „Moodle“ naudojami PHP plėtiniai: *iconv, mbstring, curl, openssl, tokenizer, xmlrpc, soap, ctype, zip, gd, simplexml, spl, pcre, dom, xml, intl, json*. „Moodle“ diegimo vedlys sustabdys diegimą sisteminės aplinkos tikrinimo procedūroje, jei neras bet kurio iš būtinų php plėtinių. PHP 5.5 ir naujesnės versijos savyje turi *OPcache* plėtinį. Tai neprivalomas plėtinys, bet jo buvimas ženkliai gerina php kodo greitaveiką (moodle.org, 2015).

2.10.1 PHP mašininio kodo spartinimas

Kaskart, kai vykdomas PHP kodas iškvietus jį komandine eilute ar per WWW serverį, PHP turi atlikti penkis žingsnius (8 pav.). PHP interpretatorius turi nuskaityti failą, patikrinti PHP kodo sintaksę, suinterpretuoti kodą, sukurti mašininį kodą ir galiausiai jį įvykdyti.

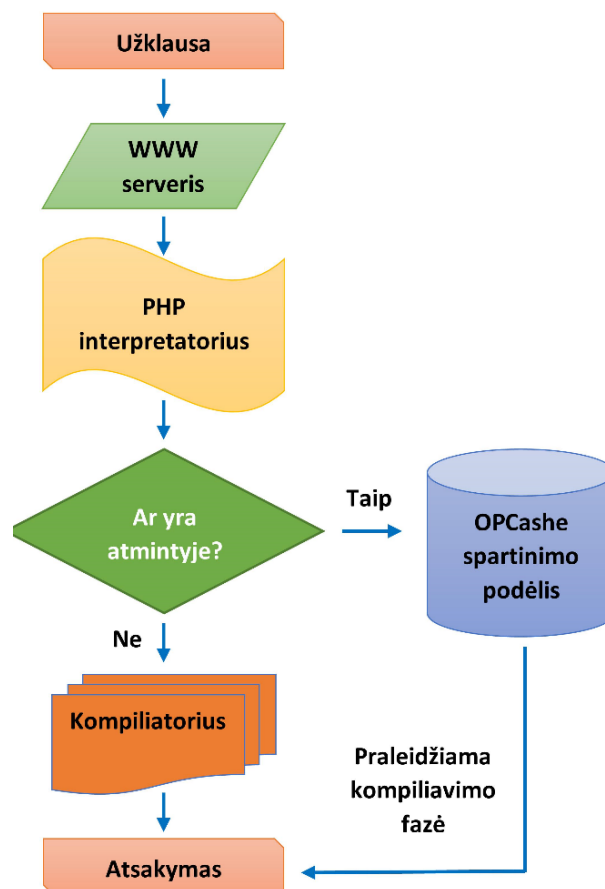


8 pav. PHP kodo vykdymo seka

PHP kodo vykdymo ciklo gyvavimas itin trumpas, bet kaskart reikalinga atlikti visus kodo vykdymo etapus. Kaskart vykdant PHP kodą kuriamas mašininis kodas, nors pats kodas prieš tai nebuvo modifikuojamas. PHP kompiliavimas nėra būtinas įgalinus mašininį kodą saugojimą spartinančioje atmintinėje. Pakartotinai iškvietus to paties PHP kodo vykdymą galima bus praleisti pirmuosius kodo vykdymo sekos žingsnius:

1. patikrinti PHP kodo sintaksę;
2. interpretuoti kodą;
3. sukurti mašininį kodą.

Pakartotinai iškvietus PHP kodo vykdymą, PHP interpretatorius paima mašininį kodą iš spartinančios atmintinės ir inicijuoja jo vykdymą (9 pav.) (Padilla & Hawkins, 2010).



9 pav. PHP kodo vykdymo seka naudojant mašininio kodo spartinimą

Yra daug PHP plėtinių, įgalinančių PHP mašininio kodo saugojimą spartinančioje atmintinėje. Pvz.: Alternative PHP Cache (APC), XCache, eAccelerator (eA), Zend Opcache (seniau Zend Optimizer+) (Wikipedia - List of PHP accelerators, 2014). VMA „Moodle“ sudarytas iš itin didelio PHP kodą talpinančių failų skaičiaus. Jų vykdymas naudotojui dirbant sistema nuolat kartojasi, todėl taikyti mašininio kodo spartinimo metodus „Moodle“ atžvilgiu būtina.

Prieš keletą metų PHP interpretatorius, pagal numatytą reikšmę, neturėdavo bytecode spartinimo priemonių. Todėl bytecode spartinimo priemonių diegimas tokiose sistemose lemdavo keletą ar keliolika kartų siekiantį ženklų greitimeikos prieauglį. Daugelis šiandieninių „Linux“ distribucijų, ar PHP diegimo paketų kitoms operacinėms sistemoms aprūpinti PHP interpretatoriumi su pagal nutylėjimą įgalintu bytecode spartinimu.

2.11 PHP APLINKOS KINTAMŲJŲ SPARTINIMAS

PHP kodas pasiima duomenis iš duomenų bazės. Duomenų bazės atidarymas, SQL užklauso nusiuntimas, gautų duomenų apdorojimas bei duomenų bazės uždarymas - tipiniai šios sekos žingsniai. PHP interpretatoriaus darbo su DBVS metu galimi vėlinimai. Dėl duomenų bazių serverio apkrovos, lentelės, iš kurios gaunami duomenys, struktūros optimizacijos, duomenų bazės valdiklio darbo optimalumo galimi vėlinimai, įtakojantys užklauso vykdymo laiką ir tuo pačiu PHP kodo vykdymo greitį. Pakartotinės arba lygiagrečios tų pačių duomenų paieškos duomenų bazėje gali tapti programos darbo sulėtėjimo priežastimi. Todėl duomenų bazėje laikomų objektų talpinimas į spartinančiąją atmintinę apsaugo nuo kelių potencialių grėsmių greitimeikai.

2.12 ŽINIATINKLIO SERVERIS IR KITI PASKIRSTYMO SPRENDIMAI

Vienas svarbiausių PHP programos ekosistemos komponentų - žiniatinklio serveris. Žiniatinklio serverio pasirinkimas įtakoja ne tik greitimeika bet ir techninių resursų panaudojimą (Sheltren, Newton, & Catchpole, 2014). Žiniatinklio serverio darbo sutrikimai tiesiogiai įtakos žiniatinklio programos greitimeiką.

Žiniatinklio serveris svarbus, VMA „Moodle“ funkcionavimui, sisteminės programinės įrangos komponentas. Žiniatinklio serveris – programinė įranga, kurios veikimas grįstas klientas-serveris komunikavimo principu. Programos kodas vykdomas nutolusiame žiniatinklio serveryje, o šios programinės įrangos funkcijas naudotojas pasiekia naudodamasis klientu (internetu naršykle).

„Moodle“ aptarnavimui rekomenduojamas Apache HTTP žiniatinklio serverio programinė įranga. IIS, Lighttpd, Nginx, Cherokee, Dzeus ir LiteSpeed naudojami rečiau. „Moodle“ veikimas šiuose žiniatinklio serveriuose gali būti mažiau ištestuotas arba palaikomos ne visos „Moodle“ funkcijos.

Žiniatinklio serveris turi būti parengtas vykdyti PHP programas. Žiniatinklio serverio versija nėra esminis veiksnys, bet rekomenduojama panaudoti naujausią žiniatinklio serverį (moodle.org, 2015).

2.12.1 Apache

Apache yra vienas iš svarbiausių žiniatinklio plėtrai. Apache buvo viena pirmųjų alternatyvų Netscape Communications Corporation tinklo serveriui, dabar žinomam Sun Java System Web Server vardu. Savo funkcionalumu ir našumu Apache konkuruoja su kitais Unix operacinės sistemos pagrindu kuriamais tinklo serveriais.

Į Apache tinklo serverį įtraukti moduliai, kurie išplečia serverio funkcionalumą - serverių programavimo kalbos (mod_perl, mod_python, Tcl, ir PHP), įgalinimo moduliai (mod_access, mod_auth ir mod_digest). Įtrauktas SSL ir TLS (mod_ssl), tarpinio serverio modulio palaikymas, URL (nuorodų) redaktorius, custom log failai ir duomenų filtravimo palaikymas. Apache pranešimų failai gali būti analizuojami naudojant naršyklę ir laisvai platinamus redaktorius AWStats/W3Perl, Visitors ar kt.

Virtual hosting funkcija leidžia vienai Apache programai aptarnauti kelias internetines svetaines. Pavyzdžiui, vienas kompiuteris su viena Apache programa gali vienu metu aptarnauti kelias svetaines.

Apache viena iš savybių yra konfigūruojami klaidų pranešimai ir DBVS, kuri yra paremta duomenų bazių įgalinimu ir turinio perdavimu. Taip pat Apache yra suderinamas su keletu grafinių sąsajų.

2.12.2 Nginx

Svetainės greitaveiką galima didinti tiek konfigūruojant serverį, pvz., „sunkiasvorio“ Apache pakeitimas į „lengvesnius“, pvz., Nginx. Nginx tai yra HTTP ir *reverse proxy* serveris. Šis serveris pasižymi didele sparta bei stabilumu.

2.12.3 Turinio paskirstymo tinklas

CDN veikia debesų technologijų principu. Ta pati informacija yra saugoma ne viename dedikuotame galingame serveryje, bet keliuose ar net keliolikoje po skirtingas šalis išmėtytuose serveriuose. Kai lankytojas pateikia užklausimą atsisiųsti tam tikrą turinį, duomenys yra perduodami ne iš pagrindinio serverio, bet iš arčiausiai tos šalies esančio serverio. Pavyzdžiui, lankytojas iš Lietuvos nueina į tam tikrą puslapį ir paspaudžia „Parsiųsti failą“. CDN paslauga iš karto nustato, kuris serveris su esančiu failu yra arčiausiai ir būtent iš to serverio lietuviai atsisiunčia norimą failą. Bet kuriam dideliame tarptautiniam interneto projektui, kuriame pagrindinis vaidmuo tenka vaizdinei, grafinei informacijai – nuotraukoms, vaizdo, garso įrašams, dokumentams ir kitiems didesnių apimčių failams. Svarbu, kad šių projektų naudotojai būtų pasklidę skirtinguose geografiniuose regionuose. CDN paslauga leidžia susitvarkyti su ypač dideliais duomenų perdavimo srautais. Kai duomenys perduodami iš arti esančio serverio, tai gali atsiliepti ir perdavimo greičiui – puslapiai atsidarinėja greičiau. Šios paslaugos klientai – tarptautiniu mastu populiarūs projektai. Dėl to sunku būtų įvardinti konkretų lietuvišką projektą, kuris galėtų naudotis tokia

paslauga. Lietuvoje CDN panaudoti prasminga, nebent teikiant masinius atvirusius kursus. Kitais „Moodle“ panaudojimo atvejais dalis naudotojų lokalūs, ir dėl Lietuvoje gerai išvystytos interneto infrastruktūros, CDN realizavimas neturi prasmės.

2.12.4 Tarpiniai (proxy) serveriai

Proxy serveris yra tarpinis serveris tarp asmeninio kompiuterio ir svetainės, kurią norima pasiekti, serverio. Proxy serveriai yra kelių tipų:

- Visiškai anoniminiai („elite“ arba „high anonymous“). Taip sukonfigūruoti proxy serveriai ne tik paslepia naudotojo IP adresą, bet ir neperduoda parametrų, iš kurių galima nustatyti, kad tai proxy serveris. Svetainių administratoriams jie atrodo kaip eiliniai naudotojai (nebent jiems yra žinomi tų proxy serverių IP ar žiniatinklio adresai).

- Anoniminiai („anonymous“) - jie taip pat paslepia naudotojo tikrą IP adresą, bet naršant per juos yra paliekami proxy serveriams būdingi pėdsakai, tad svetainių administratoriams lengva nustatyti, kad šis lankytojas kreipiasi naudodamas proxy. Kai kurie administratoriai blokuoja prieigą per proxy.

- Skaidrūs („transparent“) - jie nėra anonimiški: ne tik palieka proxy pėdsakus, bet ir perduoda tikrą IP adresą. Tokie proxy serveriai netinka saugiam anonimiam naršymui. Jie paprastai naudojami tik naršymui pagreitinti.

Pagal funkcionalumą ir paskirtį proxy serveriai skirstomi į :

- HTTP proxy - šie proxy serveriai perduoda tik http protokolo duomenis, tad tinka tik žiniatinklio tinklalapiams naršyti.

- Socks proxy - tai proxy perduodantys ne tik http protokolo duomenis, bet ir kitų protokolų. Šie proxy gali būti jungiami į proxy serverių grandinę. Socks proxy dažniausiai nėra prieinami viešam naudojimui.

- WEB proxy - šie proxy serveriai veikia per interneto tinklalapį. Žiniatinklio serveryje paleidžiamas tam tikras php arba asp skriptas, kuris ir atlieka proxy serverio vaidmenį. Tokie proxy turi daug trūkumų, dažniausiai neleidžia perduoti formų duomenų, blokuoja javascript skriptus, nesupranta rėmelių, atidaro papildomus langus ir meta reklaminius „banerius“.

2.12.5 Klasteriai

Serverių apjungimas į klasterį – technika, didelio naudotojų srauto aptarnavimą padalinant paraleliai dirbantiems serveriams. Šis metodas padidina bendrą sistemos našumą bei patikimumą.

2.13 DUOMENŲ BAZIŲ VALDYMO SISTEMA (DBVS)

Duomenų bazių valdymo sistema (DBVS) – kompiuterinė programa ar programų paketas, skirtas duomenų bazės valdymui. DBVS skirtos valdyti milžiniškus struktūrizuotų duomenų kiekius bei vienu

metu aptarnauti didelius kiekius lygiagrečiai dirbančių naudotojų. VMA „Moodle“ mokymosi medžiaga, grafinė informacija, studentų įkelti darbai laikomi failų pavidalu „Moodle“ data kataloge. Kitą informaciją „Moodle“ talpina DBVS. Todėl DBVS našumas bei greitaveika ypatingai svarbūs „Moodle“ darbo greičiui.

MySQL arba PostgreSQL rekomenduojamos „Moodle“ naudojimui duomenų bazės. Jų darbas su „Moodle“ visapusiškai išbandytas ir dokumentuotas. MSSQL DBVS tinkamas „Moodle“ funkcionavimo užtikrinimui, bet papildomi VMA „Moodle“ plėtiniai gali būti su šita duomenų baze neišbandyti. Taip pat dokumentacija ar pagalba internetiniuose forumuose nėra tokia gausi ir išsami, kaip MySQL/PostgreSQL. Oracle DBVS nėra pilnai palaikoma ir nerekomenduojama jos naudojimas su „Moodle“. MySQL DBVS naudojimas su VMA „Moodle“ labiau dokumentuotas, tuo tarpu PostgreSQL pasižymi geresniais darbo stabilumo rezultatais. „Moodle“ darbui su DBVS reikės atitinkamo PHP plėtinio jungčiai su pasirinkta DBVS (moodle.org, 2015).

DBVS negali veikti optimaliai esant tipiniams konfigūravimo parametrams. DBVS konfigūravimo parametrai derinami tiek prie techninės įrangos galimybių, tiek prie operuojamų duomenų specifikos. Todėl DBVS priderinimas prie „Moodle“ poreikių žada ženklų teigiamą poveikį „Moodle“ našumui bei greitaveikai.

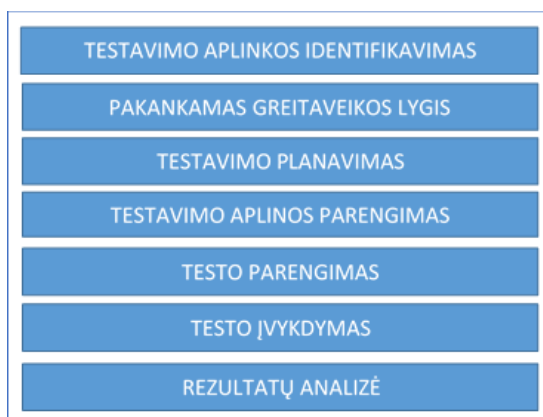
3. NAŠUMO BEI GREITAVEIKOS VERTINIMO METODIKA BEI ĮRANKIAI

Našumo testavimas naudojamas aptikti našumo problemas, susijusias su serverių techninės, programinės ir komunikacijos įrangos resursų trūkumu (procesorių, atminties, interneto kanalo pralaidumo ir kt.). Testų metu siekiama įvertinti sukurtos WWW sistemos pajėgumus, nustatyti, kada sistema dirba efektyviausiai bei kada jos veikimas pasiekia kritinę ribą. Vykdomi du testavimo etapai: apkrovimo (angl. *load*) testavimas ir įtampos (angl. *stress*) testavimas:

- apkrovimo testavimas – stebimas sistemos darbas esant maksimaliam apkrovimui, sistemos galimybių įvertinimas;
- įtampos testavimas – sistema sąmoningai perkraunama siekiant išsiaiškinti jos funkcionalumą esant kritinei situacijai.

3.1 TESTAVIMO METODIKA

Pirminis sistemos greitaveikos vertinimo tikslas surinkus statistinę informaciją, iliustruojančią bendrą žiniatinklio programos kokybinį lygį – sistemos tinkamumą atlikti paskirtą uždavinį (Microsoft Corporation, 2007). Papildomai žiniatinklio sistemos greitaveikos vertinimo testai atskleidžia sistemos probleminius taškus „butelio kakliukas“, gelbsti atliekant sistemos derinimo darbus, parodo sistemos atitikimą suplanuotiems našumo kriterijams. Sistemos greitaveikos testavimas gali nurodyti reikalavimus techninei įrangai sistemos darbinei aplinkai. Šiame darbe greitaveiką vertinama atliekant 7 žingsnius (10 pav.). Toliau trumpai pateikiama, kokie veiksmai atliekami kiekviename etape.



10 pav. Greitaveikos vertinimo etapai

3.1.1 Testavimo aplinkos identifikavimas

Fizinės terpės testavimo bei reikalavimų gamybinei aplinkai realizuoti identifikavimas (Microsoft, 2010). Įrankių bei išteklių, galimų naudoti testo atlikimui, identifikavimas. Fizinės terpės techninių,

programinių, tinklinių parametrų įvertinimas. Pilnas testavimo aplinkos įvertinimas lemia testavimo efektyvumą bei rezultatų tikrumą.

3.1.2 Pakankamos greitaveikos lygio identifikavimas

Užklauso atsakymo laiko, bendro pralaidumo, resursų apkrovos bei kitų sistemos greitaveikos parametrų, atitinkančių pakankamą lygį sistemai funkcionuoti ir atlikti jai paskirtas užduotis, identifikavimas. Papildomai galima identifikuoti greitaveikos pokyčius taikant vienokią ar kitokią sistemos konfigūraciją.

3.1.3 Testų planavimas

Testavimo procesas turi būti planuojamas, o planas turėtų nusakyti testavimo proceso tikslus, resursus, testavimo projektą, testavimo scenarijus, testavimo laikus ir užduotis. Testavimo planas turi atitikti užsibrėžtą kokybę. Testavimo duomenų parengimas. Testavimo procesas turi būti matuojamas ir renkamos matavimų rezultatai.

3.1.4 Testavimo aplinkos konfigūravimas

Testavimo aplinkos, įrankių bei resursų parengimas testavimo plano atitikimui. Testavimo aplinkos stebėsenos parengimas.

3.1.5 Testo parengimas

Pasiruošimas testo atlikimui. Testavimo aplinkos parengimas testo atlikimui. Priemonių paruošimas aktyvuojant juos parametrais atitinkančiais testavimo planą.

3.1.6 Testo įvykdymas

Testo vykdymas bei sistemos stebėseną. Testo eigos bei rezultatų įvertinimas.

3.1.7 Rezultatų analizė, ataskaitos

Rezultatų analizė bei perkėlimas į ataskaitą. Testavimo plano pokyčiai atsižvelgiant į testavimo rezultatus. Pakartotinis testavimas. Testavimas baigiamas nustatčius metrikų atitikimą tikėtiniems dydžių rėžiams, sistemos stebėsenos parametrų neviršinius kritinių reikšmių, išsaugojus dominančius testavimo rezultatus.

3.2 GREITAVEIKOS VERTINIMO ĮRANKIAI

AB ir Siege įrankiai priklauso WWW serverių greitaveikos vertinimo įrankių grupei. Jų paskirtis - pateikti statistinę informaciją apie WWW serverio HTTP atsakymų į įvairias HTTP užklausas trukmę. Šie įrankiai leidžia simuliuoti įvairaus kiekio naudotojų užklausas į konkretų WWW dokumentą. Svarbiausia

įrankių savybė – galimybė simuliuoti kelių naudotojų, paraleliai pateikiančių užklausas serveriui, darbą.

Kiekvienas įrankis gali pateikti informaciją apie:

- laiką sunaudota užklausiai įvykdyti;
- serverio atsakymo dydį;
- kiekį užklausų kurias serveri gali aptarnauti per sekundę;

AB ir Siege greitaveikos įvertinimo įrankiai pasirinkti dėl:

- paprastumo naudoti – turi tik keletą pasirenkamų parametrų;
- diegimo paprastumo – abu lengvai ir greitai įdiejami.

3.2.1 AB

Apache Benchmark (AB) vienas iš žinomiausių žiniatinklio (WWW) serverių greitaveikos testavimo priemonių. Šis įrankis įtrauktas į žiniatinklio serverio Apache paketą. Pagrindinis AB tikslas – pateikti informaciją apie nustatyto užklausų skaičiaus į konkrečią URL atsakymų parametrus:

- bendras parsųstų duomenų kiekis;
- bendras skaičius per sekundę aptarnautų užklausų;
- maksimalus užklausos aptarnavimo laikas;
- minimalus užklausos aptarnavimo laikas.

AB veikia nepriklausomai nuo Apache žiniatinklio serverio, todėl gali būti naudojamas daugelio žiniatinklio serverių greitaveikos testavimui:

```
ab -c 20 -n 1000 http://vma.elinara.lt/

Server Software: Apache/2.2.15
Server Hostname: vma.elinara.lt
Server Port: 80
Document Path: /
Document Length: 18742 bytes
Concurrency Level: 20
Time taken for tests: 11.482 seconds
Complete requests: 1000
Failed requests: 0
Write errors: 0
Total transferred: 19432788 bytes
HTML transferred: 18779484 bytes
Requests per second: 87.09 [#/sec] (mean)
Time per request: 229.639 [ms] (mean)
Time per request: 11.482 [ms] (mean, across all concurrent requests)
Transfer rate: 1652.80 [Kbytes/sec] received
```

```
Connection Times (ms)
  min mean[+/-sd] median max
Connect: 0 0 0.2 0 4
Processing: 103 228 39.3 229 408
Waiting: 54 143 29.7 143 246
Total: 104 228 39.3 229 409
```

3.2.2 Siege

Siege yra regresijos testas ir greitaveikos testavimo įrankis. Jis gali testuoti vieną URL su naudotojo nustatytu skaičiumi imituojamų naudotojų, arba jis gali WWW serverį testuoti keletu URL atsitiktinai laike parinkdamas URL, taip imituodamas natūralią serverio apkrovą. Programa ataskaitoje pateikia bendrą užklausų skaičių, atsakymo laiką, lygiagrečių naudotojų skaičių. Siege palaiko HTTP/1.0 ir 1,1 protokolus, GET ir POST užklausas, sausainėlius, naudotojų autentifikavimą. Siege funkcijos prieinamos neprivilegiuotam naudotojui:

```
siege -c 20 -b -t 20s -q http://vma.elinara.lt/

Transactions: 1857 hits
Availability: 100.00 %
Elapsed time: 19.97 secs
Data transferred: 9.66 MB
Response time: 0.21 secs
Transaction rate: 92.99 trans/sec
Throughput: 0.48 MB/sec
Concurrency: 19.90
Successful transactions: 1857
Failed transactions: 0
Longest transaction: 0.36
Shortest transaction: 0.12
```

4. NAŠUMĄ BEI GREITAVEIKĄ LEMIANČIŲ VEIKSNIŲ TYRIMAS

Numatyta atlikti eksperimentinį tyrimą ir įvertinti techninių bei programinių sisteminės aplinkos komponentų įtaką „Moodle“ našumui bei greitaveikai. Tam tikslui buvo atlikta serija eksperimentų, pasitelkiant skirtingus sisteminės aplinkos komponentus bei jų konfigūravimo parametrus.

4.1 EKSPERIMENTŲ ATLIKIMO TVARKA BEI TESTAVIMO METODIKA

Eksperimentai atliekami sukuriant nustatytų parametrų testavimo aplinką. Testavimo aplinka - eksperimentui būtinas techninių bei programinių priemonių rinkinys, pasižymintis savybėmis, kurių įtaką „Moodle“ našumui bei greitaveikai norima iširti. Siekiant išvengti buvusių eksperimentų konfigūracijų ar duomenų įtakos, kiekvienam eksperimentui kuriama nauja testavimo aplinka. Siekiama tą patį eksperimentą pakartoti skirtingomis „Moodle“ versijomis, talpinant į „Moodle“ skirtingų apimčių mokymosi kursus (1 lentelė). Eksperimentams atlikti buvo naudojama tuo metu naujausia „Moodle“ versija 2.8 bei paskutinė LTS tipo „Moodle“ versija. „Moodle“ versijų įtakos eksperimentuose naudojamos naujausios „Moodle“ 2 šeimos subversijos.

1 lentelė. Skirtingų apimčių mokymosi kursai

	Kursas XS	Kursas M	Kursas L
Vartotojai	1	1000	10000
Užduotys	1	100	500
Puslapiai	1	200	1000
Maži failai	1	128	1024
Dideli failai	1	5	10
Žinutes forume	2	500	5000

„Moodle“ našumas bei greitaveika matuojama *Apache HTTP server benchmarking tool* arba *Apache JMeter* programinės įrangos pagalba. Greitaveikos pokyčiams nustatyti pagrinde naudojama 60-ties lygiagrečiai dirbančių naudotojų imitacija. Šis parametras nustatytas empiriniais metodais, vadovaujantis tipiniais mokymosi aplinkų naudojimo scenarijais, „Moodle“ aplinkų naudojimo statistika bei išankstiniais tipinės „Moodle“ testavimo aplinkos testavimų rezultatais. Pagrindinis tyrimo objektas neautentifikuoto naudotojo užklausa į titulinį „Moodle“ aplinkos puslapį trukmė.

Kiekvienas eksperimentas atliekamas tik vieno sisteminės aplinkos parametro įtakai, „Moodle“ našumui bei greitaveikai identifikuoti.

4.2 TESTAVIMO APLINKA BEI JOS KONFIGŪRACIJA

Eksperimentuose naudojami virtualieji serveriai. Virtualiųjų serverių techninės įrangos konfigūracija yra tipinė. Eksperimentuose naudojami techninės įrangos deriniai sudaryti iš 8 procesoriaus branduolių, 8 Gb operatyviosios atminties virtualiųjų serverių (išskyrus techninės įrangos įtakos eksperimentus).

Virtualizacijos lygmuo realizuotas naudojant VMware ESXi hipervizorius. Kiekvienas eksperimente naudojamas techninės įrangos derinys startuoja atskirame fiziniame serveryje. Siekiama, kad techninės įrangos komponentai dirbtų optimaliu režimu, jų darbo neįtakotų lygiagrečiai dirbančios sistemos. Testuojamų procesų darbo neįtakotų operatyviosios ar diskinės atminties stoka, tinklo pralaidumas.

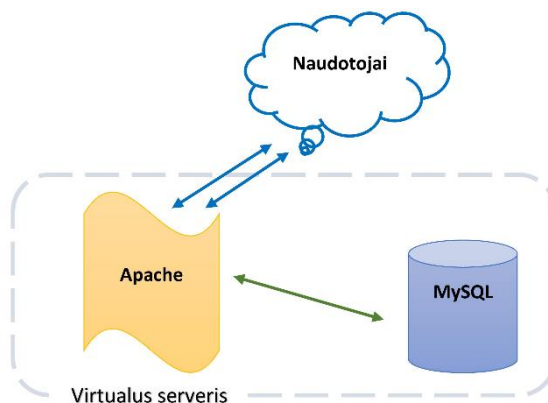
4.3 TECHNINĖ ĮRANGA

Eksperimento tikslas - nustatyti techninės serverių įrangos įtaką „Moodle“ našumui bei greitaveikai. Testai buvo atliekami keičiant techninės įrangos parametrus: procesoriaus branduolių skaičių, operatyviosios atminties kiekį, diskinę posistemę.

4.3.1 Procesoriaus bei operatyviosios atminties įtaka

Eksperimentas skirtas serverio procesoriaus bei operatyviosios atminties įtakai „Moodle“ našumui nustatyti. Eksperimentas atliktas pasitelkus 6-is skirtingus virtualiojo serverio konfigūravimo variantus su skirtingais vCPU bei RAM kiekiais.

Programinės įrangos komponentai (žiniatinklio serveris, DBVS) patalpinti viename virtualiajame serveryje (11 pav. ir 2 lentelė).



11 pav. Eksperimento komponentų schema

2 lentelė. Testavimo aplinkos konfigūracija

Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualusis serveris:	„Moodle“ serveris (vCPU kiekis: kintamas, RAM kiekis: kintamas)
Programinė įranga:	Centos 7, Apache 2.4.6, PHP 5.6.4, MySQL 5.6.21

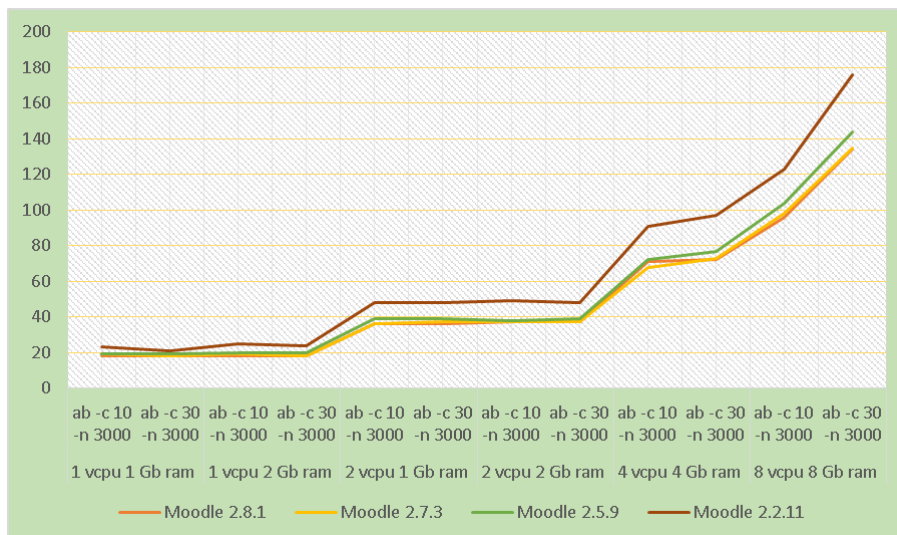
Apache WWW serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas realizuotas OPCache plėtinio pagalba.

Testuojama 2.7 ir 2.8 „Moodle“ versijų greitaveika, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų bei su XS, M ir L dydžio mokymosi kursais (žr. 1 lentelę). Greitaveikos matavimai atlikti AB programos pagalba, imituojant 10 arba 30 lygiagrečių naudotojų. Gauti eksperimento rezultatai pateikiami 3 lentelėje ir 12 pav.

3 lentelė. vCPU ir RAM kiekio didinimo įtaka

Techninės įrangos įtaka	1 vcpu 1 Gb ram		1 vcpu 2 Gb ram		2 vcpu 1 Gb ram	
	ab -c 10 -n 3000	ab -c 30 -n 3000	ab -c 10 -n 3000	ab -c 30 -n 3000	ab -c 10 -n 3000	ab -c 30 -n 3000
Moodle 2.8.1	18	18	18	18	36	36
Moodle 2.7.3	19	18	19	18	36	37
Moodle 2.5.9	19	19	20	20	39	39
Moodle 2.2.11	23	21	25	24	48	48

Techninės įrangos įtaka	2 vcpu 2 Gb ram		4 vcpu 4 Gb ram		8 vcpu 8 Gb ram	
	ab -c 10 -n 3000	ab -c 30 -n 3000	ab -c 10 -n 3000	ab -c 30 -n 3000	ab -c 10 -n 3000	ab -c 30 -n 3000
Moodle 2.8.1	37	37	71	72	96	134
Moodle 2.7.3	37	37	68	73	98	135
Moodle 2.5.9	38	39	72	77	104	144
Moodle 2.2.11	49	48	91	97	123	176



12 pav. vCPU ir RAM kiekio didinimo įtaka

„Moodle“ greitaveika priklauso nuo serverio procesoriaus greičio ir procesoriaus branduolių kiekio. Operatyvioji atmintis, esant jos pakankamam kiekiui, žymaus poveikio greitaveikai nedaro.

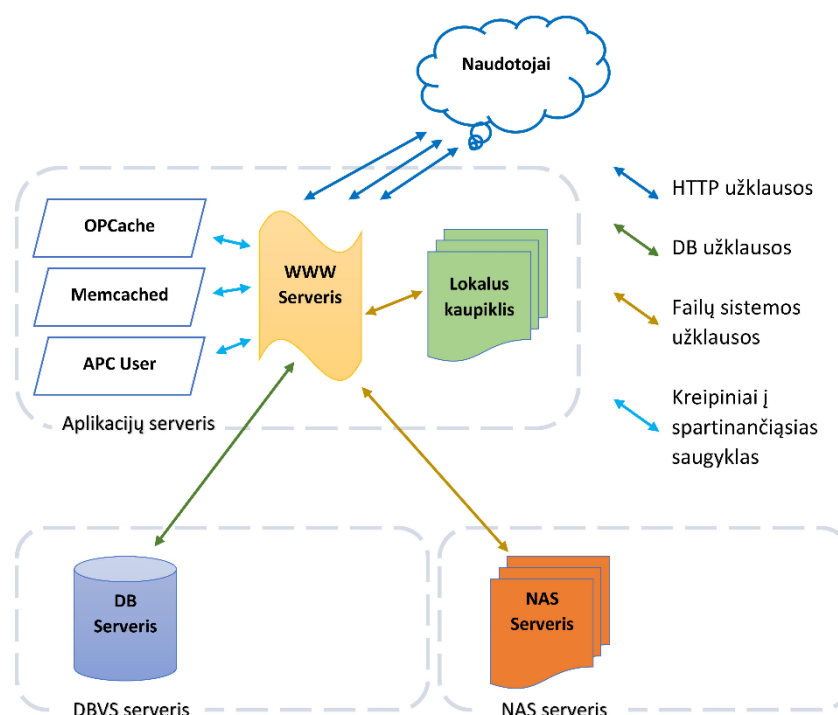
4.3.2 Diskinės posistemės bei spartinančiųjų saugyklų įtaka

Eksperimentas skirtas serverio diskinės sistemos bei spartinančiųjų saugyklų įtakai „Moodle“ našumui bei greitaveikai nustatyti. Eksperimentas atliktas pasitelkus 6-is skirtingas virtualiojo serverio diskinės posistemės bei spartinančiųjų saugyklų konfigūravimo variantus:

- HDD tipo kaupiklis;
- SSD tipo kaupiklis;

- c) SSD tipo kaupiklis su įgalinta *APC user* tipo spartinančiąja saugykla. Sesijų identifikatoriai saugomi pagrindinėje spartinančiojoje sesijų saugykloje;
- d) NFS protokolu prijungta NAS duomenų saugykla;
- e) NFS protokolu prijungta NAS duomenų saugykla su įgalinta *Memcached* tipo spartinančiąja saugykla. Sesijų identifikatoriai saugomi duomenų bazių serveryje;
- f) NFS protokolu prijungta NAS duomenų saugykla su įgalinta *APC user* tipo spartinančiąja saugykla. Sesijų identifikatoriai saugomi duomenų bazių serveryje.

WWW serveris bei DBVS talpinami skirtinguose virtualiuose serveriuose (žr. 13 pav. ir 4 lentelė).



13 pav. Eksperimento komponentų schema

4 lentelė. Testavimo aplinkos konfigūracija

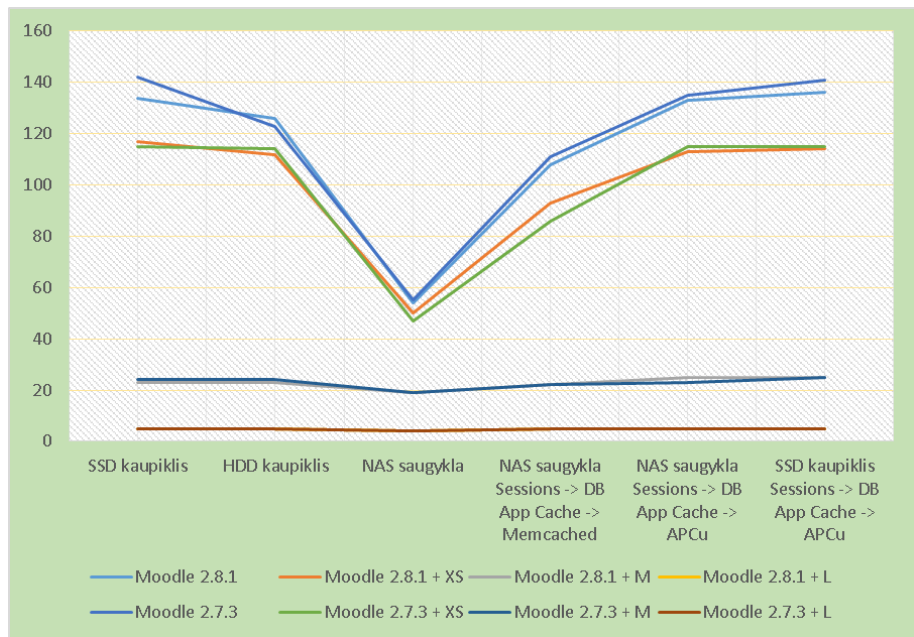
Techninė įranga:	Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz, 32Gb RAM
Virtualieji serveriai:	Programų serveris (vCPU kiekis: 4, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 4, RAM kiekis: 8 Gb) NAS domėnų saugykla (vCPU kiekis: 4, RAM kiekis: 8 Gb)
Programinė įranga:	Programų serveris (Centos 7, Apache 2.4.6, PHP 5.6.4) DBVS serveris (Centos 7, MySQL 5.6.21) NAS domėnų saugykla (FreeNAS 9.3(FreeBSD))

Apache žiniatinklio (WWW) serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas įgyvendintas OPCache plėtinio pagalba. NAS saugykla įgyvendinta FreeNAS 9.3 programinės įrangos pagalba. FreeNAS dirba FreeBSD operacinės sistemos bazėje. Memcached serveris įgalintas WWW serveryje. APC PHP plėtinys įgalintas WWW serveryje.

Testuojama 2.7 ir 2.8 „Moodle“ versijų greitaveika, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų bei su XS, M ir L dydžio mokymosi kursais (žr. 1 lentelę). Greitaveikos matavimai atlikti AB programos pagalba, imituojant 60 lygiagrečių naudotojų darbą 60 sekundžių laikotarpyje. Gauti eksperimento rezultatai pateikiami 5 lentelėje ir 14 pav.

5 lentelė. Diskinės posistemės bei spartinančiųjų saugyklų įtaka

Serverio diskinės sistemės, bei spartinančiųjų saugyklų įtaka	SSD kaupiklis	HDD kaupiklis	NAS saugykla	NAS saugykla Sessions -> DB App Cache -> Memcached	NAS saugykla Sessions -> DB App Cache -> APCu	SSD kaupiklis Sessions -> DB App Cache -> APCu
Moodle 2.8.1	134	126	54	108	133	136
Moodle 2.8.1 + XS	117	112	50	93	113	114
Moodle 2.8.1 + M	23	23	19	22	25	25
Moodle 2.8.1 + L	5	5	4	5	5	5
Moodle 2.7.3	142	123	55	111	135	141
Moodle 2.7.3 + XS	115	114	47	86	115	115
Moodle 2.7.3 + M	24	24	19	22	23	25
Moodle 2.7.3 + L	5	5	4	5	5	5



14 pav. Diskinės posistemės bei spartinančiųjų saugyklų įtaka

„Moodle“ „Moodle“ data kataloge talpina sesijų, programų bei užklausų tarpines saugyklas. Todėl perkėlus „Moodle“ data į NAS tipo saugyklą, pasižyminčia ženkliai didesnės trukmės užklausomis, krenta „Moodle“ našumas bei greitaveika. Lėtos failų sistemos įtaka „Moodle“ našumui bei greitaveikai gali būti minimizuota naudojant papildomus spartinančiosios saugyklos elementus, nesusijusius su failų sistema. Tam tikslui, gali būti naudojama Memcached saugyklos serveris, Alternative PHP cache su išjungtu

bytekodo spartinimu arba *MongoDB* ne SQL duomenų bazių serveris. Sesijų identifikatorių saugojimui galima naudoti pagrindinę DBVS.

Sesijų identifikatorius saugant DBVS 50% - 70% padidėja pastarojo apkrovimas. Perkėlus „Moodle“ „Application cache“ saugyklą į Memcached arba APC tarpinės atminties saugykla NAS serverio CPU apkrova krinta iki 0,5% - 3%.

Kadangi PHP bytekodas spartinamas *OPCache* spartintojo pagalba *Alternative PHP cache (APC)* naudojamas tik „Moodle“ programos duomenų spartinimui. Tai įgyvendinta naudojant *APCu - APC User Cache* PHP modulį.

4.4 OPERACINĖ SISTEMA

Eksperimentas skirtas serverio operacinės sistemos greitaveikos įtakai „Moodle“ našumui bei greitaveikai nustatyti. Eksperimentas atliktas pasitelkus 4-is virtualiojo serverio programinės įrangos konfigūravimo variantus:

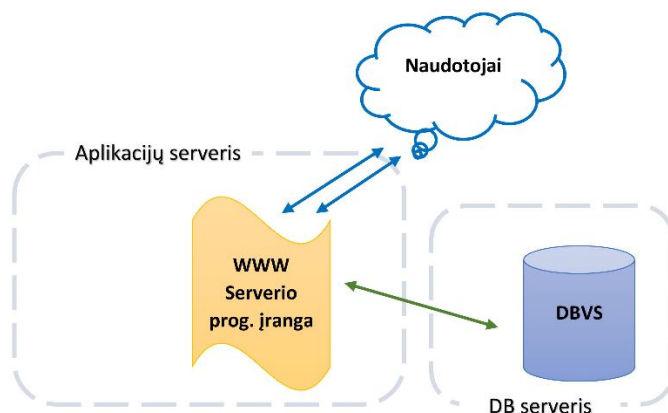
- a) Programų serveris (Centos 7, Apache 2.4.6, PHP 5.6.4)
DBVS serveris (Centos 7, MySQL 5.6.21);
- b) Programų serveris (Centos 7, Nginx 1.6.2, PHP 5.6.4)
DBVS serveris (Centos 7, MySQL 5.6.21);
- c) Programų serveris („Windows“ Server 2008 R2, Apache 2.4.10, PHP 5.6.3)
DBVS serveris (Centos 7, MySQL 5.6.21);
- d) Programų serveris („Windows“ Server 2008 R2, IIS 7, PHP 5.6.3)
DBVS serveris (Centos 7, MySQL 5.6.21).

WWW serveris bei DBVS talpinami skirtinguose virtualiuose serveriuose. Eksperimente naudoti komponentai pateikiami 15 pav. ir 6 lentelėje.

6 lentelė. Testavimo aplinkos konfigūracija

Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualieji serveriai:	Programų serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb)
Programinė įranga:	a) Programų serveris (Centos 7, Apache 2.4.6, PHP 5.6.4) DBVS serveris (Centos 7, MySQL 5.6.21) b) Programų serveris (Centos 7, Nginx 1.6.2, PHP 5.6.4) DBVS serveris (Centos 7, MySQL 5.6.21) c) Programų serveris („Windows“ Server 2008 R2, Apache 2.4.10, PHP 5.6.3)

	DBVS serveris (Centos 7, MySQL 5.6.21)
d)	Programų serveris („Windows“ Server 2008 R2, IIS 7, PHP 5.6.3)
	DBVS serveris (Centos 7, MySQL 5.6.21)



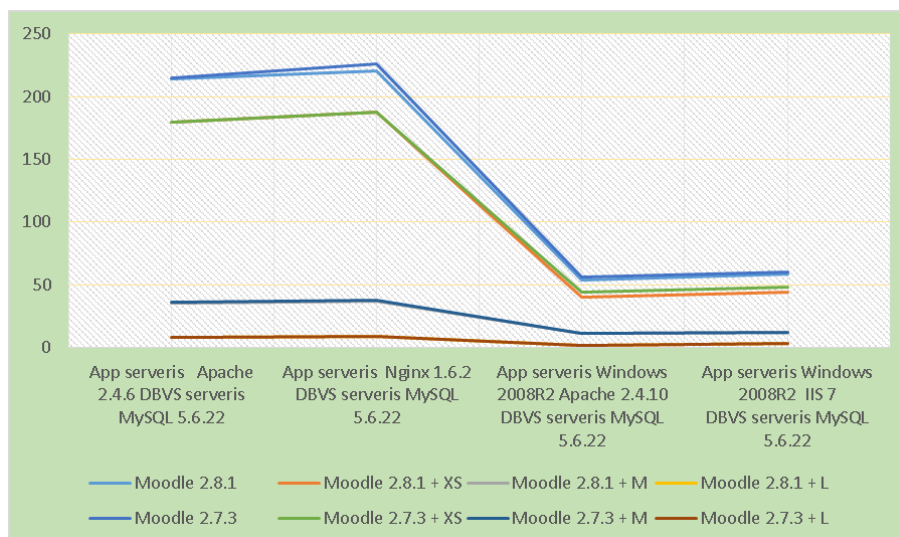
15 pav. Eksperimento komponentų schema

Apache WWW serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas realizuotas OPCache plėtinio pagalba. IIS WWW serveris PHP interpretatorių išskviečia FastCGI metodu. Testuojama 2.7 ir 2.8 „Moodle“ versijų greitaveiką, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų bei su XS, M ir L dydžio mokymosi kursais (žr. 1 lentelę). Greitaveikos matavimai atlikti AB programos pagalba, imituojant 60 lygiagrečių naudotojų darbą 60s laikotarpyje.

Gauti eksperimento rezultatai pateikiami 7 lentelėje ir 16 pav.

7 lentelė. Operacinės sistemos įtaka

Operacinės sistemos įtaka	App serveris Apache 2.4.6 DBVS serveris MySQL 5.6.22	App serveris Nginx 1.6.2 DBVS serveris MySQL 5.6.22	App serveris Windows 2008R2 Apache 2.4.10 DBVS serveris MySQL 5.6.22	App serveris Windows 2008R2 IIS 7 DBVS serveris MySQL 5.6.22
Moodle 2.8.1	214	221	54	59
Moodle 2.8.1 + XS	180	188	40	44
Moodle 2.8.1 + M	35	37	11	12
Moodle 2.8.1 + L	8	9	2	3
Moodle 2.7.3	215	226	56	60
Moodle 2.7.3 + XS	180	188	44	48
Moodle 2.7.3 + M	36	38	11	12
Moodle 2.7.3 + L	8	9	2	3



16 pav. Operacinės sistemos įtaka

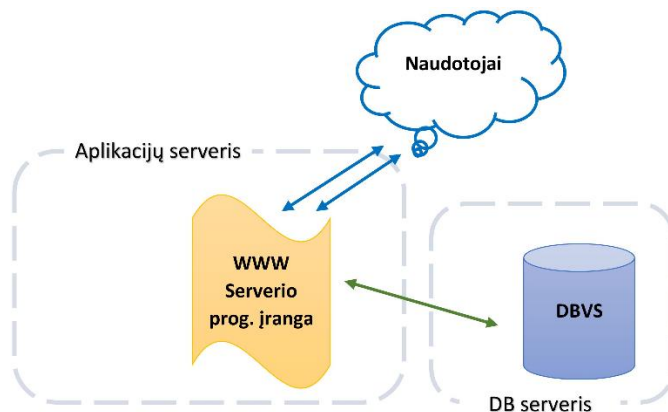
„Moodle“ greitaveika priklauso nuo serverio operacinės sistemos. Eksperimente stebėtas „Moodle“ greitaveikos skirtumas, lyginant Centos 7 ir „Windows“ Server 2008 R2 operacines sistemas, yra didelis. Todėl galima daryti išvadą, kad „Windows“ atveju, dalis programinės įrangos komponentų neveikė optimaliu režimu. Norint tiksliai įvertinti OS aspektą reikalingas papildomas „Windows“ sistemos įtakos „Moodle“ tyrimas.

4.5 ŽINIATINKLIO SERVERIS

Eksperimentas skirtas žiniatinklio (WWW) serverio programinės įrangos įtakai „Moodle“ našumui bei greitaveikai nustatyti. Eksperimentas atliktas pasitelkus 4-ias virtualiojo serverio programinės įrangos konfigūravimo variacijas:

- Programų serveris (Centos 7, Apache 2.4.6 prefork MPM (mod_php), PHP 5.6.4);
- Programų serveris (Centos 7, Apache 2.4.6 event MPM (php-fpm), PHP 5.6.4);
- Programų serveris (Centos 7, Nginx 1.6.2 (php-fpm), PHP 5.6.4);
- Programų serveris („Windows“ Server 2008 R2, IIS 7 (FastCGI), PHP 5.6.3).

Žiniatinklio serveris bei DBVS talpinami skirtinguose virtualiuose serveriuose. Eksperimente naudoti komponentai pateikiami 17 pav. ir 8 lentelėje.



17 pav. Eksperimento komponentų schema

8 lentelė. Testavimo aplinkos konfigūracija

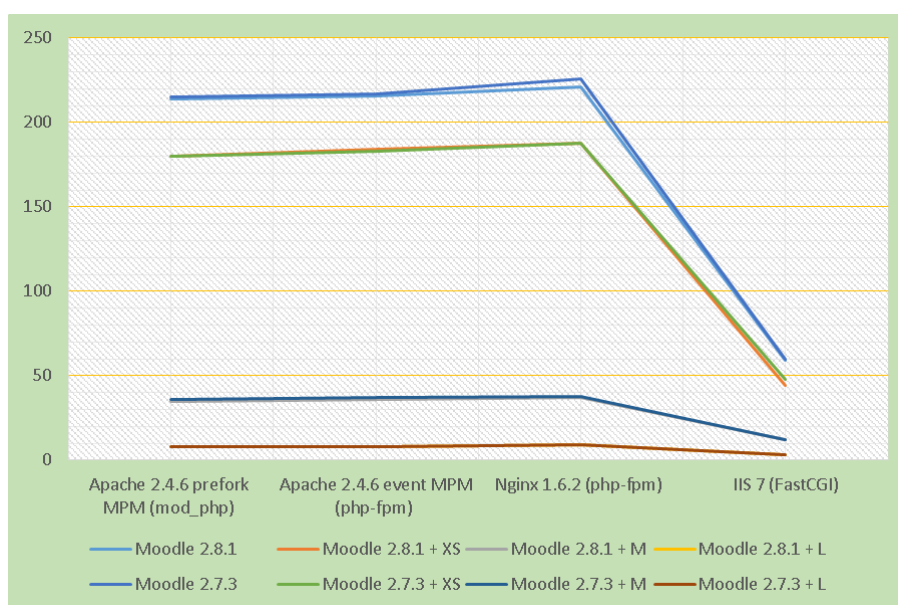
Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualieji serveriai:	Programų serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb)
Programinė įranga:	<p>a) Programų serveris (Centos 7, Apache 2.4.6 prefork MPM (mod_php), PHP 5.6.4) DBVS serveris (Centos 7, MySQL 5.6.21)</p> <p>b) Programų serveris (Centos 7, Apache 2.4.6 event MPM (php-fpm), PHP 5.6.4) DBVS serveris (Centos 7, MySQL 5.6.21)</p> <p>c) Programų serveris (Centos 7, Nginx 1.6.2 (php-fpm), PHP 5.6.4) DBVS serveris (Centos 7, MySQL 5.6.21)</p> <p>d) Programų serveris („Windows“ Server 2008 R2, IIS 7 (FastCGI), PHP 5.6.3) DBVS serveris (Centos 7, MySQL 5.6.21)</p>

Apache žiniatinklio serveris dirba su integruotu PHP interpretatoriumi (mod_php). Nginx serveris – su PHP interpretatoriumi ir susietas naudojant PHP-FPM procesų valdiklį. IIS žiniatinklio serveris PHP interpretatorių iškviečia FastCGI metodu. PHP kodo spartinimas realizuotas OPCache plėtinio pagalba. Testuojama 2.7 ir 2.8 „Moodle“ versijų greیتaveika, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų bei su XS, M ir L dydžio mokymosi kursais (žr. 1 lentelėje). Greitaveikos matavimai atlikti AB programos pagalba, imituojant 60 lygiagrečių naudotojų darbą 60s laikotarpyje.

Gauti eksperimento rezultatai pateikiami 9 lentelėje ir 18 pav.

9 lentelė. WWW serverio programinės įrangos įtaka

WWW serverio programinės įrangos įtaka	Apache 2.4.6 prefork MPM (mod_php)	Apache 2.4.6 event MPM (php-fpm)	Nginx 1.6.2 (php-fpm)	IIS 7 (FastCGI)
Moodle 2.8.1	214	216	221	59
Moodle 2.8.1 + XS	180	184	188	44
Moodle 2.8.1 + M	35	36	37	12
Moodle 2.8.1 + L	8	8	9	3
Moodle 2.7.3	215	217	226	60
Moodle 2.7.3 + XS	180	183	188	48
Moodle 2.7.3 + M	36	37	38	12
Moodle 2.7.3 + L	8	8	9	3



18 pav. WWW serverio programinės įrangos įtaka

„Moodle“ greitaveika mažai priklauso nuo konkrečios WWW serverio programinės įrangos. IIS serverio greitaveikos rezultatus reikia papildomai tikrinti. Eksperimente stebėtas „Moodle“ greitaveikos skirtumas lyginant *Apache prefork MPM*, *Apache event MPM* ir *Nginx* minimalus. *Nginx* patogu naudoti esant ribotiems RAM ištekliams. Taip pat *Nginx* greičiau pateikia statinį turinį. Bet jei sistemos funkcionavimui reikalingas *.htaccess* konfigūravimo failas, *Nginx* naudoti netikslinga.

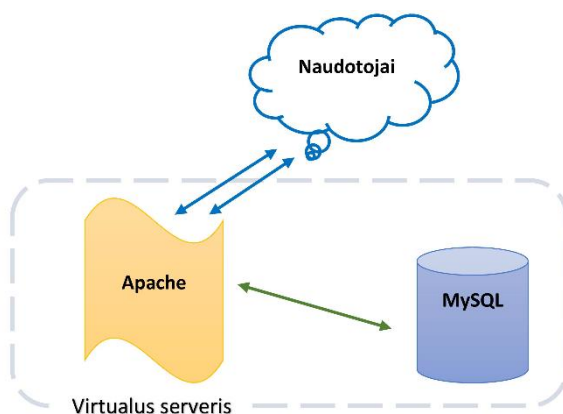
4.6 DUOMENŲ BAZIŲ VALDYMO SISTEMA

Eksperimentas skirtas duomenų bazių valdymo sistemų bei jų skirtingų konfigūracijų įtakai „Moodle“ našumui bei greitaveikai nustatyti. Eksperimentas atliktas pasitelkus 5-ias testavimo aplinkos konfigūravimo variacijas:

- „Moodle“ serveris: Centos 7, Apache 2.4.6, PHP 5.6.4, MySQL 5.6.22 (nustatymai nekeisti);

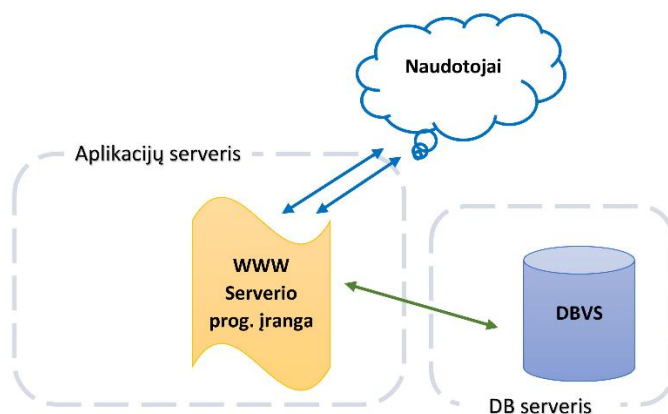
- b) „Moodle“ serveris: Centos 7, Apache 2.4.6, PHP 5.6.4, MySQL 5.6.22 (nustatymai modifikuoti naudojant *mysqltuner*);
- c) Programų serveris: (Centos 7, Apache 2.4.6, PHP 5.6.4)
DBVS serveris: Centos 7, MySQL 5.6.22 (nustatymai nekeisti);
- d) Programų serveris: (Centos 7, Apache 2.4.6, PHP 5.6.4)
DBVS serveris: Centos 7, MySQL 5.6.22 (nustatymai modifikuoti naudojant *mysqltuner*);
- e) Programų serveris: (Centos 7, Apache 2.4.6, PHP 5.6.4)
DBVS serveris: Centos 7, PostgreSQL 9.2.7 (nustatymai „Moodle“.org).

A ir B testavimo aplinkos variantuose žiniatinklio serveris bei DBVS talpinami tame pačiame serveryje (19 pav.).



19 pav. A ir B eksperimento komponentų schemas variantai

C, D ir E testavimo aplinkos variantuose DBVS serveris talpinamas atskirame virtualiame serveryje. A, B, C ir D naudojamas MySQL duomenų bazių serveris, E variante PostgreSQL. A ir C MySQL serverio konfigūravimo parametrai nekeisti, B ir D modifikuoti vadovaujantis, MySQL optimizavimo programinės įrangos – *mysqltuner*, pateiktomis rekomendacijomis. E testavimo aplinkos variante PostgreSQL konfigūracija modifikuota vadovaujantis moode.org forume pateiktais rekomenduojamais PostgreSQL konfigūracijos parametrais. Testavimo aplinkos konfigūracija pateikiama 19 ir 20 pav. bei 10 lentelėje.



20 pav. C, D ir E eksperimento komponentų schemas variantai

10 lentelė. Testavimo aplinkos konfigūracija

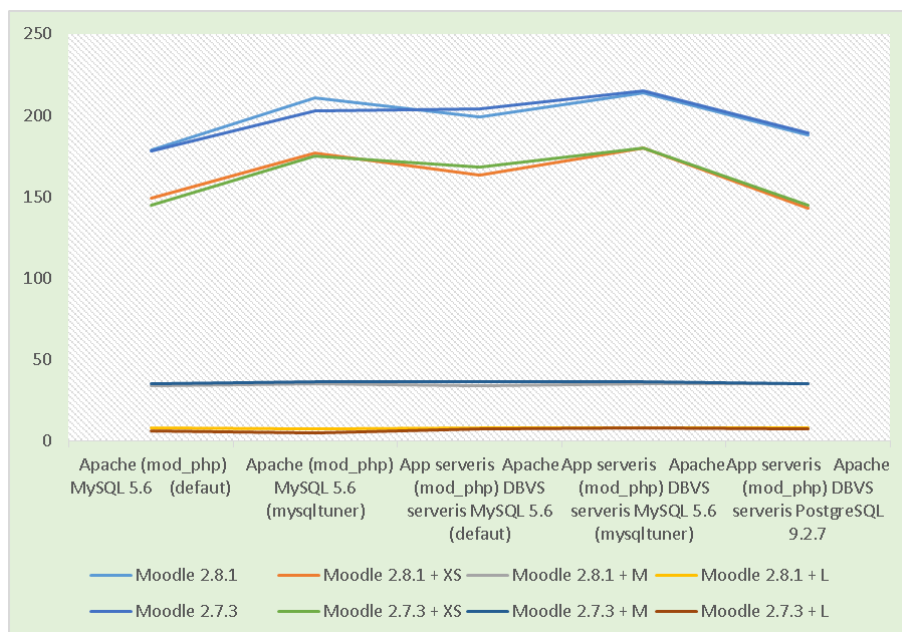
Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualieji serveriai:	„Moodle“ serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) Programų serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb)
Programinė įranga:	a) „Moodle“ serveris: Centos 7, Apache 2.4.6, PHP 5.6.4, MySQL 5.6.22 b) „Moodle“ serveris: Centos 7, Apache 2.4.6, PHP 5.6.4, MySQL 5.6.22 c) Programų serveris: (Centos 7, Apache 2.4.6, PHP 5.6.4) DBVS serveris: Centos 7, MySQL 5.6.22 d) Programų serveris: (Centos 7, Apache 2.4.6, PHP 5.6.4) DBVS serveris: Centos 7, MySQL 5.6.22 e) Programų serveris: (Centos 7, Apache 2.4.6, PHP 5.6.4) DBVS serveris: Centos 7, PostgreSQL 9.2.7

Apache žiniatinklio serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas realizuotas OPCache plėtinio pagalba. Testuojama 2.7 ir 2.8 „Moodle“ versijų greیتaveika, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų bei su XS, M ir L dydžio mokymosi kursais (žr. 1 lent). „Moodle“ greیتaveikos matavimai atlikti AB programos pagalba, imituojant 60 lygiagrečių naudotojų darbą 60s laikotarpyje.

Gauti eksperimento rezultatai pateikiami 11 lentelėje ir 21 pav.

11 lentelė. DBVS įtaka

DBVS įtaka	Apache (mod_php) MySQL 5.6 (default)	Apache (mod_php) MySQL 5.6 (mysqltuner)	App serveris Apache (mod_php) DBVS serveris MySQL 5.6 (default)	App serveris Apache (mod_php) DBVS serveris MySQL 5.6 (mysqltuner)	App serveris Apache (mod_php) DBVS serveris PostgreSQL 9.2.7
Moodle 2.8.1	179	211	199	214	188
Moodle 2.8.1 + XS	149	177	163	180	143
Moodle 2.8.1 + M	34	35	34	35	35
Moodle 2.8.1 + L	8	7	8	8	8
Moodle 2.7.3	178	203	204	215	189
Moodle 2.7.3 + XS	145	175	168	180	145
Moodle 2.7.3 + M	35	36	36	36	35
Moodle 2.7.3 + L	6	5	7	8	7



21 pav. DBVS įtaka

„Moodle“ greitaveika priklauso tiek nuo konkrečios DBVS serverio programinės įrangos, tiek nuo DBVS konfigūracijos parametrų. „Moodle“ serverį apkraunant DBVS užduotimis „Moodle“ sistemos našumas bei greitaveika mažėja. DBVS darbo optimizavimas, parinkus palankų konfigūracijos parametrų rinkinį, atlaisvina dalį CPU resursų ir didina „Moodle“ sistemos greitaveiką. DBVS perkėlimas į atskirą serverį sistemos našumą bei greitaveiką didina neženkliai. Tam turi įtakos pasidėjusios SQL užklausų trukmės.

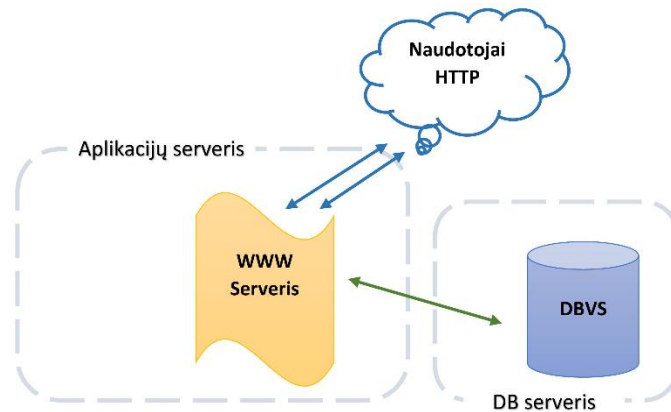
Pagrindinė išvada – dideliems naudotojų srautams aptarnauti reikia didelio max_connections skaičiaus.

4.7 SRAUTO PASKIRSTYMO PRIEMONIŲ BEI SSL ŠIFRAVIMO ĮTAKA

Eksperimentas skirtas srauto paskirstymo priemonių bei naudotojų užklausų šifravimo SSL protokolu įtakai „Moodle“ našumui bei greitaveikai nustatyti. Eksperimentas atliktas pasitelkus 4-is virtualiųjų serverių programinės įrangos konfigūravimo variacijas:

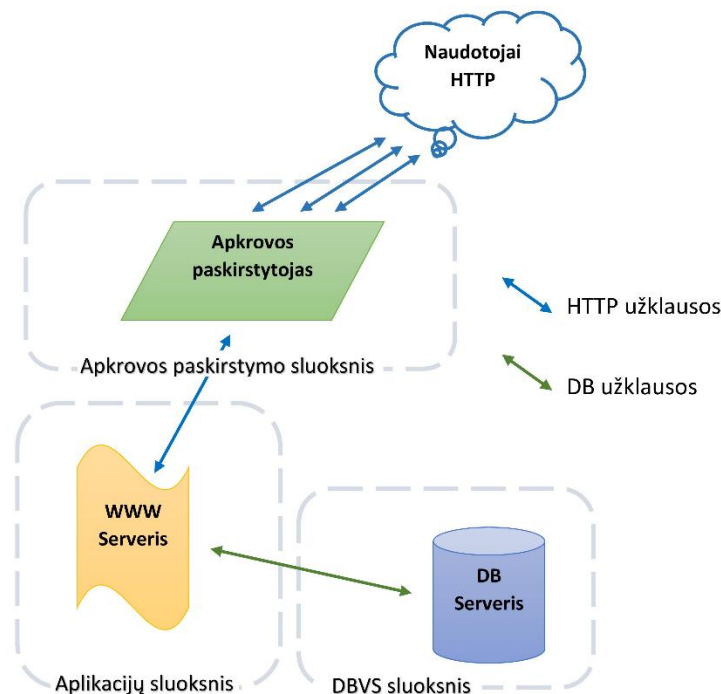
- Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6,4
DBVS serveris: Centos 7, MySQL 5.6.22;
- Srauto skirstytuvas: Centos 7, HAProxy 1.5.3
Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6,4
DBVS serveris: Centos 7, MySQL 5.6.22;
- Programų serveris: Centos 7, Apache 2.4.6 (mod_php, mod_ssl), PHP 5.6,4
DBVS serveris: Centos 7, MySQL 5.6.22
- SSL proxy: Centos 7, HAProxy 1.5.3;
Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6,4
DBVS serveris: Centos 7, MySQL 5.6.22.

Srauto skirstytuvas, SSL proxy serveris, žiniatinklio (WWW) serveris bei DBVS talpinami skirtinguose virtualiuose serveriuose. A testavimo aplinkos variantuose žiniatinklio serveris aptarnauja HTTP užklausas (22 pav.). B testavimo aplinkos variante HTTP užklausos žiniatinklio serverį pasiekia per srauto paskirstymo serverį (23 pav.). C testavimo aplinkos variantuose žiniatinklio serveryje įgalintas mod_ssl papildinys. Naudotojų užklausos serverį pasiekia HTTPS protokolu. D testavimo aplinkos variante HTTPS užklausas aptarnauja ir jų šifravimą vykdo SSL proxy serveris o žiniatinklio serveriui persiunčia jau nešifruotas HTTP užklausas.



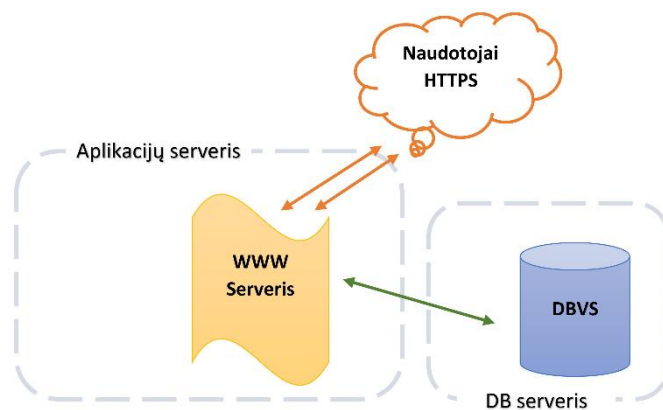
22 pav. A eksperimento komponentų schema

B testavimo aplinkos variante HTTP užklausos žiniatinklio serverį pasiekia per srauto paskirstymo serverį.



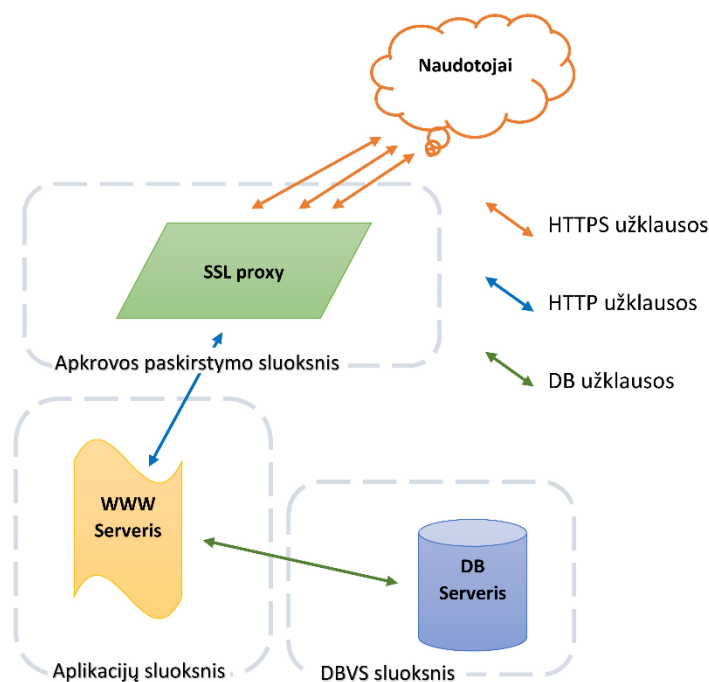
23 pav. B eksperimento komponentų schema

C testavimo aplinkos variantuose žiniatinklio serveryje įgalintas mod_ssl papildinys. Naudotojų užklausos serverį pasiekia HTTPS protokolu (24 pav.).



24 pav. C eksperimento komponentų schema

D testavimo aplinkos variante HTTPS užklausa aptarnauja ir jų šifravimą vykdo SSL proxy serveris o žiniatinklio serveriui persiūčia jau nešifruotas HTTP užklausa (25 pav.). Bendra testavimo komponentų konfigūracija pateikiama 12 lentelėje.



25 pav. D eksperimento komponentų schema

12 lentelė. Testavimo aplinkos konfigūracija

Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualieji serveriai:	Srauto skirstytuvas (vCPU kiekis: 4, RAM kiekis: 4 Gb) SSL proxy (vCPU kiekis: 4, RAM kiekis: 4 Gb) Programų serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb)

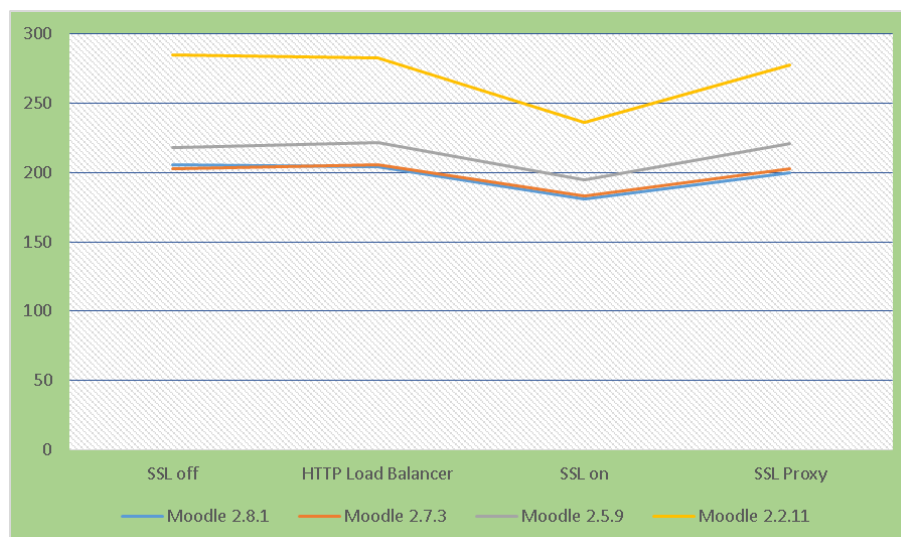
Programinė įranga:	a) Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6,4 DBVS serveris: Centos 7, MySQL 5.6.22
	b) Srauto skirstytuvai: Centos 7, HAProxy 1.5.3 Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6,4 DBVS serveris: Centos 7, MySQL 5.6.22
	c) Programų serveris: Centos 7, Apache 2.4.6 (mod_php, mod_ssl), PHP 5.6,4 DBVS serveris: Centos 7, MySQL 5.6.22
	d) SSL proxy: Centos 7, HAProxy 1.5.3 Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6,4 DBVS serveris: Centos 7, MySQL 5.6.22

Apache žiniatinklio serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas realizuotas OPCache plėtinio pagalba. Testuojama 2.2, 2.5, 2.7 ir 2.8 „Moodle“ versijų greitaveika, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų. „Moodle“ greitaveikos matavimai atlikti AB programa, imituojant 60 lygiagrečių naudotojų darbą 60s laikotarpyje.

Gauti eksperimento rezultatai pateikiami 13 lentelėje ir 26 pav.

13 lentelė. Srauto paskirstymo priemonių bei SSL šifravimo įtaka

SSL įtaka greitaveikai	SSL off	HTTP Load Balancer	SSL on	SSL Proxy
Moodle 2.8.1	206	204	181	200
Moodle 2.7.3	203	206	183	203
Moodle 2.5.9	218	222	195	221
Moodle 2.2.11	285	283	236	278



26 pav. Srauto paskirstymo priemonių bei SSL šifravimo įtaka

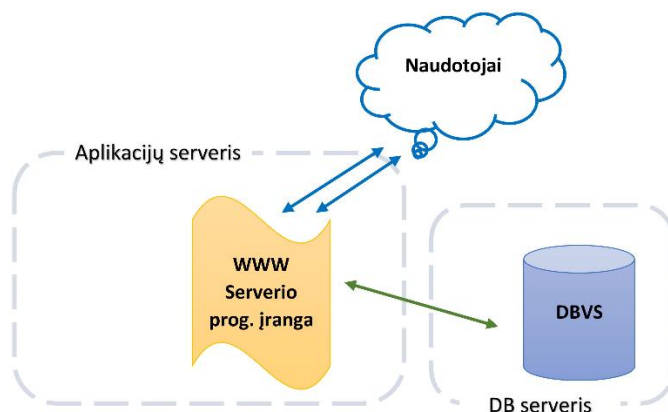
Srauto skirstymo priemonės (*load balancer*) „Moodle“ greitaveikos neįtakoja. Įgalinus užklausų šifravimą SSL protokolu, stebimas greitaveikos kritimas. Šifravimą SSL protokolu perkėlus iš žiniatinklio serverio į SSL proxy serverį greitaveika atsistato į buvusį lygį.

4.8 PHP PROGRAMŲ STEKAS

Eksperimentas skirtas PHP interpretatoriaus versijų įtakos „Moodle“ našumui bei greitaveikai nustatyti. Eksperimentas atliktas pasitelkus 3-is virtualiųjų serverių programinės įrangos konfigūravimo variacijas:

- Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.4
DBVS serveris: Centos 7, MySQL 5.6.22;
- Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.5
DBVS serveris: Centos 7, MySQL 5.6.22;
- Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6
DBVS serveris: Centos 7, MySQL 5.6.22.

Žiniatinklio (WWW) serveris bei DBVS talpinami skirtinguose virtualiuose serveriuose. Testavimo komponentai pateikiami 27 pav. ir 14 lentelėje.



27 pav. Eksperimento komponentų schema

14 lentelė. Testavimo aplinkos konfigūracija

Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualieji serveriai:	Programų serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb)
Programinė įranga:	<ol style="list-style-type: none"> Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.4 DBVS serveris: Centos 7, MySQL 5.6.22 Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.5 DBVS serveris: Centos 7, MySQL 5.6.22 Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6 DBVS serveris: Centos 7, MySQL 5.6.22

Apache žiniatinklio serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas įgyvendintas OPCache plėtiniu. Testuojama 2.2, 2.5, 2.7 ir 2.8 „Moodle“ versijų greitaveika, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų. „Moodle“ greitaveikos matavimai atlikti AB programa, imituojant 60 lygiagrečių naudotojų darbą 60s laikotarpyje.

Gauti eksperimento rezultatai pateikiami 15 lentelėje ir 28 pav.

15 lentelė. PHP interpretatoriaus versijų įtaka

PHP versijų įtaka	php 5.4	php 5.5	php 5.6
Moodle 2.8.1	169	170	174
Moodle 2.7.3	169	169	177
Moodle 2.5.9	180	182	187
Moodle 2.2.11	225	221	225



28 pav. PHP interpretatoriaus versijų įtaka

„Moodle“ greitaveika, naudojant skirtingas PHP interpretatoriaus versijas, kinta nežymiai. Internetiniai šaltiniai pažymi PHP 5.3 ir 5.7 versijų greitaveikos skirtumus (MacDonald, 2013) (Benchmarking The Performance Of PHP 5.5, 2015), bet dėl 5.3 versijos nesuderinamumo su naujausiomis „Moodle“ versijos jos testavimas ar naudojimas neturi prasmės. O 5.7 versija - dar tik ankstyvoje kūrimo stadijoje ir jos greitaveiką šiuo metu nėra galimybės įvertinti. „Moodle“ kol kas nėra suderinamas su *HipHop Virtual Machine (HHVM)* (Moodle General developer forum, 2015). Šios virtualiosios mašinos ypatybės leidžia ypatingai greitai vykdyti PHP kodą, kas suteikia potencialo ateityje panaudoti ją su „Moodle“.

4.9 „MOODLE“ SAVYBĖS

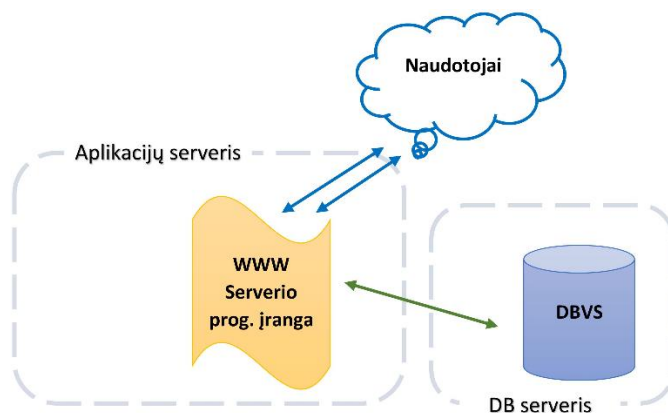
Ši eksperimentų serija skirta nustatyti „Moodle“ našumo bei greitaveikos priklausomybę nuo specifinių „Moodle“ savybių: versijų, turinio pobūdžio bei kiekio.

4.9.1 „Moodle“ versijų įtaka

Eksperimentas skirtas „Moodle“ programinio kodo versijų įtakai našumui bei greitaveikai nustatyti. Eksperimentas atliktas pasitelkus virtualiųjų serverių programinės įrangos konfigūravimo variaciją:

- Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6.4,
DBVS serveris: Centos 7, MySQL 5.6.22

Žiniatinklio (WWW) serveris bei DBVS talpinami skirtinguose virtualiuose serveriuose. Testavimo komponentai pateikiami 29 pav. ir 16 lentelėje.



29 pav. Eksperimento komponentų schema

16 lentelė. Testavimo aplinkos konfigūracija

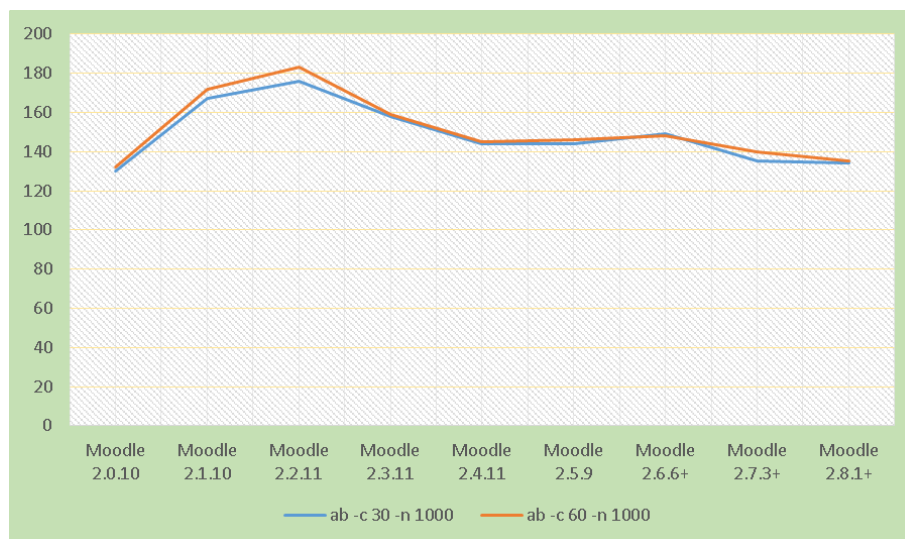
Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualieji serveriai:	Programų serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb)
Programinė įranga:	Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6.4 DBVS serveris: Centos 7, MySQL 5.6.22

Apache žiniatinklio serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas įgyvendintas OPCache plėtiniumi. Testuojama 2.7 ir 2.8 „Moodle“ versijų greitaveika, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų. Greitaveikos matavimai atlikti AB programa, imituojant 30 ir 60 lygiagrečių naudotojų darbą 1000 užklausų atlikti.

Gauti eksperimento rezultatai pateikiami 17 lentelėje ir 30 pav.

17 lentelė. „Moodle“ programinio kodo įtaka

Moodle versijų įtaka	ab -c 30 -n 1000	ab -c 60 -n 1000
Moodle 2.0.10	130	132
Moodle 2.1.10	167	172
Moodle 2.2.11	176	183
Moodle 2.3.11	158	159
Moodle 2.4.11	144	145
Moodle 2.5.9	144	146
Moodle 2.6.6+	149	148
Moodle 2.7.3+	135	140
Moodle 2.8.1+	134	135



30 pav. „Moodle“ programinio kodo įtaka

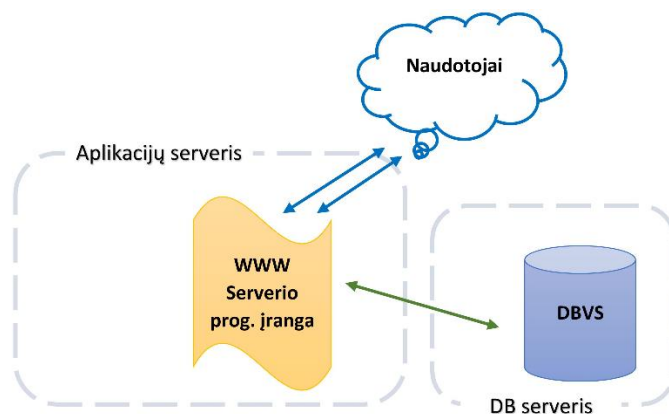
„Moodle“ greitaveiką ženkliai lemia „Moodle“ programinio kodo tobulinimas. Galima teigti, kad pirmosios „Moodle“ 2 versijos (2.0) pasižymėjo greitaveikos stoka, bet tolesnėse versijose (2.1, 2.2) kodo optimizavimo, klaidų taisymo pasėkoje greitaveika išaugo. Vėlesnėse versijose (2.3, 2.4, 2.5, 2.6, 2.7, 2.8) dėl nuolat papildomo funkcionalumo programinis kodas palaipsniui sudėtingėja, ir tai neigiamai veikia greitaveiką.

4.9.2 Turinio kiekio įtaka

Eksperimentas skirtas „Moodle“ programinio kodo versijų įtakai našumui bei greitaveikai nustatyti. Eksperimentas atliktas pasitelkus virtualiųjų serverių programinės įrangos konfigūravimo variacijas:

- Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6.4
DBVS serveris: Centos 7, MySQL 5.6.22;
- Programų serveris: Centos 7, Nginx 1.6.2 (php-fpm), PHP 5.6.4
DBVS serveris: Centos 7, MySQL 5.6.21.

Žiniatinklio (WWW) serveris bei DBVS talpinami skirtinguose virtualiuose serveriuose. Testavimo komponentai pateikiami 31 pav. ir 18 lentelėje.



31 pav. Eksperimento komponentų schema

18 lentelė. Testavimo aplinkos konfigūracija

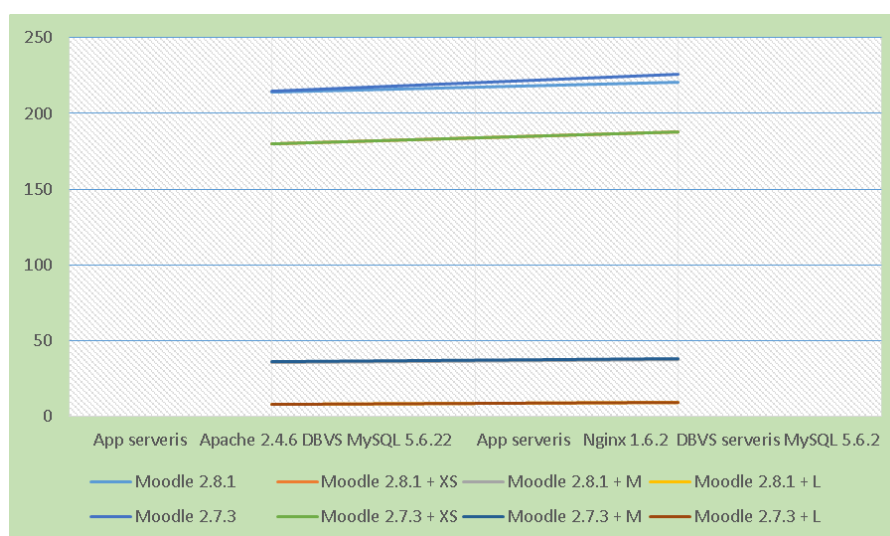
Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualieji serveriai:	Programų serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb)
Programinė įranga:	a) Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6.4 DBVS serveris: Centos 7, MySQL 5.6.22 b) Programų serveris (Centos 7, Nginx 1.6.2 (php-fpm), PHP 5.6.4) DBVS serveris (Centos 7, MySQL 5.6.21)

Apache žiniatinklio serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas realizuotas OPCache plėtiniu. Testuojama 2.7 ir 2.8 „Moodle“ versijų greitaveika, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų bei su XS, M ir L dydžio mokymosi kursais (žr. 1 lentelę). „Moodle“ greitaveikos matavimai atlikti AB programa, imituojant 60 lygiagrečių naudotojų darbą 60s laikotarpyje.

Gauti eksperimento rezultatai pateikiami 32 pav. ir 19 lentelėje.

19 lentelė. Turinio kiekio įtaka

Mokymosi kursų įtaka	App serveris Apache 2.4.6 DBVS MySQL 5.6.22	App serveris Nginx 1.6.2 DBVS serveris MySQL 5.6.2
Moodle 2.8.1	214	221
Moodle 2.8.1 + XS	180	188
Moodle 2.8.1 + M	35	37
Moodle 2.8.1 + L	8	9
Moodle 2.7.3	215	226
Moodle 2.7.3 + XS	180	188
Moodle 2.7.3 + M	36	38
Moodle 2.7.3 + L	8	9



32 pav. Turinio kiekio įtaka

Didėjant „Moodle“ turinio kiekiui DBVS serverio apkrova mažėja. Auga programų serverio CPU apkrova. „Moodle“ greیتaveika ženkliai lemiamą mokymosi kursų. Kuo didesni mokymosi kursai, tuo greیتaveika mažėja. Reikalingi papildomi eksperimentai nustatyti, kokie mokymosi kursų elementai labiausiai lemia „Moodle“ greیتaveiką. Taip pat reikalinga atlikti paiešką priemonių neigiamo poveikio minimizavimui.

4.9.3 „Moodle“ klasterio ir jo komponentų skaičiaus įtaka

Eksperimentas skirtas „Moodle“ programų klasterio efektyvumo patikrinimui ir klasterį sudarančių elementų skaičiaus įtakai „Moodle“ našumui bei greیتaveikai nustatyti. Eksperimentas atliktas pasitelkus 3-is skirtingus „Moodle“ programų klasterio variantus:

- a) klasteris sudarytas iš vieno programų serverio;
- b) klasteris sudarytas iš dviejų programų serverių;
- c) klasteris sudarytas iš trijų programų serverių.

„Moodle“ programų klasterį sudaro keletas skirtingas funkcijas atliekančių serverių. Priklausomai norimam pasiekti našumui jų sudarančių serverių skaičius gali kisti. Pasaulinėje praktikoje dažnai naudojamas automatizuotas klasterį sudarančių komponentų skaičiaus kitimas reaguojant į pasikeitusį naudotojų skaičių.

Ugniasienė filtruoja neteisingas bei kenksmingas užklausas. Tai sumažina apkrovą įprastu atveju tenkančių žiniatinklio serveriui.

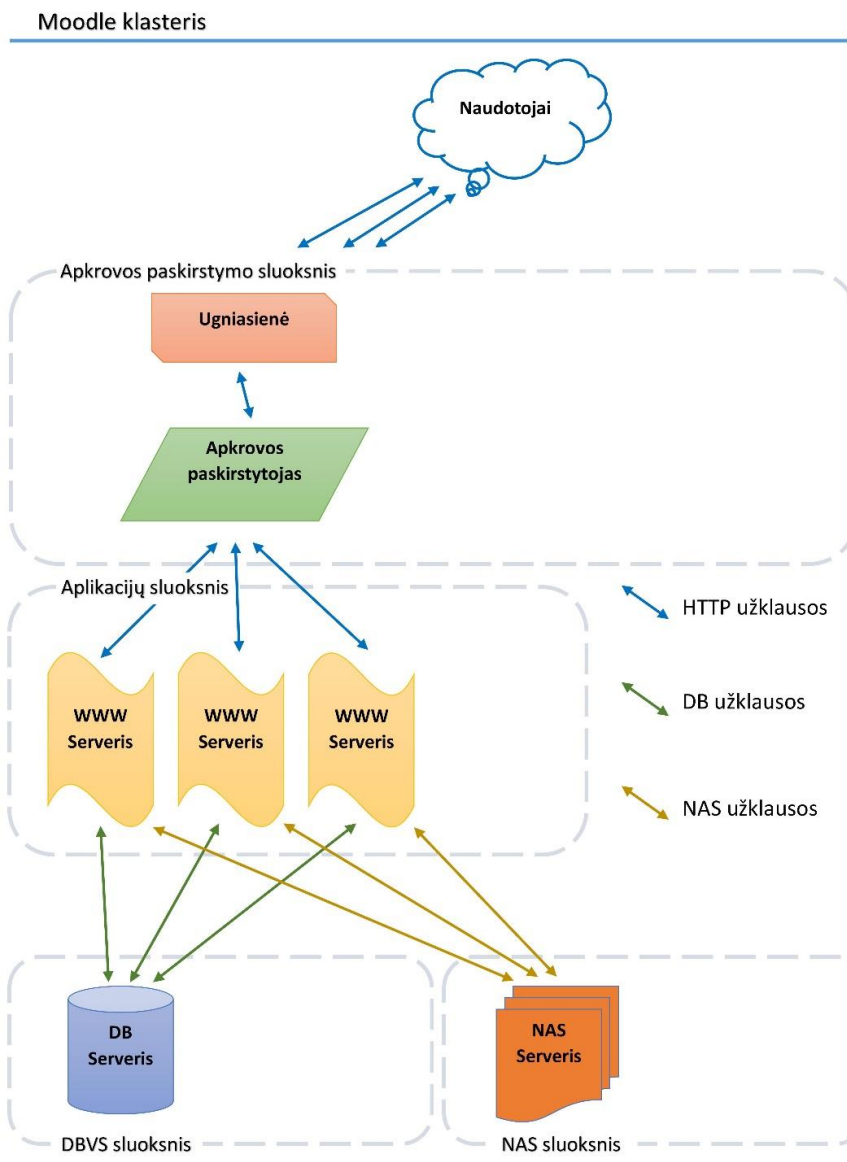
Apkrovos paskirstytojas tolygiai paskirsto naudotojų užklausų srautą keletui lygiagrečiai dirbančių žiniatinklio serverių.

Keletas lygiagrečiai dirbančių žiniatinklio serverių: kiekvienas jų aptarnauja proporcingą dalį bendro naudotojų skaičiaus. Žiniatinklio serverių skaičiaus didinimas mažina vienam serveriui tenkantį naudotojų skaičių. Mažėjantis naudotojų skaičius vienam „Moodle“ klasterį aptarnaujančiam žiniatinklio serveriui, didina bendrą sistemos našumą bei greیتaveiką.

NAS (Network attached storage) failų saugykla užtikrina „Moodle“ klasterį aptarnaujančių žiniatinklio serverių prieigą prie bendros failų saugyklos. NAS komponentas klasteryje leidžia kiekvienam žiniatinklio serveriui operuoti tais pačiais duomenimis.

DBVS serveris užtikrina „Moodle“ klasterį aptarnaujančių žiniatinklio serverių prieigą prie bendros duomenų bazės. „Moodle“ klasterio atveju, DBVS laikomos naudotojų prisijungimų sesijų duomenys. Tai padidina apkrovą DB serveriui. Todėl didelėse sistemose galimas keletas lygiagrečiai dirbančių DBVS panaudojimas.

Eksperimento komponentai pateikiami 33 pav. ir 20 lentelėje.



33 pav. Eksperimento komponentų schema

20 lentelė. Testavimo aplinkos konfigūracija

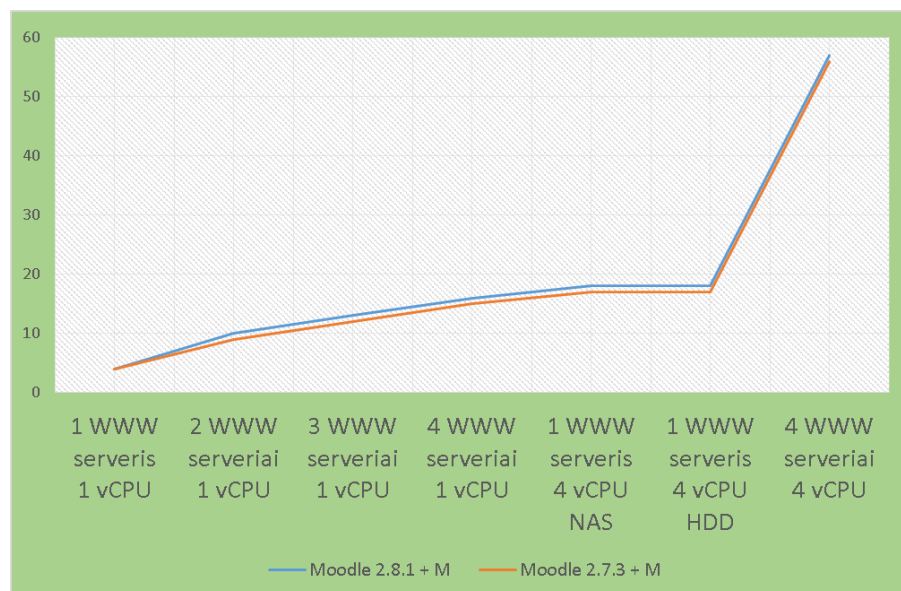
Techninė įranga:	Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz, 32Gb RAM
Virtualieji serveriai:	Ugniasienė bei apkrovos paskirstytojas (vCPU kiekis: 1, RAM kiekis: 2 Gb) WWW serveris (vCPU kiekis: 1, RAM kiekis: 2 Gb) DBVS serveris (vCPU kiekis: 4, RAM kiekis: 16 Gb) NAS domenų saugykla (vCPU kiekis: 4, RAM kiekis: 8 Gb)
Programinė įranga:	Ugniasienė bei apkrovos paskirstytojas (Iptables, HAProxy v1.5.4) Programų serveris (Centos 7, Apache 2.4.6, PHP 5.6.4) DBVS serveris (Centos 7, MySQL 5.6.21) NAS domenų saugykla (FreeNAS 9.3 (FreeBSD))

Ugniasienė įgyvendinta iptables priemonėmis. Užklausų srauto paskirstymui naudojama HAProxy. Apache žiniatinklio serveris dirba su integruotu PHP interpretatoriumi (mod_php). PHP kodo spartinimas realizuotas OPCache plėtiniu. NAS saugykla realizuota FreeNAS 9.3 programinės įrangos pagalba. FreeNAS dirba FreeBSD operacinės sistemos bazėje. DBVS serveris naudoja MySQL. Naudotojų sesijų identifikatorių saugojimui naudojama DBVS. Katalogas „Moodle“ data talpinamas NAS saugykloje, pasižyminčia ženkliai didesnės trukmės užklausomis. Neigiamai NAS įtakai kompensuoti įgalintas papildomas spartinančiosios saugyklos elementas (2 priedas), nesusijusius su failų sistema - *Alternative PHP cache* su išjungtu bytekodo spartinimu.

Testuojama 2.7 ir 2.8 „Moodle“ versijų greitaveika pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų bei su M dydžio mokymosi kursais (žr. 1 lentelę). Greitaveikos matavimai atlikti AB programa, imituojant 60 lygiagrečių naudotojų darbą 60 sekundžių laikotarpyje. Gauti eksperimento rezultatai pateikiami 34 pav. ir 21 lentelėje.

21 lentelė. „Moodle“ klasterio bei jo komponentų skaičiaus įtaka

Moodle klasterio, bei jo komponentų skaičiaus įtaka	1 WWW serveris 1 vCPU	2 WWW serveriai 1 vCPU	3 WWW serveriai 1 vCPU	4 WWW serveriai 1 vCPU	1 WWW serveris 4 vCPU NAS	1 WWW serveris 4 vCPU HDD	4 WWW serveriai 4 vCPU
Moodle 2.8.1 + M	4	10	13	16	18	18	57
Moodle 2.7.3 + M	4	9	12	15	17	17	56



34 pav. „Moodle“ klasterio bei jo komponentų skaičiaus įtaka

Šio eksperimento metu VMA „Moodle“ įdiegtas į žiniatinklio programų klasterį. Naudotojų sesijų identifikatoriai talpinami DB. Objektų spartinimas vykdomas įdiegus *Alternative PHP cache (APC)* spartinimo podėlį. Pirmųjų keturių testų metu matuojamas darbo našumas klasteryje su atitinkamai vienu,

dviem, trimis ir keturiais žiniatinklio serveriais. Našumo prieauglis proporcingai auga didinat žiniatinklio serverių darbą. Penktasis ir šeštasis testavimai atlikti su vienu serveriu „Moodle“ duomenis patalpinus NAS arba HDD. Sesijų talpinimas DB bei APC spartinimo podėlio naudojimas leidžia naudojant NAS tipo failų saugyklą neprarasti greitaveikos. Suminis keturių 1 vCPU turinčių žiniatinklio serverių našumas 12% mažesnis už 4 vCPU turintį serverį. Suminis keturių 4 vCPU turinčių žiniatinklio serverių našumas 316% didesnis už 4 vCPU turintį serverį. Eksperimento rezultatai rodo, kad klasterio darbo efektyvumas 12% - 26% mažesnis nei pavienio serverio. Todėl jo taikymas yra pagrįstas tik tais atvejais, kai turimos techninės įrangos galingiausio vieneto CPU galimybės nepakankamos pageidaujama „Moodle“ našumui bei greitaveikai pasiekti.

4.10 „MOODLE“ NAŠUMO ĮVERTINIMAS

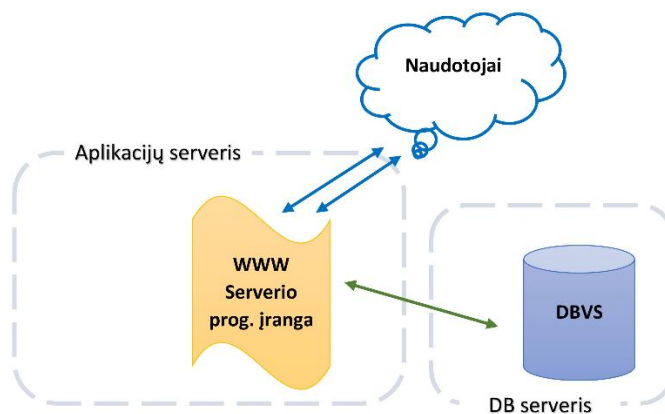
Įtampos testavimas – sistemos sąmoningas perkrovimas, siekiant išsiaiškinti jos funkcionalumą, esant kritinei situacijai. Įtampos testavimo metu nustatoma:

- saugūs sistemos naudojimo limitai;
- našumo atitikimas specifikacijai;
- sistemos darbo sutrikimų pobūdis.

Eksperimentas skirtas „Moodle“ našumo įvertinimui bei kritiniam „Moodle“ apkrovos taškui nustatyti. Eksperimentas atliktas pasitelkus virtualiųjų serverių programinės įrangos konfigūravimo variacijas:

- a) Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6.4
DBVS serveris: Centos 7, MySQL 5.6.22;
- b) Programų serveris: Centos 7, Nginx 1.6.2 (php-fpm), PHP 5.6.4
DBVS serveris: Centos 7, MySQL 5.6.21.

Žiniatinklis serveris bei DBVS talpinami skirtinguose virtualiuosiuose serveriuose. Eksperimento komponentai pateikiami 35 pav. ir 22 lentelėje.



35 pav. Eksperimento komponentų schema

22 lentelė. Testavimo aplinkos konfigūracija

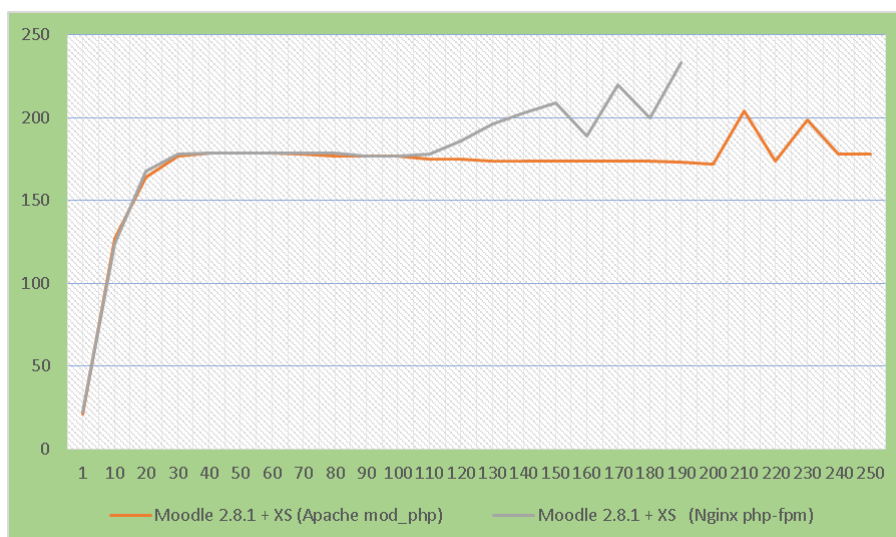
Techninė įranga:	PowerEdge R710, Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 72Gb RAM
Virtualieji serveriai:	Programų serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb) DBVS serveris (vCPU kiekis: 8, RAM kiekis: 8 Gb)
Programinė įranga:	a) Programų serveris: Centos 7, Apache 2.4.6 (mod_php), PHP 5.6.4 DBVS serveris: Centos 7, MySQL 5.6.22 b) Programų serveris: Centos 7, Nginx 1.6.2 (php-fpm), PHP 5.6.4 DBVS serveris: Centos 7, MySQL 5.6.21

Apache žiniatinklio serveris dirba su integruotu PHP interpretatoriumi (mod_php). Nginx serveris su PHP interpretatoriumi susietas per PHP-FPM procesų menedžerį. IIS žiniatinklio serveris PHP interpretatorių iškviečia FastCGI metodu. PHP kodo spartinimas realizuotas OPCache plėtiniumi. Testuojama 2.8 „Moodle“ versija, pasitelkus naujai įdiegtą „Moodle“ be mokymosi kursų. Greitaveikos matavimai atlikti AB programa, imituojant kintantį lygiagrečių naudotojų darbą 60s laikotarpyje.

Gauti eksperimento rezultatai pateikiami 36 pav. ir 23 lentelėje. Raudona spalva pažymėtos reikšmės gautos sutrikus sistemos stabilumui.

23 lentelė. „Moodle“ našumo įvertinimas

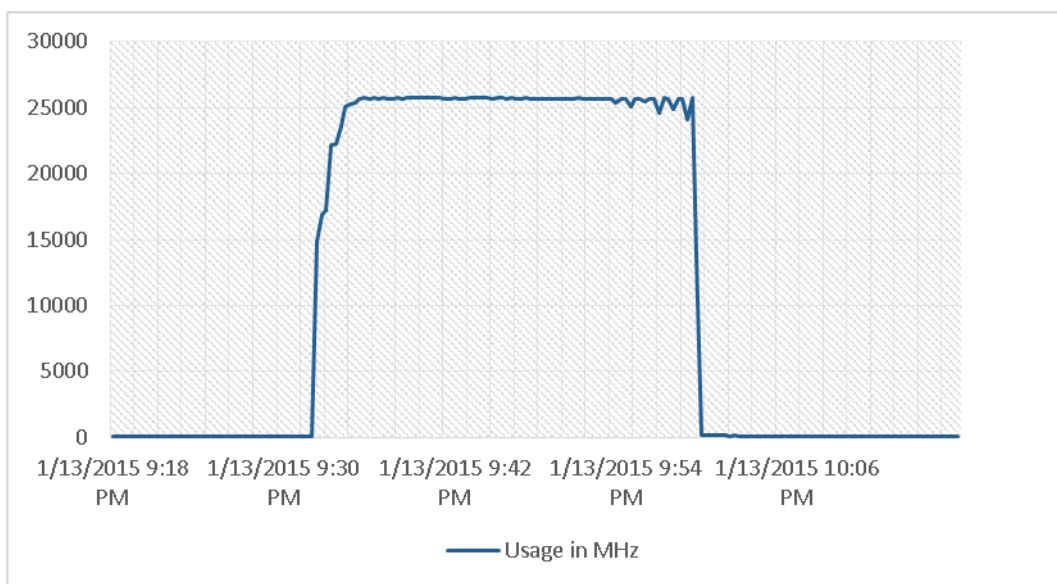
Vartotojai	1	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250
Moodle 2.8.1 + XS (Apache mod_php)	21	127	164	177	179	179	179	178	177	177	177	175	175	174	174	174	174	174	174	173	172	204	174	199	178	178
Moodle 2.8.1 + XS (Nginx php-fpm)	22	124	168	178	179	179	179	179	179	177	177	178	186	196	203	209	189	220	200	233						



36 pav. „Moodle“ našumo įvertinimas

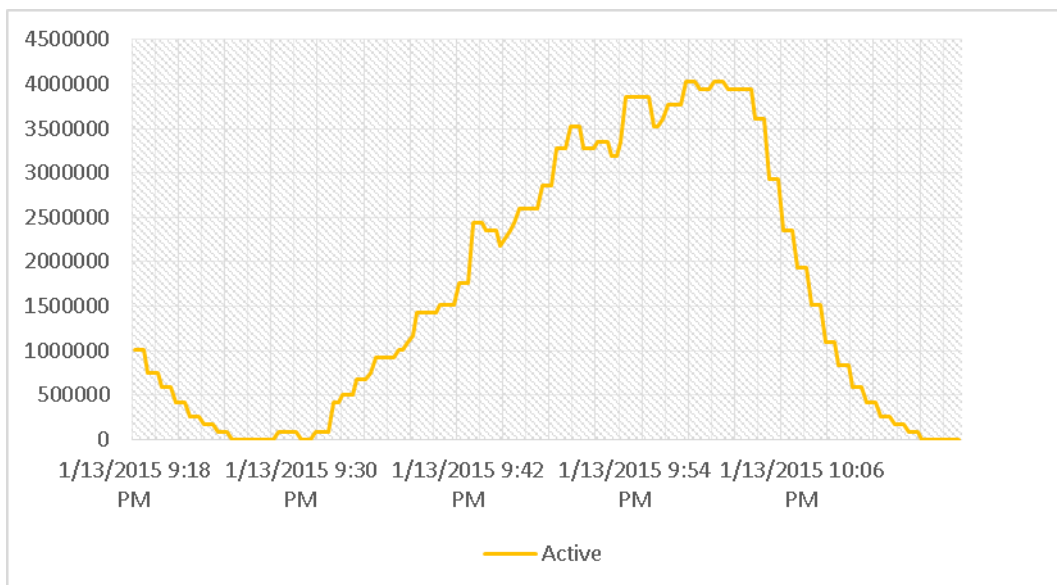
Didinant lygiagrečiai dirbančių naudotojų skaičių „Moodle“ greitaveika auga kol pasiekiamas 100% žiniatinklio serverio CPU naudojimas. Sistemos su dedikuotais 8-iais Intel(R) Xeon(R) CPU X5670 branduoliais galimybės pilnai išnaudojamos esant 30 ir daugiau lygiagrečiai dirbančių naudotojų. Sistema

praranda stabilumą prie 110 naudotojų, naudojant Nginx programinę įrangą, prie 200 naudotojų – Apache programinę įrangą.



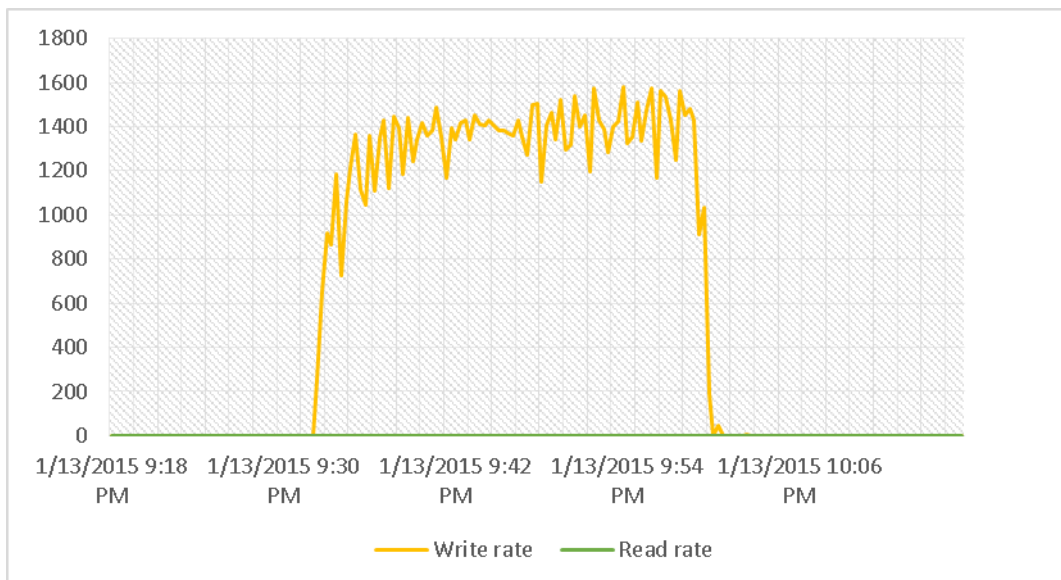
37 pav. CPU apkrova A testavimo aplinkos variante

CPU apkrova testo metu išlieka 100%. Dėl sutrikusio žiniatinklio serverio procesų darbo testavimo pabaigoje, naudotojų skaičiui pasiekus 200 vienetų, apkrovimo grafike pastebimi CPU apkrovos sumažėjimai (37 pav.).



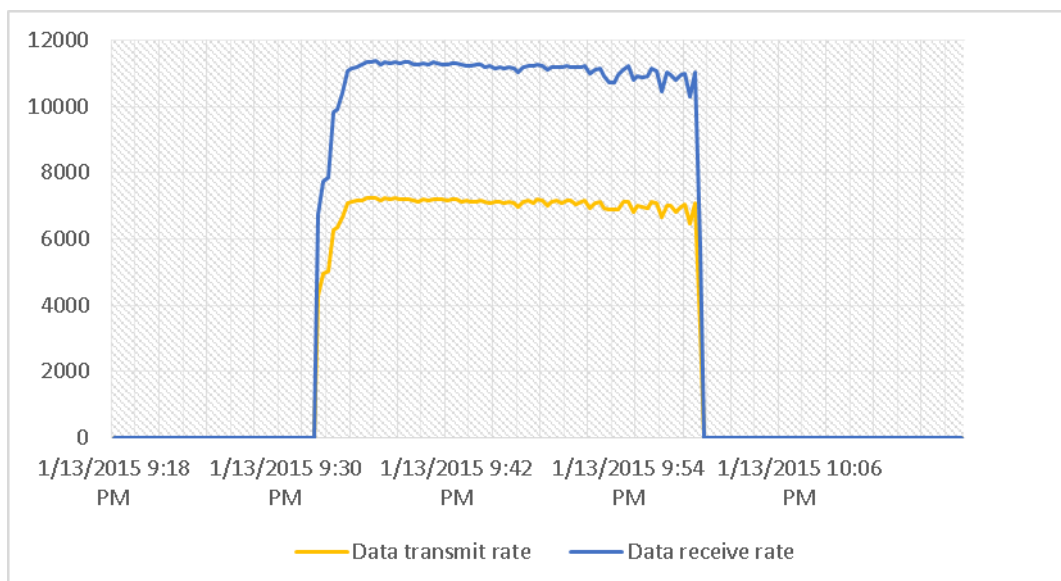
38 pav. Aktyvios RAM kiekis KB A testavimo aplinkos variante

Testo laikotarpyje stebimas tolygus aktyvios operatyviosios atminties kiekio didėjimas. Jos kiekio mažėjimas fiksuojamas tik testavimo procesui pasibaigus (38 pav.).



39 pav. Disko apkrova KBps A testavimo aplinkos variante

Diskinės posistemės aktyvumo lygis sutampa su CPU aktyvumu. Testo metu fiksuojamas vidutinis 1.3 MB/s rašymo į diską operacijų srautas. Rašymo srautas susijęs su „Moodle“ sesijų identifikatorių talpinimu *moodledata* kataloge (39 pav.). Talpinant „Moodle“ sesijų identifikatorius DBVS tikėtina ženkliai mažesnė diskinės sistemos apkrova.

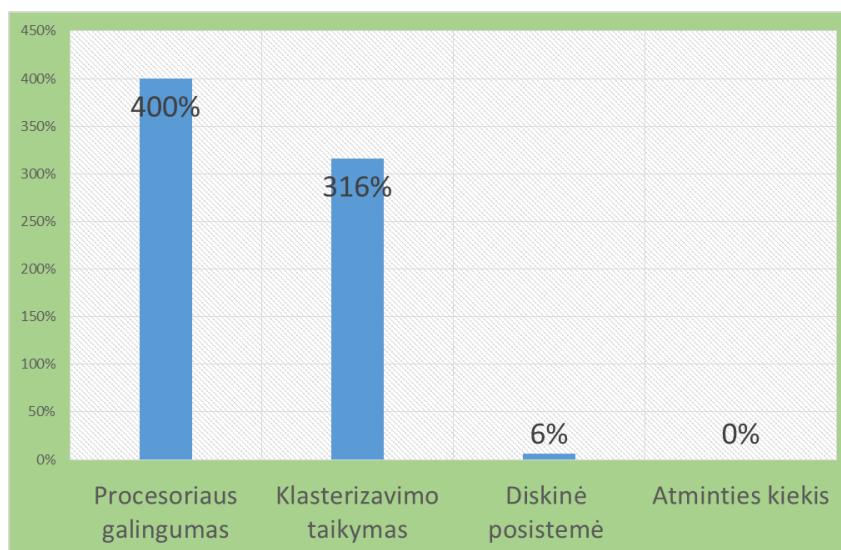


40 pav. Duomenų srautai tinkle KBps A testavimo aplinkos variante

Tinklo duomenų srauto lygis tiksliai atkartoja CPU aktyvumo lygį testo laikotarpyje. Testo laikotarpiu stebimas 11 MB/s išsiunčiamų duomenų srautas ir tai sudaro apie 10% testavimo aplinkos tinklinių ryšių pralaidumo (40 pav.). Tai dar kartą pabrėžia CPU pajėgumo svarbą „Moodle“ našumui bei greitaveikai.

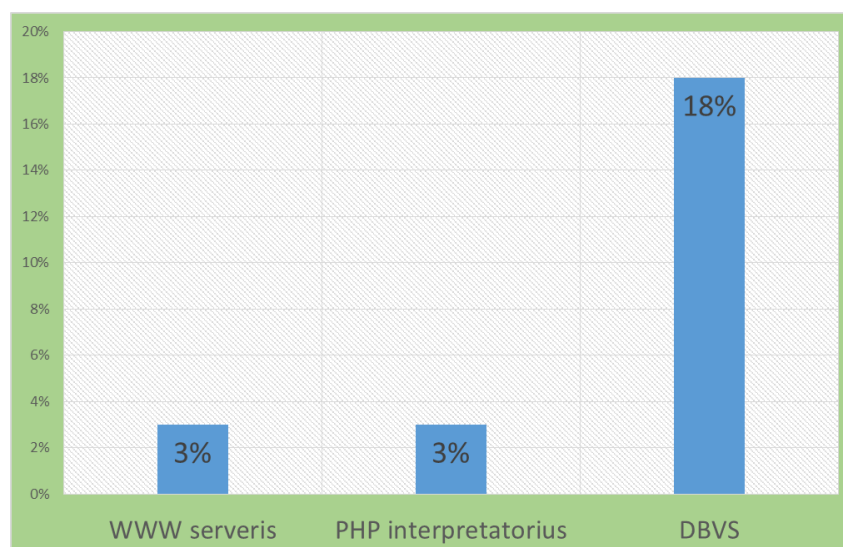
4.11 EKSPERIMENTINIO TYRIMO APIBENDRINIMAS

„Moodle“ funkcionavimui būtinas sisteminių komponentų derinys, būdingas tipinėms PHP programoms. Tai – techninės įrangos, operacinės sistemos, žiniatinklio serverio, PHP interpretatoriaus bei duomenų bazių valdymo sistemos derinys. Literatūroje pateikiamas platus spektras PHP programų spartinimo metodikų (Armando Padilla, 2010; Eric Higgins, 2014; King, 2008; Padilla & Hawkins, 2010; Moodle performance recommendations, 2014; Sheltren, Newton, & Catchpole, 2014; Souders, 2007; Xin Guo, Qing Shi, Danjue Zhang, 2013). Šios spartinimo technikos apima visas „Moodle“ funkcionavimui būtinas techninės bei programinės įrangos komponentus. Bet dėl VMA „Moodle“ programinės įrangos specifikos daug literatūroje aprašomų žiniatinklio programų spartinimo technikų arba neefektyvios, arba turi neigiamą poveikį (Armando Padilla, 2010; Eric Higgins, 2014; King, 2008; Padilla & Hawkins, 2010; Moodle performance recommendations, 2014; Sheltren, Newton, & Catchpole, 2014; Souders, 2007; Xin Guo, Qing Shi, Danjue Zhang, 2013). Pavyzdžiui, tokios tipinės rekomenduojamos priemonės, kaip turinio kompresija, neigiamai veikia „Moodle“ našumą bei greitaveiką. Dalies spartinimo technikų taikymas negalimas dėl su jomis susijusio poreikio keisti „Moodle“ programinę įrangą.



41 pav. Testų metu fiksuoti maksimalūs našumo bei greitaveikos padidėjimai, susiję su techninės įrangos konfigūracijos pokyčiais

Techninė įrangos pajėgumas pagrindinis veiksnys, lemiantis VMA „Moodle“ našumą bei greitaveiką (41 pav.). „Moodle“ aptarnaujančios sistemos procesoriaus pajėgumo didinimas arba kelių sistemų apjungimas į klasterį duoda didžiausią „Moodle“ našumo bei greitaveikos prieauglį. Palyginus su procesoriaus našumu diskinės posistemės greitis turi santykinai mažą įtaką „Moodle“ našumui bei greitaveikai. Esant pakankamam operatyviosios atminties kiekiui tolesnis jos apimties didinimas „Moodle“ greitaveikai įtakos nedaro. Operatyviosios atminties apimties didinimas prasmingas tada, kai dėl jos trūkumo naudojama diskinė posistemė.



42 pav. Testų metu fiksuoti maksimalūs našumo bei greitaveikos padidėjimai, susiję su programinės įrangos konfigūracijos pokyčiais

„Moodle“ programinės įrangos specifika neleidžia efektyviai taikyti spartinimo metodų, susijusių su žiniatinklio serverio programinės įrangos parinkimu. „Moodle“ greitaveika labai menkai priklauso nuo žiniatinklio serverio gebėjimo greitai pateikti statinį turinį, nuo jo operatyviosios atminties sunaudojimo ar maksimalaus aptarnaujamų klientų skaičiaus (42 pav.). Literatūros šaltiniai rekomenduoja žiniatinklio serverio programinę įrangą keisti įrangą, vienam klientui sunaudojančia mažiau operatyviosios atminties. Tai ženkliai padidina maksimalų galimą aptarnauti klientų skaičių mašinos, turinčiose mažą operatyviosios atminties kiekį (Armando Padilla, 2010). Kitose rekomendacijose patariama rinktis žiniatinklio serverio programinę įrangą, sparčiau pateikiančią statinius puslapio elementus: HTML, grafiką ir panašiai (Armando Padilla, 2010). VMA „Moodle“ atveju, šios priemonės neturi teigiamos įtakos. Didžioji „Moodle“ naudotojui pateikiamo turinio dalis generuojama dinamiškai. O naudotojų skaičius santykinai nėra didelis. Todėl žiniatinklio serverio programinės įrangos keitimas arba konfigūracijos parametrų modifikavimas neefektyvus ir greičiau gali privesti prie „Moodle“ našumo bei greitaveikos sumažėjimo.

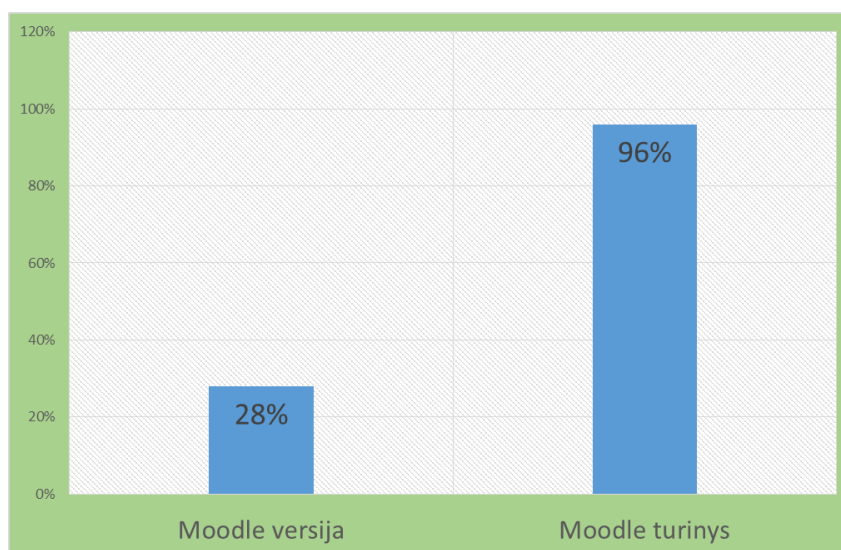
„Moodle“ našumą bei greitaveiką nulemia PHP interpretatoriaus turinio generavimo sparta. Esant įgalintam PHP mašininio kodo spartinimui kiti PHP interpretatoriaus spartinimo metodai suteikia santykinai nedidelį „Moodle“ darbo pagreitėjimą – iki 3%. Kadangi mašininio kodo spartinimas šiandieninėse PHP distribucijose įgalintas, pagal nutylėjimą reiktų laikyti, kad PHP interpretatoriaus modifikavimas siekiant pagerinti „Moodle“ darbą taip pat netikslingas.

Nagrinėjant sisteminę programinę įrangą ir jos įtaką „Moodle“ greitaveikai DBVS programinės įrangos parinkimas bei jos konfigūravimo modifikacijos efektyviausios didinant „Moodle“ maksimalų našumą bei greitaveiką (žr. 42 pav.). DBVS negali veikti optimaliai esant tipiniams DBVS konfigūravimo

parametrams. DBVS konfigūravimo parametrai derinami tiek prie techninės įrangos galimybių, tiek prie operuojamų duomenų specifikos. Todėl DBVS priderinimas prie esamos situacijos suteikia didžiausią teigiamą poveikį „Moodle“ našumui bei greitaveikai.

HTML turinio optimizavimas, informacijos suspaudimas, CSS optimizavimas, grafikos optimizavimas skirti sumažinti siunčiamo turinio apimtį, tuo pagreitinant puslapio užkrovimo greitį kliento pusėje. Bet „Moodle“ atveju, problemos susijusios su dinaminio turinio parengimo siųsti laiku, o ne su jo siuntimo laiku. Todėl tokios priemonės, kaip GZIP turinio spaudimas, nepagreitina „Moodle“ turinio užkrovimo greičio, bet tą laiką prailgina, nes turinio spaudimui procesorius apkraunamas papildomomis užduotimis, trukdančiomis „Moodle“ turinio generavimui.

Operacinės sistemos įtaka tyrimo metu nebuvo galutinai nustatyta, todėl išvados dėl operacinės sistemos įtakos „Moodle“ našumui bei greitaveikai šiame darbe nepateikiamos.



43 pav. Testų metu fiksuotas maksimalus našumo bei greitaveikos neigiamas pokytis, susijęs su „Moodle“ versija bei turiniu

Yra daug veiksnių, negatyviai veikiančių našumą bei greitaveiką, susijusių su „Moodle“ aplinkos programiniu kodu bei „Moodle“ teikiamu mokymosi turiniu. Tai „Moodle“ programinės įrangos versijų kaita, papildinių diegimas, mokymosi objektų gausa. „Moodle“ nuolat tobulinamas, įgalinami nauji priedai, pridedamas papildomas funkcionalumas. Visa tai didina „Moodle“ programinio kodo apimtį. Jo vykdymas tampa vis labiau komplikuoju. Naujų „Moodle“ programinės įrangos versijų darbas pasižymi našumo bei greitaveikos mažėjimu (43 pav.). Kitas, ypatingai neigiamai „Moodle“ darbo greitį veikiantis veiksnys yra mokymosi objektų skaičius. Didelius kiekius naudotojų, kursų ar užduočių talpinantis VMA „Moodle“ pasižymi ženkliai mažesne greitaveika, nei talpinantis mažesnius mokymosi objektų skaičius.

IŠVADOS

1. Literatūros analizė parodė, kad egzistuoja daug žiniatinklio programų spartinimo technikų, tačiau jų praktinis pritaikymas priklauso nuo VMA „Moodle“ programinės įrangos specifikos.
2. Techninės įrangos pajėgumas - pagrindinis veiksnys, lemiantis VMA „Moodle“ našumą bei greitaveiką. Išvados, susijusios su VMA „Moodle“ aptarnaujančiais techniniais komponentais:
 - žiniatinklio serverio, aptarnaujančio VMA „Moodle“, procesoriaus pajėgumo didinimas – pagrindinis būdas pagreitinti „Moodle“ aplinkos darbą;
 - esant nepakankamam procesoriaus pajėgumui efektyvu apjungti keletą lygiagrečiai dirbančių žiniatinklio serverių į klasterį;
 - operacinės atminties kiekis turi būti pakankamas aptarnauti maksimalų žiniatinklio serverio procesoriaus aptarnaujamų naudotojų skaičių. Tolesnis operacinės atminties apimties didinimas nekeičia „Moodle“ greitaveikos;
 - „Moodle“ duomenų talpinimas NAS tipo duomenų saugykloje gali neigiamai veikti „Moodle“ našumą bei greitaveiką. NAS neigiama įtaka kompensuojama perkeliant „Moodle“ spartinančiosios atmintinės podėlius į DBVS bei operatyviają atmintį (2 priedas).
3. Šiandien dauguma „Moodle“ darbui reikalingų sisteminės programinės įrangos komponentų dirba optimaliai. Gamintojo parinkti sisteminės programinės įrangos konfigūravimo parametrai užtikrina optimalų darbo našumą bei greitaveiką.
4. Sisteminės programinės įrangos komponentų tarpe išskirtinas DBVS poveikis „Moodle“ darbo greičiui. DBVS programinės įrangos parinkimas bei jos konfigūravimo modifikacijos efektyviausiai didina „Moodle“ našumą bei greitaveiką.
5. Pagrindiniai veiksniai, neigiamai veikiantys našumą bei greitaveiką ir susiję su „Moodle“ aplinkos programiniu kodu bei „Moodle“ teikiamu mokymosi turiniu – tai „Moodle“ programinės įrangos versijų kaita, papildinių diegimas, mokymosi objektų gausa.
6. Remiantis atliktais eksperimentais parengtos rekomendacijos „Moodle“ našumui bei greitaveikai užtikrinti (1 priedas) gali būti praktiškai taikomos švietimo institucijose.

LITERATŪRA

1. Microsoft. (2010). *Performance Testing Guidance for Web Applications*. O'Reilly Media, Inc.
2. (2015 m. 03 01 d.). Nuskaityta iš Benchmarking The Performance Of PHP 5.5: http://www.phoronix.com/scan.php?page=news_item&px=MTM0NjI
3. Armando Padilla, T. H. (2010). *Pro PHP Application Performance: Tuning PHP Web Projects for Maximum Performance*. Apress.
4. Capterra. (2014 m. 10 01 d.). *Top LMS Software*. Nuskaityta iš <http://www.capterra.com/learning-management-system-software/#infographic>
5. Eric Higgins. (2014 m. 01 23 d.). *PHP performance tips*. Nuskaityta iš PHP performance tips: <https://developers.google.com/speed/articles/optimizing-php>
6. *Free & open source, high-performance, distributed memory object caching system*. (2014 m. 01 22 d.). Nuskaityta iš Memcached: <http://memcached.org/>
7. *Hypertext Transfer Protocol*. (2014 m. 06 16 d.). Nuskaityta iš Wikipedia: http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
8. King, A. B. (2008). *Website Optimization*. O'Reilly Media, Inc.
9. MacDonald, R. (2013 m. 08 01 d.). *PHP Roadmap & Performance*. Nuskaityta iš The Homegrown Hosting Blog: <https://blog.asmallorange.com/2013/08/php-roadmap-performance/>
10. Microsoft Corporation. (2007). *Performance Testing Guidance for Web Applications*. Microsoft Corporation.
11. *Moodle General developer forum*. (2015 m. 02 10 d.). Nuskaityta iš Moodle and HipHop for PHP: <https://moodle.org/mod/forum/discuss.php?d=190767>
12. *Moodle performance recommendations*. (2014 m. 01 22 d.). Nuskaityta iš Moodle: http://docs.moodle.org/26/en/Performance_recommendations
13. moodle.org. (2015 m. 01 24 d.). *Installing Moodle*. Nuskaityta iš moodle.org: http://docs.moodle.org/26/en/Installing_Moodle
14. *Moodle: A Low-Cost Solution for Successful e-Learning*. (2015 m. 02 01 d.). Nuskaityta iš Learning Solutions Magazine: <http://www.learningsolutionsmag.com/articles/71/moodle-a-low-cost-solution-for-successful-e-learning>
15. Padilla, A., & Hawkins, T. (2010). *Pro PHP Application Performance*. apress.
16. Sheltren, J., Newton, N., & Catchpole, N. (2014). *High Performance Drupal*. O'Reilly Media, Inc.
17. Solutions, Lambda. (2013 m. 01 24 d.). *Moodle Market Share Statistics*. Nuskaityta iš <http://blog.lambdasolutions.net/moodle-market-share-statistics-show-its-the-market-leader>
18. Souders, S. (2007). *High Performance Web Sites*. O'Reilly Media, Inc.

19. *Wikipedia - List of PHP accelerators*. (2014 m. 01 22 d.). Nuskaityta iš Wikipedia:
http://en.wikipedia.org/wiki/List_of_PHP_accelerators
20. Xin Guo, Qing Shi, Danjue Zhang. (2013). A Study on Moodle Virtual Cluster in Cloud Computing.
2013 Seventh International Conference on Internet Computing for Engineering and Science.

PRIEDAI

1. Rekomendacijos „Moodle“ našumui bei greitaveikai užtikrinti, 4 lapai.
2. „Moodle“ spartinančiosios atmintinės podėlio diegimas, 2 lapai.
3. Straipsnis „Sisteminės aplinkos optimizavimas VMA „Moodle“ našumui bei greitaveikai“, 10 lapų.
4. Pažyma „Apie Remigijaus Kuto dalyvavimą tarptautinėje konferencijoje Alta‘2015“, 1 lapas.

REKOMENDACIJOS „MOODLE“ NAŠUMUI BEI GREITAVEIKAI UŽTIKRINTI

Serverio procesoriaus pajėgumas. Serverio, aptarnaujančio VMA „Moodle“, procesoriaus galia – pagrindinis veiksnys, lemiantis greitaveiką. VMA „Moodle“ aptarnaujančiose sistemose nepakankama procesoriaus galia tampa kritiniu veiksniumi, apribojančiu sistemos greitaveiką. Rekomenduotina, kad WWW serverio apkrovos pikas neiššauktų:

- procesoriaus apkrovos didesnės nei 80%,
- procesoriaus eilės ilgis neviršytų 2 vienam sistemos procesoriui.

Operatyviosios atminties kiekis. Esant nepakankamam operatyviosios atminties kiekiui sistema naudoja diskinę sistemą, o tai ženkliai neigiamai veikia sistemos greitaveiką. Minimalus operatyviosios atminties kiekis – 256MB, „Moodle“ aptarnaujančiam serveriui rekomenduojama turėti 1GB ar daugiau atminties. Pagrindinė taisyklė, numatant operatyviosios atminties kiekį – „Moodle“ aptarnaujantis WWW serveris nuo 10 iki 20 lygiagrečiai dirbančių naudotojų aptarnavimui naudoja 1GB operatyviosios atminties. Taigi, sistemą, kuri turės aptarnauti 100 lygiagrečiai tuo pačiu metu dirbančių naudotojų, reikia aprūpinti 6 Gb operatyviosios atminties. Tai orientacinis dydis - praktinis aptarnaujamų naudotojų skaičius priklauso nuo žiniatinklio serverio programinės įrangos, PHP interpretatoriaus darbo režimo bei kitų sistemos konfigūracijos ypatumų. Kitas būdas praktiškai nustatyti būtiną žiniatinklio serverio operatyviosios atminties kiekį – pasinaudoti žiniatinklio serverio našumo testu. Žiniatinklio serverio našumo testavimo metu, esant 100% procesoriaus apkrovai apie 20% operatyviosios atminties turi likti nepanaudota. Jei testo metu serverio operatyvioji atmintis išnaudojama 100% ir operacinė sistema pradeda naudoti *swap* atmintinę, reikia didinti operatyviosios atminties kiekį.

Diskinė posistemė. Esant nepakankamai diskinės posistemės greitaveikai „Moodle“ darbo greitis pastebimai lėtėja. Dažniausiai, tai nutinka naudojant NAS tipo duomenų saugyklas. Dėl to, kad kreipiniai į failų saugyklą perduodami kompiuterių tinklu, drastiškai išauga užklausų į diskinę sistemą trukmė. Tai neigiamai veikia „Moodle“ greitaveiką. Norint kompensuoti neigiamą lėtos diskinės sistemos poveikį reikia įgalinti naudotojų sesijų identifikatorių saugojimą DBVS bei įdiegti spartinančiąją atmintinę, saugojančią duomenų objektus operatyviojoje atmintyje. Pavyzdžiui, *mecache*, *mecached* arba *APCu* (Spartinančiosios atmintinės podėlio diegimo instrukcija pateikiama 2 priede).

Įgalintas PHP bytecode spartinimas. Prieš keletą metų PHP interpretatorius daugelyje distribucijų pagal numatymą neturėdavo bytecode spartinimo priemonių. Todėl bytecode spartinimo priemonių diegimas tokiose sistemose lemdavo keletą ar keliolika kartų siekiantį ženklų greitaveikos augimą. Daugelis šiandieninių „Linux“ distribucijų ar PHP diegimo paketų kitoms operacinėms sistemoms yra aprūpinti PHP

interpretatoriumi su, pagal numatymą, įgalintu bytecode spartinimu. Todėl „Moodle“ veikimui reikia užtikrinti, kad PHP interpretatorius turėtų įgalintus visus „Moodle“ darbui reikalingus plėtinius: *iconv, mbstring, curl, openssl, tokenizer, xmlrpc, soap, ctype, zip, gd, simplexml, spl, pcre, dom, xml, intl, json* bei plėtinį darbui su pasirinkta DBVS.

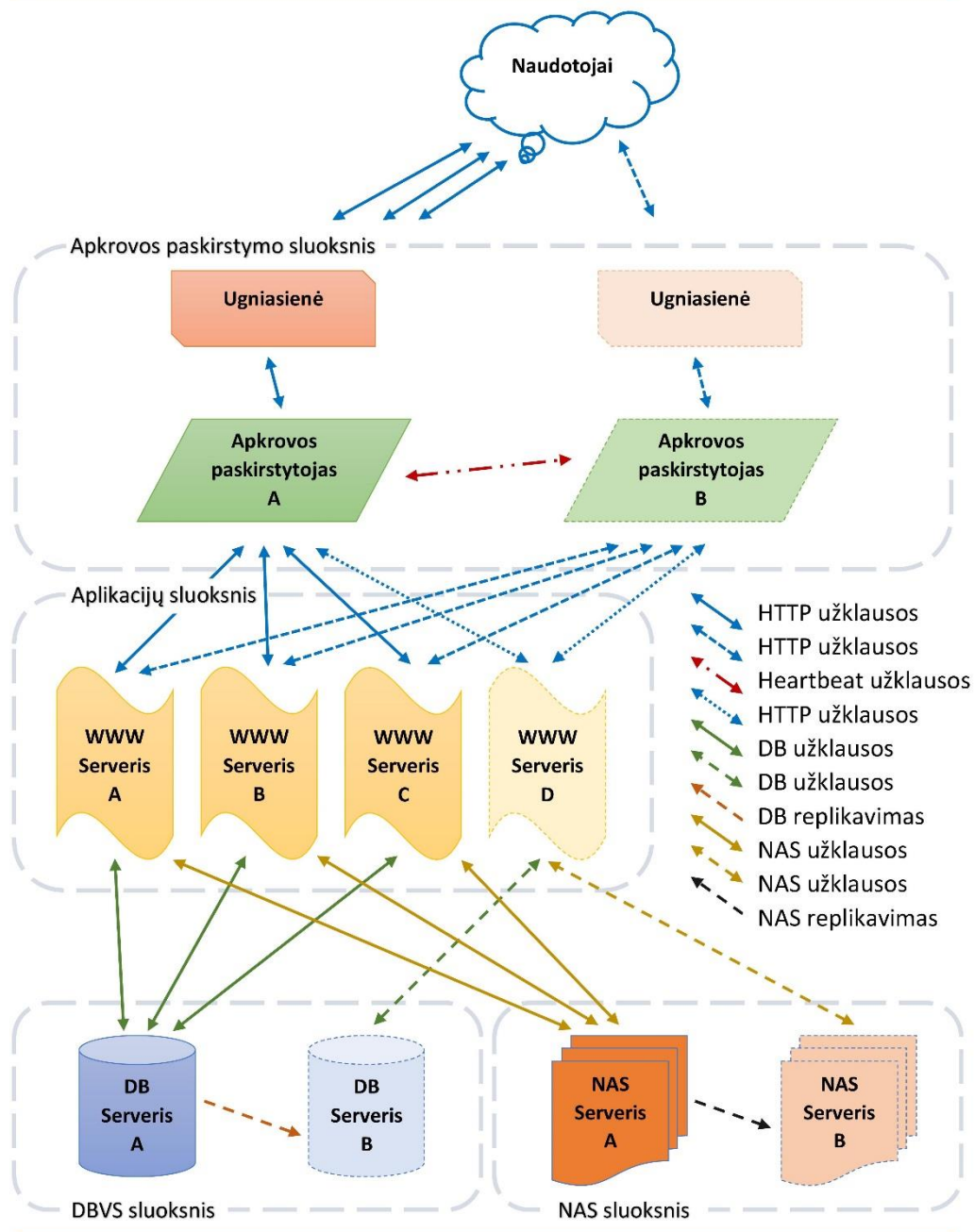
Užtikrinti optimalų DBVS darbą. DMBS darbo kokybė gali ženkliai veikti bendrą sistemos greitaveiką. Įtaka gali siekti nuo kelių iki keliolikos procentų. Labai svarbu DBVS galimybė aptarnauti apkrovos pikus. Pavyzdžiui, dėl nepakankamo aptarnaujamo jungčių skaičiaus naudojimo piko metu, gali sutrikti visos paslaugos darbas. Kiekviena DBVS turi savo konfigūravimo parametrus, jų parinkimas vyksta kiekvienu atveju individualiai. DBVS sistemos darbo optimizavimas, dažniausiai susijęs su DB duomenis, indeksus, užklausas ar kitą informaciją talpinančių spartinančiųjų atmintinių apimčių suderinimu su turimu duomenų kiekiu bei DBVS serverio techninėmis charakteristikomis.

Pavyzdys: MySQL su *InnoDB* DBVS konfigūravimo parametras *innodb_buffer_pool_size*. Šis parametras nustato *InnoDB* spartinančiosios atmintinės duomenims apimtį. Jei *innodb_buffer_pool_size* parametras nustato mažos apimties duomenų spartinančiąją atmintinę, o DBVS serveris turi pakankamai operatyviosios atminties, parametro reikšmės didinimas daugiau DB duomenų iš diskinės sistemos perkelia į operatyviają atmintį. Kreipinių skaičiaus į diskinę posistemę sumažinimas ženkliai paspartina DBVS užklausų vykdymą. Kartu paspartėja VMA „Moodle“ darbas.

„Moodle“ aptarnaujančios sistemos klasterizavimas. Išnaudojus programinės įrangos optimizavimo, techninės įrangos gerinimo galimybes, vienintelis būdas padidinti VMA „Moodle“ našumą bei greitaveiką yra žiniatinklio programų klasterio kūrimas. Žiniatinklio programų klasterį sudaro keletas žiniatinklio serverių, veikiančių kaip vieninga sistema.

„Moodle“ programų klasterį sudaro keletas skirtingas funkcijas atliekančių serverių. Priklausomai norimam pasiekti našumui jų sudarančių serverių skaičius gali kisti. Pasaulinėje praktikoje dažnai naudojamas automatizuotas klasterį sudarančių komponentų skaičiaus kitimas, reaguojant į pasikeitusią apkrovą.

Klasteriai naudojami ir norint padidinti sistemų patikimumą. Kadangi klasteris sudarytas iš keleto lygiagrečiai tą pačią funkciją atliekančių komponentų, vieno ar kelių komponentų gedimas negali sutrikdyti bendro sistemos darbo, nes jų funkcijas perima dubliuojantys komponentai (1 pav.).



1 pav. Padidinto patikimumo „Moodle“ klasterio komponentų schema

Ugniasienė, filtruojanti neteisingas bei kenksmingas užklauso, sumažina apkrovą įprastu atveju tenkančią žiniatinklio serveriui.

Apkrovos paskirstytojas tolygiai paskirsto naudotojų užklauso srautą keletui lygiagrečiai dirbančių žiniatinklio serverių.

Keletas lygiagrečiai dirbančių žiniatinklio serverių aptarnauja proporcingą dalį bendro naudotojų skaičiaus. Žiniatinklio serverių skaičiaus didinimas mažina vienam serveriui tenkantį naudotojų skaičių. Mažėjantis naudotojų skaičius vienam „Moodle“ klasterį aptarnaujančiam žiniatinklio serveriui didina bendrą sistemos našumą bei greitaveiką.

NAS (angl. *Network attached storage*) failų saugykla užtikrina „Moodle“ klasterį aptarnaujančių žiniatinklio serverių prieigą prie bendros failų saugyklos. NAS komponentas klasteryje leidžia kiekvienam žiniatinklio serveriui naudoti tuos pačius duomenis.

DBVS serveris užtikrina „Moodle“ klasterį aptarnaujančių žiniatinklio serverių prieigą prie bendros duomenų bazės. „Moodle“ klasterio atveju DBVS laikomos naudotojų prisijungimų sesijų duomenys. Tai padidina apkrovą DB serveriui. Todėl didelėse sistemose galimas keletu lygiagrečiai dirbančių DBVS panaudojimas.

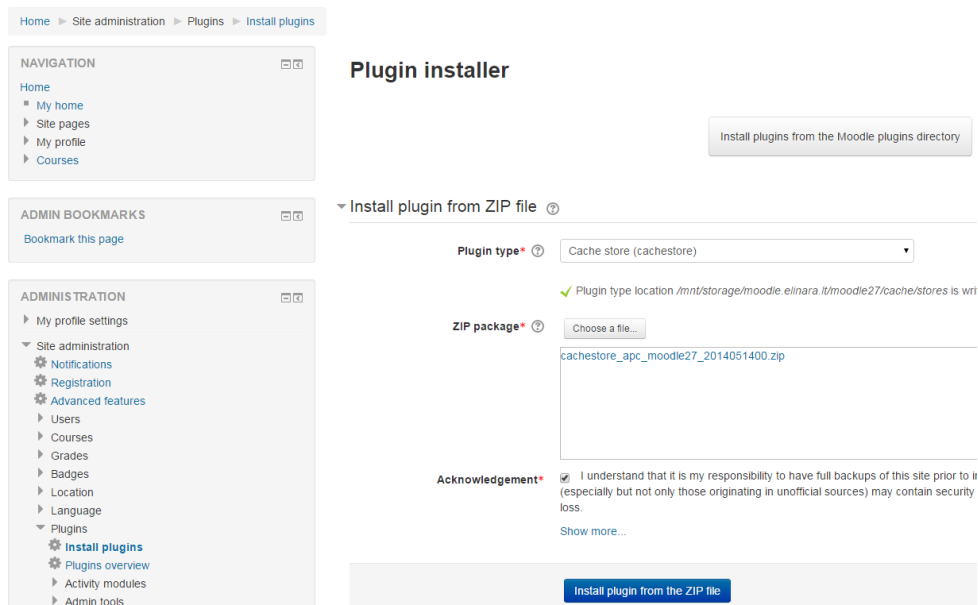
VMA „Moodle“ turinys. Reikia atminti, kad „Moodle“ greitaveika mažėja augant mokymosi objektų skaičiui – didėjant studentų, mokymosi kursų, užduočių ar forumų kiekiui. Todėl, tam tikrais atvejais, naudotojų srautus galima paskirstyti į keletą skirtingų „Moodle“ aplinkų. „Moodle“ našumas taip pat gali ženkliai sumažėti įdiegus trečių šalių papildinius. Rekomenduojama vengti nestandartinių papildinių naudojimo.

„MOODLE“ SPARTINANČIOSIOS ATMINTINĖS PODĖLIO DIEGIMAS

Pradedant 2.4 versija „Moodle“ atsirado spartinančiosios atminties valdiklis - „Moodle“ *Universal Cache* (MUC). Jo paskirtis – valdyti duomenis, sesijas bei užklausas talpinančius spartinančiosios atminties podėlius, diegti naujus podėlius ir nustatyti, kokio tipo duomenys juose bus laikomi.

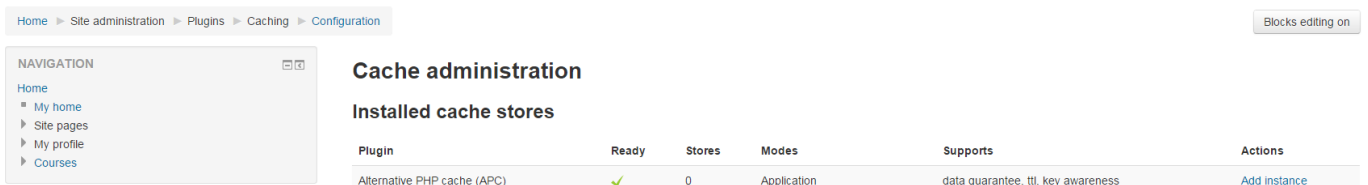
Įdiegus „Moodle“ nustatyta, kad duomenys laikomi diskinėje posistemėje organizuotuose spartinančiosios atminties podėliuose. Jei žiniatinklio serverio diskinė posistemė lėta, reikalinga spartinančiosios atminties podėlius perkelti į operatyviają atmintį. Tam tikslui naudojama *memcache*, *memcached*, APC ir kiti spartinančiosios atminties darbą organizuojanti programinė įranga. Žemiau pateikta APC diegimo instrukcija.

1. APCu paketo diegimas. Kiekvienoje sistemoje jis skirtingas. Pavyzdys „Linux“ Centos 7 sistemos komanda: `yum -y install php-pecl-apcu`
2. „Moodle“ papildinio Alternative PHP cache (APC) parsisiuntimas:
https://moodle.org/plugins/view/cachestore_apc
3. Alternative PHP cache (APC) papildinio diegimas. Tolesni veiksmai vykdomi „Moodle“ aplinkos „Moodle“ *Universal Cache* priemone. Pasirenkamas papildinio tipas – *Cache store* (1 pav.).



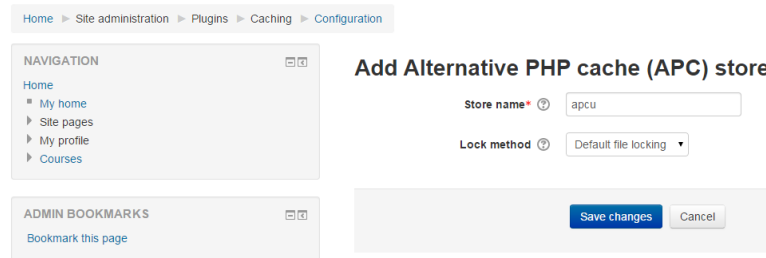
2 pav. „Moodle“ papildinio diegimas

4. Spartinančiosios atminties podėlio kūrimas. Prie Alternative PHP cache (APC) pasirinkti nuorodą *Add instance* (2 pav.).



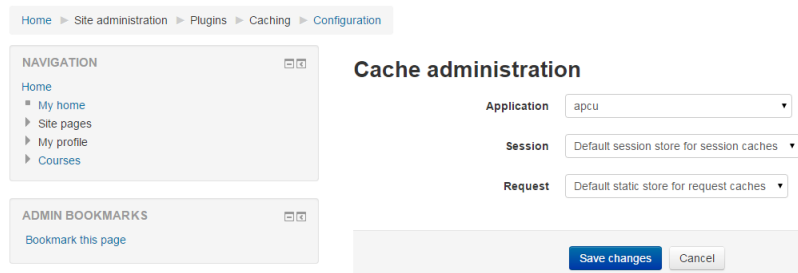
3 pav. Spartinančiosios atminties podėlio kūrimas

Nurodyti podėlio pavadinimą ir pasirinkti *Save Changes* mygtuką (3 pav.).



4 pav. Spartinančiosios atminties podėlio kūrimas

5. Spartinančiosios atminties podėlio įgalinimas. Nurodyti *Application* tipo duomenų saugojimą naujai įdiegtame APCu podėlyje (4 pav.).



5 pav. Spartinančiosios atminties podėlio įgalinimas.

Sisteminės aplinkos optimizavimas VMA „Moodle“ našumui bei greitaveikai

Remigijus Kutas

ANOTACIJA

Straipsnyje pristatomas darbas nagrinėjantis VMA „Moodle“ naudotojų užklausų trukmės priklausomybę nuo techninės bei sisteminės programinės įrangos konfigūracijos. Šio darbo tikslas yra testuoti ir patikrinti VMA „Moodle“ našumo bei greitaveikos pokyčius, taikant skirtingus sisteminės aplinkos komponentus ir jų konfigūravimo parametrus, ir pasiūlyti efektyvią VMA „Moodle“ naudotojų užklausų trukmės mažinimo metodiką. Straipsnyje nagrinėjamas poreikis užtikrinti pakankamą VMA „Moodle“ našumą bei greitaveiką. Analizuojami veiksniai, didinantys „Moodle“ aplinkos apkrovą: naudotojų skaičiaus augimas, apkrovos pikai. Aprašomos našumo ir greitaveikos testavimo metodikos, VMA „Moodle“ greitaveikos testavimo būdai. Aprašomi darbo eigoje atlikti bandymai, kurių tikslas – nustatyti įvairių sisteminės aplinkos komponentų įtaką VMA „Moodle“ našumui bei greitaveikai.

Ekspimentinio tyrimo rezultatai parodė, kad techninės įrangos pajėgumas – pagrindinis veiksnys, lemiantis VMA „Moodle“ našumą ir greitaveiką. „Moodle“ aptarnaujančios sistemos procesoriaus pajėgumo didinimas arba kelių sistemų apjungimas į klasterį užtikrina didžiausią „Moodle“ našumo bei greitaveikos augimą. Palyginti su procesoriaus našumu, diskinės posistemės greitis turi santykinai mažą įtaką „Moodle“ našumui bei greitaveikai. Pasiekus pakankamą operatyviosios atminties kiekį, tolesnis jos apimties didinimas „Moodle“ greitaveikai įtakos nedaro. Nagrinėjant sisteminę programinę įrangą ir jos įtaką „Moodle“ greitaveikai bei našumui nustatyta, kad didžiausią įtaką daro DBVS programinės įrangos parinkimas ir jos konfigūravimo modifikacijos.

ĮVADAS

Vienas svarbiausių veiksnių, pastaraisiais dešimtmečiais lemiančių esminius pokyčius formuojant studijų aplinką švietimo institucijose, yra intensyvus informacinių komunikacinių technologijų (IKT) diegimas. Tai lemia tiek bendros įvairių gyvenimo sričių kompiuterizavimo tendencijos, tiek mokymosi institucijų poreikiai modernizuoti studijų aplinką. Todėl šiandien daugelis Lietuvos švietimo institucijų turi ne tik kompiuterių tinklus, kompiuterizuotas auditorijas ar bibliotekas, bet ir virtualiąsias mokymosi aplinkas (VMA).

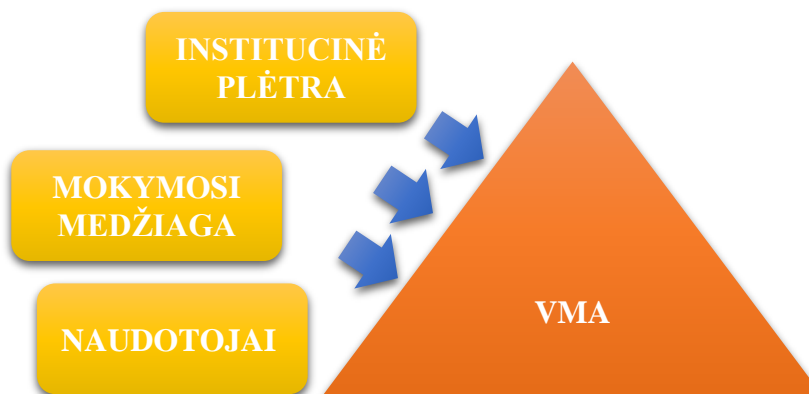
VMA taikymas šiandien būdingas ne tik nuotoliniam mokymuisi, bet dažnai yra svarbus ir tradiciniame mokyme ar studijose. Lietuvos švietimo institucijose daugelis dėstomų dalykų bent iš dalies

naudoja įvairias virtualiųjų mokymosi aplinkų galimybes: medžiagos talpinimą, žinių patikrinimo priemones. Šiandienos švietimo institucijose VMA naudotojų skaičius nuolatos auga. Atvirų švietimo iniciatyvų dėka institucijos naudotojų skaičius dažnai viršija institucijoje besimokančiųjų skaičių. Nuolat augantys aktyvių naudotojų skaičiai, mokymosi objektų kiekiai bei mokymosi medžiagos apimtys kelia naujus iššūkius institucijų serverinei infrastruktūrai bei VMA serverius prižiūrintiems administratoriams.

„Moodle“ – viena populiariausių virtualiųjų mokymosi aplinkų Lietuvoje, suprojektuota organizuoti mokymosi kursus tinkle ir atitinkanti kursų valdymo sistemų grupę. Tačiau sistema taip greitai plėtojama, kad jau dabar ją galima vadinti mokymosi valdymo sistema. Dėl naudojimo Lietuvoje populiarumo, VMA „Moodle“ našumas bei greیتaveika buvo pasirinkti pagrindiniu šio tyrimo objektu. Tyrimo eigoje atlikta įvairių sisteminės aplinkos komponentų įtakos VMA „Moodle“ našumui analizė. Tyrimo darbo rezultatai atskleidė efektyviausius „Moodle“ našumo bei greیتaveikos didinimo būdus. Jais remiantis parengtos sisteminės aplinkos konfigūravimo gairės, padėsiančios optimaliai išnaudoti techninius išteklius ir užtikrinti maksimalų VMA „Moodle“ darbo našumą bei greیتaveiką.

PROBLEMOS ANALIZĖ

VMA „Moodle“ programinė įranga šiandien – tai ne tik nuotolinių studijų organizavimo priemonė. Tai kasdieninio naudojimo priemonė įvairiuose mokymuose. Dėl patogaus mokymo proceso organizavimo, medžiagos talpinimo, žinių tikrinimo galimybių, „Moodle“ naudojama įvairiems taikymo scenarijams platesnėje auditorijoje. „Moodle“ naudojamas jau ne tik aukštosiose mokyklose ar specializuotuose nuotolinio mokymo centruose, bet ir tokiose institucijose, kaip vidurinio lavinimo mokyklos, suaugusiųjų švietimo centrai, nacionalinės valdžios įstaigos (pvz., LR Seimas ar VMI), verslo konsultacinės bendrovės ar didelės korporacijos.

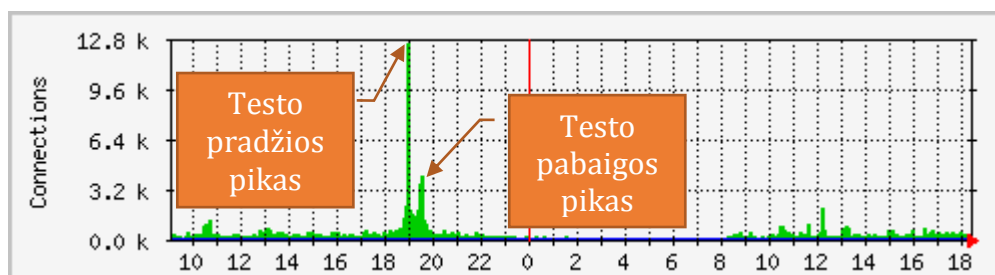


1 pav. Veiksniai, didinantys VMA apkrovą

Dėl platesnės VMA naudojimo skvarbos į tradicinius mokymosi metodus jos naudojimo apimtys nuolatos auga (Moodle: A Low-Cost Solution for Successful e-Learning, 2015). Naudojimas auga vykstant švietimo institucijų plėtrai. Augimą skatina neformaliojo švietimo, atvirų švietimo iniciatyvų paplitimas.

Išvardyti veiksniai didina naudotojų bei mokymosi objektų skaičių (Capterra, 2014; Lambda Solutions, 2013). Tai neišvengiamai didina reikalavimus techninės bei programinės įrangos galimybėms aptarnauti išaugusius naudotojų poreikius. Svarbiausi veiksniai, didinantys VMA apkrovą, pateikti 1 pav.

VMA „Moodle“ aplinkos apkrova nėra tolygi. Mokymosi proceso eigoje „Moodle“ gali patirti apkrovos padidėjimo pikus. Apkrovos pikai – tai staigūs „Moodle“ apkrovos padidėjimai, atsirandantys dėl mokymosi proceso ypatumų. Apkrovos intensyvumas piko metu nuo įprastinės apkrovos gali skirtis dešimtis kartų. 2 pav. pateiktas pavyzdys, iliustruojantis, kaip kinta HTTP užklausų skaičius, kai 300 studentų laiko žinių patikrinimo testą. Testo pradžios laikas fiksuotas nuo 19:00 iki 19:30. Testui atlikti skirta 30 min.



2 pav. HTTP užklausų skaičiaus kaita studentų žinių patikros metu

Išskirtini du apkrovos padidėjimo pikai. Pirmasis susietas su testo pradžia – testo aktyvavimo pradžios laikas fiksuotas, o jo aktyvumo periodas pakankamai trumpas. Studentai nekantriai laukia testavimo pradžios, nuolat atnaujindami VMA „Moodle“ puslapį su nuoroda į testą. Sistemai aktyvavus testą didžioji studentų dalis pradeda atlikti testą pirmąją jo aktyvavimo minutę. Pirmasis apkrovos pikas pasižymėjo 8,5 karto didesniu užklausų skaičiumi palyginti su užklausų skaičiumi testo laikymo eigoje. Antrasis pikas sukeltas fiksuotos testo trukmės – vienu metu pradėję testo laikymą jie ir baigia panašiu laiku išsekus testui skirtam laikui. Kadangi klausimų atsakymams studentai užtrunka skirtingą laiko tarpą, tai antrasis apkrovos pikas (2,5 karto) nėra toks ryškus ir labiau pasiskirstęs laike.



3 pav. Parodytų puslapių skaičiaus kaita aukštosios mokyklos mokymosi semestro eigoje

Vertinant ilgos trukmės laiko periodus, apkrovos pikai susiję su mokymosi periodais: semestro pradžia, pabaiga, egzaminų sesija. Pavyzdžiui, aukštosios mokyklos studentai intensyviau naudojami VMA

tarpinių semestro atsiskaitymų, semestro pabaigos ar sesijos metu (3 pav.) (Xin Guo, Qing Shi, Danjue Zhang, 2013).

Mokymosi institucijų plėtra, VMA panaudojimo atvejų, naudotojų skaičiaus, mokymosi medžiagos apimties augimas didina nuolatinę VMA tenkančią apkrovą. Todėl apkrovos piko metu atsiranda galimybė sistemos darbas sutrikimams. Įvairūs VMA darbo sutrikimai, ypatingai pasitaikantys žinių patikrinimo metu nepageidaujami. Todėl VMA aptarnaujančios sistemos turi būti parengtos dirbti su apkrovomis dešimtis kartų viršijančias vidutines. Tam būtina iširti „Moodle“ našumo bei greitaveikos priklausomybę nuo techninės bei programinės įrangos.

ŽINIATINKLIO PROGRAMŲ SPARTINIMO METODAI IR PRIEMONĖS

Virtualioji mokymosi aplinka „Moodle“ – modulinė PHP ir MySQL pagrindu sukurta sistema. Šiuo metu „Moodle“ sudaro apie 7000 PHP kodo failų. Nagrinėjant būdus optimizuoti „Moodle“ greitaveiką, pritaikant dideliems naudotojų srautams aptarnauti, svarbiausias dėmesys turi būti skiriamas į PHP programoms būdingas greitaveikos problemas. Todėl „Moodle“ funkcionavimui būtinas sisteminių komponentų derinys, būdingas tipinei PHP programai vykdyti. Tai techninės įrangos, operacinės sistemos, žiniatinklio (WWW) serverio, PHP interpretatoriaus bei duomenų bazių valdymo sistemos derinys. Veiksniai, darantys įtaką VMA „Moodle“ greitaveikai yra iš esmės tie patys, kaip ir kiekvienai žiniatinklio programai parašytai PHP programavimo kalba. Žiniatinklio serverio darbo optimizavime reiktų susitelkti ties veiksniais, kurie turės didžiausią reikšmę naudotojui. Pavyzdžiui, jei yra daugiau naudotojų, intensyviau naršančių statinius duomenis nei naudojančių komponentus, lemiančius duomenų bazės greitaveiką, rekomenduotina spartinti žiniatinklio serverį (Moodle performance recommendations, 2014).

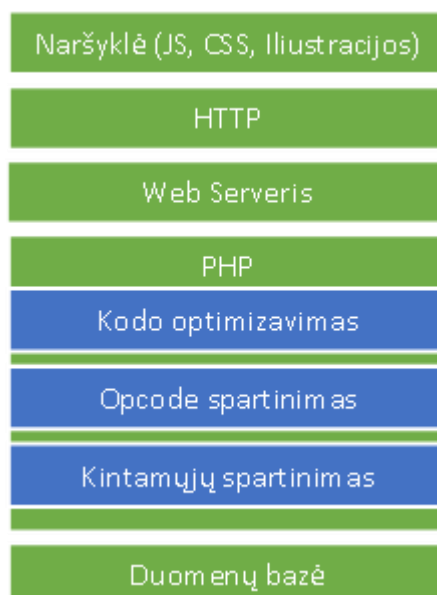
„Moodle“ architektūra sukurta atsižvelgiant į potencialų poreikį aptarnauti daug naudotojų. Didelį naudotojų skaičių aptarnaujančios svetainės paprastai atskiria žiniatinklio serverį ir duomenų bazių aptarnavimo serverį į skirtingus fizinius serverius. Taipogi, galimas VMA „Moodle“ diegimas, naudojantis apkrovos balansavimo metodus, kai naudojamas daugiau kaip vienas žiniatinklio serveris. Atskiras žiniatinklio serveris prijungiamas prie tos pačios duomenų bazės ir susietas su ta pačia „Moodle“ failų talpykla bei laikinų objektų podėliu. Bet dažniausiai VMA „Moodle“ darbo greitaveikai užtikrinti taikomas tik žiniatinklio ir DBVS serverių atskyrimas. Esant nepakankamai DBVS greitaveikai, panašiai kaip žiniatinklio serveriai, DBVS serveriai gali būti apjungiami į serverių grupę, pvz.: MySQL klasterį („Moodle“ performance recommendations, 2014).

Vienas iš pagrindinių veiksnių, lemiančių VMA „Moodle“ naudotojo užklausų trukmę yra PHP interpretatoriaus darbo greitaveika. PHP yra labai populiari programavimo kalba, naudojama daugelyje populiarių interneto svetainių. PHP kodo interpretavimo greitis gali labai keistis, atsižvelgiant į PHP interpretatoriaus versiją, žiniatinklio serverio sisteminės aplinkos bei PHP kodo sudėtingumą. Todėl

nustatant PHP greitaveikos problemas reikia identifikuoti taškus, turinčius neigiamą įtaką greitaveiką. Dažnai PHP kodo greitaveikos problemos susijusios ne su jo optimizavimu, o su duomenų baze ar tinklu. PHP kodo vykdymą spartina PHP *bytecode* podėlių, paskirstytos atminties objektų podėlių naudojimas. Šios priemonės spartina PHP programų veikimą, kaupdamos jau sukompiliuoto PHP kodo fragmentus ar duomenų objektų kopijas laikinos atminties podėlyje. PHP naudoja atminties buferį, kad sukauptų duomenis, kuriuos programa ruošia išvedimui. Šis buferis gali turėti neigiamos įtakos laikui, per kurį naudotojui atverčiamas žiniatinklio puslapis. Pasak E. Higgins, galima padaryti tam tikrus PHP interpretatoriaus nustatymų pakeitimus, kurie privers PHP greičiau atnaujinti išvedimo buferius, ir svetainė naudotojams taps greitesnė (Eric Higgins, 2014).

Darbą pradėti galima tik apibrėžus potencialias PHP programos greitaveikos problemas ir jų sprendimo galimybes bei išnagrinėjus, ar galimi šių problemų sprendinių taikymai VMA „Moodle“ atžvilgiu. Nes nekeičiant programos turinio (išeities kodo), tik dalį greitaveikos optimizavimo sprendimų galima įgyvendinti. Tuo tarpu, net 80% žiniatinklio puslapio krovimo laiko priklauso nuo jo turinio (King, 2008). Pagrindinis dėmesys literatūroje skiriamas žiniatinklio puslapio optimizavimo problemoms naršyklės lygmenyje spręsti. Literatūroje minimas didelis skaičius, veiksmų darančių įtaką greitaveikai (Armando Padilla, 2010; King, 2008; Padilla & Hawkins, 2010; Moodle performance recommendations, 2014; Sheltren, Newton, & Catchpole, 2014; Souders, 2007). Žemiau pateikta iliustracija išvardija komponentus, sudarančius žiniatinklio programos ekosistemą bei galinčius lemti VMA „Moodle“ greitaveiką. Daugelis PHP programų įgyvendina savo sąsają su naudotoju žiniatinklio naršyklės pagalba. Sąsajai užtikrinti naudojamos HTML, JS, CSS, Flash bei kitos žiniatinklio programos. Naršyklėje atvaizduota programos sąsaja HTTP protokolu per žiniatinklio serverį komunikuoja su programos PHP kodu, kuris savo ruožtu dirba su duomenimis iš išorinio šaltinio – duomenų bazės.

Greitaveika priklauso nuo išvardytų PHP programos darbą įgyvendinančių komponentų (4 pav.). Todėl nagrinėjant problemas, susijusias su VMA „Moodle“ greitaveika bei būdus kaip tas problemas spręsti, reikia nagrinėti tipinės PHP programos komponentus.

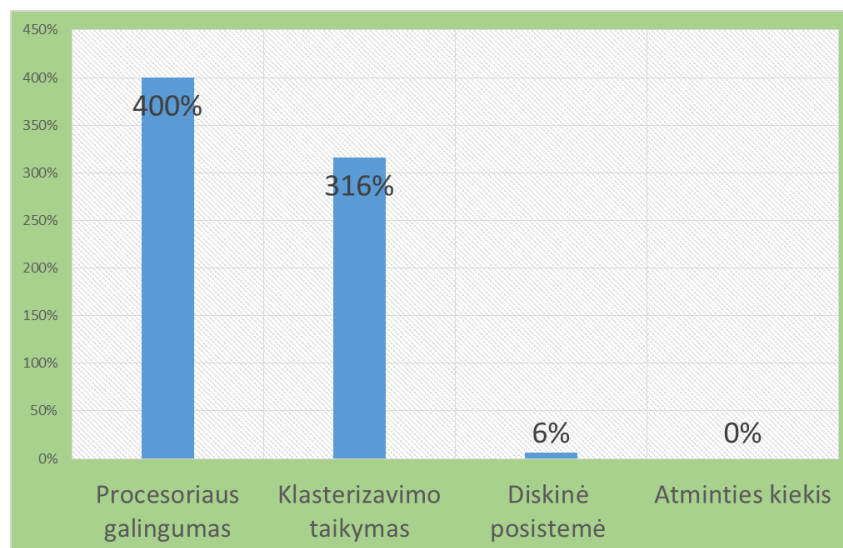


4 pav. PHP programos komponentai (Armando Padilla, 2010)

Pagrindiniai greitaveikos spartinimo aspektai, susiję su PHP interpretavimo procesu būtų šie: PHP kodo optimizavimas, PHP mašininio kodo spartinimas, duomenų mainų su išoriniais duomenų šaltiniais spartinimas. Tikėtina, kad VMA „Moodle“ PHP kodas pakankamai gerai parašytas. Optimizuoti „Moodle“ kodą labiau nei tai padarė kūrėjai – mažai tikėtina. Tuo tarpu operacijų su PHP mašininio kodu bei duomenų objektais spartinimas talpinant juos spartinančiose atmintinėse „Moodle“ yra būtinas.

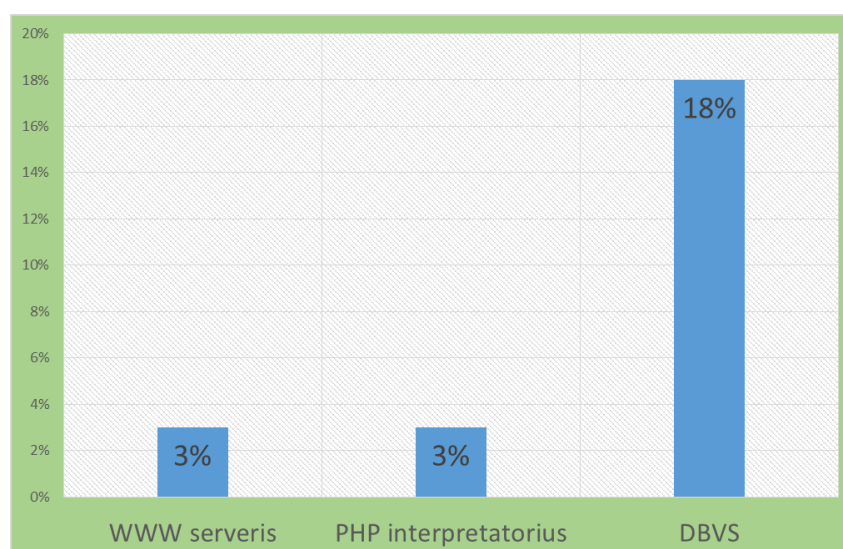
EKSPERIMENTINIO TYRIMO APIBENDRINIMAS

„Moodle“ funkcionavimui būtinas sisteminių komponentų derinys, būdingas tipinėms PHP programoms. Tai – techninės įrangos, operacinės sistemos, žiniatinklio serverio, PHP interpretatoriaus bei duomenų bazių valdymo sistemos derinys. Literatūroje pateikiamas platus spektras PHP programų spartinimo metodikų (Armando Padilla, 2010; Eric Higgins, 2014; King, 2008; Padilla & Hawkins, 2010; Moodle performance recommendations, 2014; Sheltren, Newton, & Catchpole, 2014; Souders, 2007; Xin Guo, Qing Shi, Danjue Zhang, 2013). Šios spartinimo technikos apima visas „Moodle“ funkcionavimui būtinas techninės bei programinės įrangos komponentus. Bet dėl VMA „Moodle“ programinės įrangos specifikos daug literatūroje aprašomų žiniatinklio programų spartinimo technikų arba neefektyvios, arba turi neigiamą poveikį (Armando Padilla, 2010; Eric Higgins, 2014; King, 2008; Padilla & Hawkins, 2010; Moodle performance recommendations, 2014; Sheltren, Newton, & Catchpole, 2014; Souders, 2007; Xin Guo, Qing Shi, Danjue Zhang, 2013). Pavyzdžiui, tokios tipinės rekomenduojamos priemonės, kaip turinio kompresija, neigiamai veikia „Moodle“ našumą bei greitaveiką. Dalies spartinimo technikų taikymas negalimas dėl su jomis susijusio poreikio keisti „Moodle“ programinę įrangą.



5 pav. Testų metu fiksuoti maksimalūs našumo bei greیتaveikos padidėjimai, susiję su techninės įrangos konfigūracijos pokyčiais

Techninė įrangos pajėgumas pagrindinis veiksnys, lemiantis VMA „Moodle“ našumą bei greیتaveiką (5 pav.). „Moodle“ aptarnaujančios sistemos procesoriaus pajėgumo didinimas arba kelių sistemų apjungimas į klasterį duoda didžiausią „Moodle“ našumo bei greیتaveikos prieauglį. Palyginus su procesoriaus našumu diskinės posistemės greitis turi santykinai mažą įtaką „Moodle“ našumui bei greیتaveikai. Esant pakankamam operatyviosios atminties kiekiui tolesnis jos apimties didinimas „Moodle“ greیتaveikai įtakos nedaro. Operatyviosios atminties apimties didinimas prasmingas tada, kai dėl jos trūkumo naudojama diskinė posistemė.



6 pav. Testų metu fiksuoti maksimalūs našumo bei greیتaveikos padidėjimai, susiję su programinės įrangos konfigūracijos pokyčiais

„Moodle“ programinės įrangos specifika neleidžia efektyviai taikyti spartinimo metodų, susijusių su žiniatinklio serverio programinės įrangos parinkimu. „Moodle“ greیتaveika labai menkai priklauso nuo

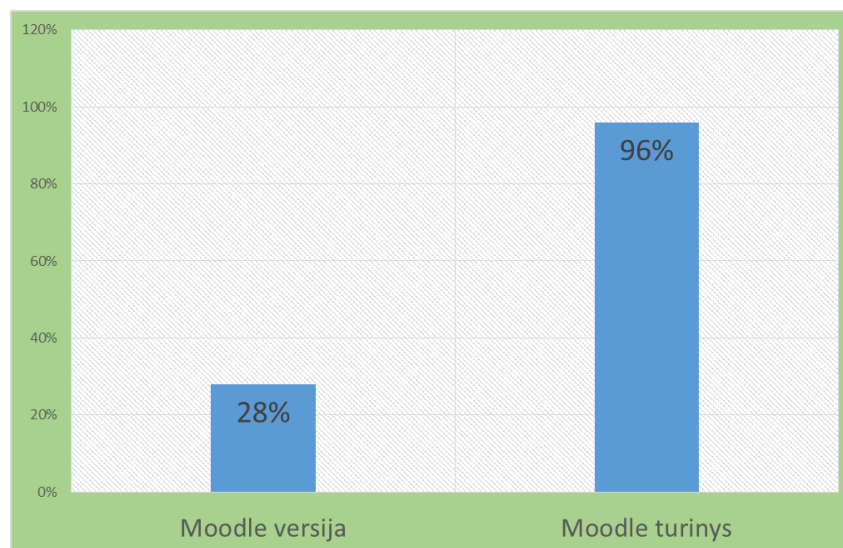
žiniatinklio serverio gebėjimo greitai pateikti statinį turinį, nuo jo operatyviosios atminties sunaudojimo ar maksimalaus aptarnaujamų klientų skaičiaus (6 pav.). Literatūros šaltiniai rekomenduoja žiniatinklio serverio programinę įrangą keisti įranga, vienam klientui sunaudojančia mažiau operatyviosios atminties. Tai ženkliai padidina maksimalų galimą aptarnauti klientų skaičių mašinos, turinčiose mažą operatyviosios atminties kiekį (Armando Padilla, 2010). Kitose rekomendacijose patariama rinktis žiniatinklio serverio programinę įrangą, sparčiau pateikiančią statinius puslapio elementus: HTML, grafiką ir panašiai (Armando Padilla, 2010). VMA „Moodle“ atveju, šios priemonės neturi teigiamos įtakos. Didžioji „Moodle“ naudotojui pateikiamo turinio dalis generuojama dinamiškai. O naudotojų skaičius santykinai nėra didelis. Todėl žiniatinklio serverio programinės įrangos keitimas arba konfigūracijos parametrų modifikavimas neefektyvus ir greičiau gali privesti prie „Moodle“ našumo bei greitimeikos sumažėjimo.

„Moodle“ našumą bei greitimeiką nulemia PHP interpretatoriaus turinio generavimo sparta. Esant įgalintam PHP mašininio kodo spartinimui kiti PHP interpretatoriaus spartinimo metodai suteikia santykinai nedidelį „Moodle“ darbo pagreitejimą – iki 3%. Kadangi mašininio kodo spartinimas šiandieninėse PHP distribucijose įgalintas, pagal nutylėjimą reiktų laikyti, kad PHP interpretatoriaus modifikavimas siekiant pagerinti „Moodle“ darbą taip pat netikslingas.

Nagrinėjant sisteminę programinę įrangą ir jos įtaką „Moodle“ greitimeikai DBVS programinės įrangos parinkimas bei jos konfigūravimo modifikacijos efektyviausios didinant „Moodle“ maksimalų našumą bei greitimeiką (žr. 42 pav.). DBVS negali veikti optimaliai esant tipiniams DBVS konfigūravimo parametrams. DBVS konfigūravimo parametrai derinami tiek prie techninės įrangos galimybių, tiek prie operuojamų duomenų specifikos. Todėl DBVS priderinimas prie esamos situacijos suteikia didžiausią teigiamą poveikį „Moodle“ našumui bei greitimeikai.

HTML turinio optimizavimas, informacijos suspaudimas, CSS optimizavimas, grafikos optimizavimas skirti sumažinti siunčiamo turinio apimtį, tuo pagreitinant puslapio užkrovimo greitį kliento pusėje. Bet „Moodle“ atveju, problemos susijusios su dinaminio turinio parengimo siųsti laiku, o ne su jo siuntimo laiku. Todėl tokios priemonės, kaip GZIP turinio spaudimas, nepagreitina „Moodle“ turinio užkrovimo greičio, bet tą laiką prailgina, nes turinio spaudimui procesorius apkraunamas papildomomis užduotimis, trukdančiomis „Moodle“ turinio generavimui.

Operacinės sistemos įtaka tyrimo metu nebuvo galutinai nustatyta, todėl išvados dėl operacinės sistemos įtakos „Moodle“ našumui bei greitimeikai šiame darbe nepateikiamos.



7 pav. Testų metu fiksuotas maksimalus našumo bei greیتaveikos neigiamas pokytis, susijęs su „Moodle“ versija bei turiniu

Yra daug veiksnių, negatyviai veikiančių našumą bei greیتaveiką, susijusių su „Moodle“ aplinkos programiniu kodu bei „Moodle“ teikiamu mokymosi turiniu. Tai „Moodle“ programinės įrangos versijų kaita, papildinių diegimas, mokymosi objektų gausa. „Moodle“ nuolat tobulinamas, įgalinami nauji priedai, pridedamas papildomas funkcionalumas. Visa tai didina „Moodle“ programinio kodo apimtį. Jo vykdymas tampa vis labiau komplikuoju. Naujų „Moodle“ programinės įrangos versijų darbas pasižymi našumo bei greیتaveikos mažėjimu (7 pav.). Kitas, ypatingai neigiamai „Moodle“ darbo greitį veikiantis veiksnys yra mokymosi objektų skaičius. Didelius kiekius naudotojų, kursų ar užduočių talpinantis VMA „Moodle“ pasižymi ženkliai mažesne greیتaveika, nei talpinantis mažesnius mokymosi objektų skaičius.

IŠVADOS

Egzistuoja daug žiniatinklio programų spartinimo technikų, tačiau jų praktinis pritaikymas priklauso nuo VMA „Moodle“ programinės įrangos specifikos. Techninės įrangos pajėgumas - pagrindinis veiksnys, lemiantis VMA „Moodle“ našumą bei greitaveiką. Išvados, susijusios su VMA „Moodle“ aptarnaujančiais techniniais komponentais:

- žiniatinklio serverio, aptarnaujančio VMA „Moodle“, procesoriaus pajėgumo didinimas – pagrindinis būdas pagreitinti „Moodle“ aplinkos darbą;
- esant nepakankamam procesoriaus pajėgumui efektyvu apjungti keletą lygiagrečiai dirbančių žiniatinklio serverių į klasterį;
- operacinės atminties kiekis turi būti pakankamas aptarnauti maksimalų žiniatinklio serverio procesoriaus aptarnaujamų naudotojų skaičių. Tolesnis operacinės atminties apimties didinimas nekeičia „Moodle“ greitaveikos;
- „Moodle“ duomenų talpinimas NAS tipo duomenų saugykloje gali neigiamai veikti „Moodle“ našumą bei greitaveiką. NAS neigiama įtaka kompensuojama perkeliant „Moodle“ spartinančiosios atmintinės podėlius į DBVS bei operatyviają atmintį (2 priedas).

Šiandien dauguma „Moodle“ darbui reikalingų sisteminės programinės įrangos komponentų dirba optimaliai. Gamintojo parinkti sisteminės programinės įrangos konfigūravimo parametrai užtikrina optimalų darbo našumą bei greitaveiką. Sisteminės programinės įrangos komponentų tarpe išskirtinas DBVS poveikis „Moodle“ darbo greičiui. DBVS programinės įrangos parinkimas bei jos konfigūravimo modifikacijos efektyviausiai didina „Moodle“ našumą bei greitaveiką.

Pagrindiniai veiksniai, neigiamai veikiantys našumą bei greitaveiką ir susiję su „Moodle“ aplinkos programiniu kodu bei „Moodle“ teikiamu mokymosi turiniu – tai „Moodle“ programinės įrangos versijų kaita, papildinių diegimas, mokymosi objektų gausa.