TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Marek Ott 206163IAAB

Dünaamiliselt seadistatud seire ning häireteavitused kettamahu ja mälukasutuse näitel

Bakalaureusetöö

Juhendaja: Lauri Anton
Bakalaureus

Tallinn 2023

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Marek Ott

24.04.2023

Annotatsioon

Töös käsitletakse mälukasutuse ning kettamahtude seiret kasutades dünaamilist seadistust. Eesmärgiks on universaalne seireraamistik, mida saab modifitseerimata kasutusele võrra sõltumata seireobjekti spetsifikatsioonidest.

Töö raames testitakse tulevaste väärtuste prognoosimist ning testimise tulemuste põhjal luuakse seireraamistik. Seireraamistik võetakse kasutusele toodangukeskkonnas ja kogutakse selle efektiivsuse kohta andmeid. Lõplike andmete põhjal tehakse järeldused, kas ja millisel kujul saaks valminud raamistikku kasutada seire teostamisel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 23 leheküljel, 4 peatükki, 11 joonist, 3 tabelit.

Abstract

Dynamic Monitoring and Notifications for Disk and Memory Usage

This thesis covers creation of framework for dynamic monitoring of disk and memory usage.

Creation of framework starts with testing phase, where different monitoring items are created. Those items are used for data collection and prediction of future values. Testing phase of items lasts until satisfactory results are achieved. Based on those future values, triggers are configured to alert when a monitored resource is predicted to be used up. Triggers are tested for alert creation times and accuracy.

After the completion of testing phase, the created framework will be activated in a live environment with 897 monitored hosts. Framework will be activated beside static framework and differences in trigger activation times and accuracy in trigger activation necessity will be analysed. A conclusion will be made if created monitoring items are suitable for monitoring disk and memory usage. A comparison between dynamic and static monitoring effectiveness will be made.

Related topics like resource usage and necessary configuration updates for dynamic monitoring are referred.

The thesis is in Estonian and contains 23 pages of text, 4 chapters, 11 figures, 3 tables.

Lühendite ja mõistete sõnastik

Arvutatud mõõtepunkt	Zabbix rakenduse kirje, mille sisse on võimalik kirjutada tehteid ning kombineerida erinevaid andmeid
Forecast	Funktsioon mis prognoosib varasemate väärtuste põhjal tulevasi väärtusi
GPL	General Public License, vaba tarkvara litsents
Mõõtepunkt	Zabbix rakenduse kirje ühe kogutava andmepunkti kohta
NVPS	New values per second, uute andmete hulk sekundi jooksul
RAM	Random Access Memory, muutmälu
Season	Vaatlusperiood
TCP	Transmission Control Protocol, transpordiprotokoll
Timeleft	Funktsioon mis kalkuleerib varasemate andmete põhjal seadistatud piirmäärani jõudmise aja sekundites

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Efektiivne seire on tehnoloogiaettevõtte eduka toimimise üks alustaladest. Käesoleval ajal on kõrgkäideldavus ning rikkekindlus IT infrastruktuuri arenduste fookuses. Arenduste edukust saab kõige objektiivsemalt valideerida seireinfole tuginedes. Samuti on IT teenuste tarbijate eelduseks teenuste püsiv kättesaadavus. Tõrgete tuvastamine ning käideldavusega seotud lepingulised lubadused oleksid ilma eduka seireta teostamatud.

Praegune seire praktika näeb ette seda, et kogutud andmetele määratakse fikseeritud piirväärtused. Piirväärtuse ületamisel algatatakse ressursi eest vastutava osapoole teavitamise protsess. Antud lähenemine ei sisalda andmete analüüsi. Selline metoodika välistab proaktiivsed teavitused ebastandardsete näitajate puhul, mis ei ületa veel piirväärtusi, aga on siiski indikaatoriks kõrvalekaldest süsteemi tavapärasest käitumisest. Varajane teavitus ebastandardsetest seirenäitajatest, annaks rohkem aega olukorrale reageerimiseks ning aitaks vähendada tõenäosust, et seiratava süsteemi töös tekib seisak. Andmete võrdlemisel vaid piirväärtustega, aga mitte süsteemi ressursside kasutamise trendiga, tekib teavitamise protsessi valealarme. Valealarmide tekitajaks on ajutised piirväärtuse ületamised, mille järgselt naaseb ressursi kasutus piiridesse, mida peetakse fikseeritud piirväärtuse järgi normaalseks.

1.1 Eesmärk

Lõputöö eesmärgiks on dünaamilise seire funktsionaalsuste testimine Zabbix rakenduses. Luuakse sobivaid funktsioone kasutades seireraamistik, analüüsitakse selle lahenduse võimekust probleemide tuvastamisel ja seeläbi teavituste tekitamisel ning valealarmide vähendamisel. Eesmärgi saavutamiseks teostatakse esmane seireraamistiku loomine ning erinevate metoodikate testimine virtuaalmasinatel loodud testkeskkonnas. Testimise käigus loodud seireraamistik rakendatakse toodangukeskkonnas 897 serveri peal. Selles valimis saavad olema nii füüsilised kui ka virtuaalsed serverid.

Raamistikuga testitakse kahe näitaja seiret: kettamahu täituvus ja mälukasutus. Dünaamiline seire pannakse tööle staatilise seireraamistikuga samade objektide peal ning

neid analüüsitakse kahe näitaja järgi: valealarmide kogus ja alarmi tekitamise kiirus. Töö lõpus võrreldakse, kumma meetodi puhul tuvastati alarmi tõstatamist vajav olukord varem ning kumb meetod tekitas vähem alarme olukordades, mis ei vajanud inimese sekkumist. Kettamahtude seire puhul loetakse lisaks kokku proaktiivsed teavitused mahu peatsest lõppemisest. Analüüsi käigus tuvastatakse kui suur kogus teavitustest olid korrektseid ja vajasid inimese sekkumist ning kui paljud olid valealarmid. Kuna proaktiivset võimekust staatilisel seirel ei ole, siis seda arvu ei saa kahe meetodi puhul võrrelda, küll aga saame siit vajalikku infot varajase teavituse efektiivsuse kohta.

2 Seiresüsteemid

Seiresüsteemide peamine eesmärk on andmete kogumine ning vajadusel alarmide tekitamine. Kogutud andmed salvestatakse andmebaasi ning neid võrreldakse reeglitega, mis on antud mõõtepunktile seatud. Näitajate ning neile seatud reeglite võrdluse tulemusel selgub, kas andmeid peetakse normaalseks või tuleb neile mingil moel reageerida, näiteks algatada teavitusprotsess. Üks levinud teavituse viis on seireobjekti eest vastutajat meili teel teavitada.

Seireobjekt võib olla mistahes seade, mis oskab mingil moel enda kohta andmeid väljastada. Levinumad seireobjektid on nii füüsilised kui ka virtuaalsed serverid, tööjaamad, andmebaasid, rakendused ja võrguseadmed nagu näiteks kommutaator või marsruuter. Seire jaoks saab koguda kõiki andmeid, mida kontrollitav objekt suudab väljastada. Siiski on hea praktika koguda vaid andmeid, mille põhjal saab seireobjekti seisukorra või mõne muu olulise aspekti kohta vajalikku infot, kuna iga objekt võib suure regulaarsusega välja anda sadu või isegi tuhandeid andmepunkte, mis tekitab seiresüsteemis suurt koormust ning palju sissekandeid andmebaasi [1].

Edukas seire loob nähtavuse IT süsteemide seesmistesse protsessidesse. Sellele informatsioonile tuginedes saab reageerida kiireloomulistele intsidentidele, otsida intsidentide juurpõhjuseid ning seoseid erinevate komponentide tõrgete vahel. Seire käigus kogutud andmeid kasutatakse ressursside kasutuse hindamiseks ning vajaduste kaardistamiseks tulevasi süsteeme planeerides. Seireandmeid kasutatakse ka arendustööle järgneva analüüsi puhul, võrreldes relevantseid näitajaid enne ja peale uue arenduse töökeskkonda rakendamist [2].

2.1 Seirepraktikad tänapäeval

Seire käigus kogutavad andmed võib liigitada kolme kategooriasse [3]. Esimeseks neist on kumulatiivsed andmed. Siia alla kuuluvad andmed, mille väärtust aja jooksul suurendatakse nagu näiteks loendurid, mida hakatakse suurendama alates süsteemi käivitumise hetkest.

Teine ja ühtlasi levinuim andmetüüp on mööduva olemusega. See tähendab, et varasemad väärtused kirjutatakse üle uutega. Seda liiki andmetüübi alla kuulub vaba mälumahu mõõt, mis muutub suure regulaarsusega ning vana väärtus muutub ebaoluliseks.

Kolmas andmetüüp on muutumatud andmed. Seda tüüpi väärtused võivad näidata erinevate süsteemi elementide versioone ja tüüpe.

Levinud praktika on kasutada seirerakendustega kaasas olevaid malle. Need mallid on aastate jooksul optimeeritud ning heaks alguspunktiks, millele saab enda süsteemi jaoks spetsiifilised kontrollid juurde lisada. Mallid on jagatud erinevat tüüpi seireobjektide ning seiremeetodite järgi. Lisaks on mallid reeglina jagatud kindlat tüüpi andmeid koguma, et oleks võimalikult lihtne vaid vajalikud andmed süsteemist kätte saada, ilma üleliigset koormust tekitamata.

Joonisel 1 on kuvatud mõõtepunktid Zabbix rakendusega kaasas olevast mallist „Linux memory by Zabbix agent“. See mall on mõeldud Linux operatsioonisüsteemis mälukasutuse kohta andmeid koguma. Joonisel kuvatakse interval veerul info, et andmeid küsitakse korra minutis. Osadel mõõtepunktidel on triggers veerul sissekanne, mis tähendab, et antud väärtust kasutatakse alarmide tõstatamiseks.

Name ▲	Triggers	Key	Interval
Available memory	Triggers 1	vm.memory.size[available]	1m
Available memory in %		vm.memory.size[pavailable]	1m
Free swap space		system.swap.size[,free]	1m
Free swap space in %	Triggers 1	system.swap.size[,pfree]	1m
Available memory in %: Memory utilization	Triggers 1	vm.memory.utilization	
Total memory	Triggers 1	vm.memory.size[total]	1m
Total swap space	Triggers 1	system.swap.size[,total]	1m

Joonis 1. Zabbix mall, Linux operatsioonisüsteemi mälukasutuse seireks.

Joonisel 2 on kuvatud triger kirjed, mis käivitavad teavitusprotsessi. Trigeri sisuks on loogiline tehe. Joonisel 2 on kõik kolm kirjet loodud sarnase loogikaga. Nendes võrreldakse viimase viie minuti andmeid makrodega, milles on defineeritud piirväärused antud trigerite jaoks. Makrodes olevad piirväärtused on malliga kaasas. Selline lahendus ei ole efektiivne, kuna seadistus on üldine ning ei võta arvesse konkreetse süsteemi

eripärasid. Iga erijuhtumi korral on vaja makro sisu ära muuta, vastasel juhul võib jääda oluline teavitus saamata või vastupidi tekitatakse liiga palju alarme.

Severity	Name ▲	Operational data	Expression
Average	High memory utilization		**min(**/Linux memory by Zabbix agent/vm.memory.utilization,5m**)**>{$MEMORY.UTIL.MAX}
Warning	High swap space usage	Free: {ITEM.LASTVALUE1}, total: {ITEM.LASTVALUE2}	**max(**/Linux memory by Zabbix agent/system.swap.size[,pfree],5m**)**<{$SWAP.PFREE.MIN.WARN} and **last(**/Linux memory by Zabbix agent/system.swap.size[,total]**)**>0
Average	Lack of available memory	Available: {ITEM.LASTVALUE1}, total: {ITEM.LASTVALUE2}	**max(**/Linux memory by Zabbix agent/vm.memory.size[available],5m**)**<{$MEMORY.AVAILABLE.MIN} and **last(**/Linux memory by Zabbix agent/vm.memory.size[total]**)**>0

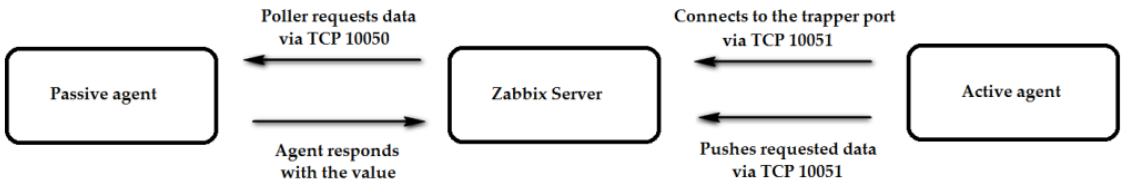
Joonis 2. Zabbix mall, triger kirjed Linux operatsioonisüsteemi mälukasutuse seireks.

2.2 Zabbix monitooringu tarkvara

Käesolevas diplomitöös kasutatakse seire teostamiseks Zabbix rakendust. Tegemist on vabavaralise rakendusega, millele kehtib *GPL* (*General Public License*) litsentsi versioon number 2 [4]. Zabbix rakendus osutus valituks, kuna see on kasutusel ettevõttes, mille serverite peal lõplik seireraamistik kasutusele võetakse. Sellisel juhul saab antud diplomitöö skoopi paremini suunata, keskendudes vaid raamistiku loomisele. Zabbix rakendusel on aktiivne uuendustsükkel, mille käigus tuleb uus väljalase vähemalt kahel korral aastas [5].

Zabbix on sobiv valik seatud eesmärgi jaoks, kuna sel rakendusel on võimekus trende kalkuleerida ning selles on arvutatud mõõtepunktide näol elemendid, mille sisse on võimalik kirjutada vajalikke tehteid ning kombineerida erinevaid andmeid. Viimane on eriti vajalik uudsete lähenemiste katsetamiseks, kuna see annab vabad käed vajalikke andmeid juurde luua ning seireraamistiku loomisel ei pea sõltuma vaid nendest andmepunktidest, mida rakendus oskab koguda või seireobjekt suudab väljastada.

Zabbix kogub andmeid agentprogrammiga Zabbix Agent. See programm vastutab andmete kogumise ja edastamise eest. Agent saab toimida aktiivses ja passiivses režiimis. Aktiivses režiimis küsib agent serveri käest, milliseid andmeid koguda on vaja ning saadab neid kokkulepitud sagedusega serveri või puhverserveri suunal. Passiivses režiimis ootab agent andmeid küsivat päringut. Serveri või puhverserveri poolt saadetavas päringus sisaldub info, milliseid andmeid agendi käest küsitakse ning agent saadab vastuseks päritud andmepunktid. Joonisel 3 on kuvatud agendi ja serveri vaheline suhtlus [6].



Joonis 3. Aktiivse ja passiivse agendi suhtlus serveriga TCP portide 10050 ja 10051 vahendusel.

2.3 Dünaamiline seire

Dünaamilise seire all mõeldakse raamistikku, milles pole defineeritud kindlaid piirväärtusi. Selleks, et piirväärtusi seadistamata ei kaotaks seire enda võimekust probleeme tuvastada, on võimalik kasutada mitmeid erinevaid lähenemisi. Üheks sobivaks meetodiks praeguste andmete võrdlemine varasema relevantse ajaaknaga. Näitena sobib olukord, kus serveri viimased andmed viitavad olulisele koormuse tõusule. Selgitamaks kas antud olukord on ebatavaline, tuleks praeguseid andmeid võrrelda 24 tundi tagasi kogutud andmetega. Selline võrdlus loob selguse, kas tegemist on erakorralise sündmusega või on see midagi tavapärast, näiteks suurt koormust tekitav protsess, mis käivitub korra 24 tunni jooksul.

Teiseks sobivaks meetodiks on andmete põhjal trendide ja prognooside kalkuleerimine. Nendele kalkulatsioonidele tuginedes on võimalik tuvastada probleemsed olukorrad, näiteks vaba ressursi peatse lõppemise.

Dünaamilise seire puhul peab arvesse võtma asjaolu, et arvutatud mõõtepunkti elementide loomisel peab rakendus iga andmepunkti eraldi andmebaasist küsima või salvestama väärtuste puhvris. Väärtuste puhvriks on eraldatud osa mälust [8]. Andmebaasist küsimine on aeglasem variant, samas on rohkete andmete talletamine vahemälus ressurssi raiskav. Parimaks praktikaks peetakse suure väärtuste puhvri seadistamist, kooskõlas seireraamistiku optimeerimisega. Seireraamistik peaks olema optimeeritud selliselt, et vaadataks läbi minimaalne kogus andmepunkte, mille põhjal on võimalik luua eesmärgistatult täpseid prognoose [9].

2.4 Dünaamiliseks seireks sobivad andmestikud

Dünaamilise seire üheks eelduseks on see, et andmetest peab olema võimalik trende tuvastada ning trendid ei tohiks olla liiga sagedasti suunda muutvad [10].

Kettamahu kasutuse andmestik on hea, kuna reeglina ei ole sellel väga ulatuslikke kõikumisi või on need ajutised. Heaks näiteks suuremahulisest, kuid ajutisest mahu kasutuse muutusest, on andmete kustutamine või pakkimine, mida tehakse korra 24 tunni jooksul. Kuna andmestik on aeglaselt muutuva iseloomuga, saab sellele tuginedes üsna täpselt prognoosida, kui kauaks vaba ressurssi jagub sarnase kasutuse jätkumise korral. Ühtlasi peab trendi kalkulatsioon olema loodud selliselt, et prognoos reageeriks piisavalt kiiresti senisest trendist erinevate tulemuste peale.

Kettamahu kasutuse seire jälgib vaba mahu suurust ning selle muutumist läbi aja. Seire jaotatakse haakepunktide kaupa erinevate kirjetena, kuna nende vaba maht ei ole omavahel seotud ning ka haakepunktide olulisus süsteemi töö säilitamisel, peale vaba ruumi lõppemist, on erinev [11]. Salvestuspinna mahud on olulised mõõtepunktid, kuna vaba mahu lõppemisel lakkab enamus süsteeme töötamast või kaotavad need suure osa enda funktsionaalsusest.

Muutmälu ehk *RAM* (*Random Access Memory*) kasutatakse andmete kirjutamiseks ning lugemiseks. See on kiire lahendus vajalike andmete ajutiseks talletamiseks ning jagamiseks teistele komponentidele kasutamiseks. Andmeid talletatakse vaid süsteemi töö ajal ning järgmise käivitumise korral on vanad andmed kustunud [12]. Mälumahtude seire on oluline, et tekitada ülevaade, kas süsteemis on piisavalt ressurssi töös olevate rakenduste jaoks vajalike andmete talletamiseks. Vaba mälumahu puudumisel jäävad enamasti rakendused töösse, aga oluliselt langenud sooritusvõimega. Tihti kasutatakse sellises olukorras saaleala, kui muutmälu pole piisavalt saada. Reeglina ei ole see eelistatud lahendus, kuna muutmälu on saalealast olulisemalt kiirem [13].

Sarnaselt kettamahu seirele jälgib mälumahtude seire vaba mahu suurust ning selle muutumist läbi aja. Kuna muutmälu operatsioonid on salvestuspinnaga võrreldes kiiremad ning andmete vahetus võib olla sagedasem, võib olla ka selle andmestiku põhjal keerukam usaldusväärse trendi tuvastamine. Antud töö skoopi sai see andmestik valitud, et testida dünaamilise seire võimekust teha korrektseid prognoose kiiresti muutuva andmestikuga keskkonnas.

2.5 Dünaamilist seiret võimaldavad funktsionaalsused

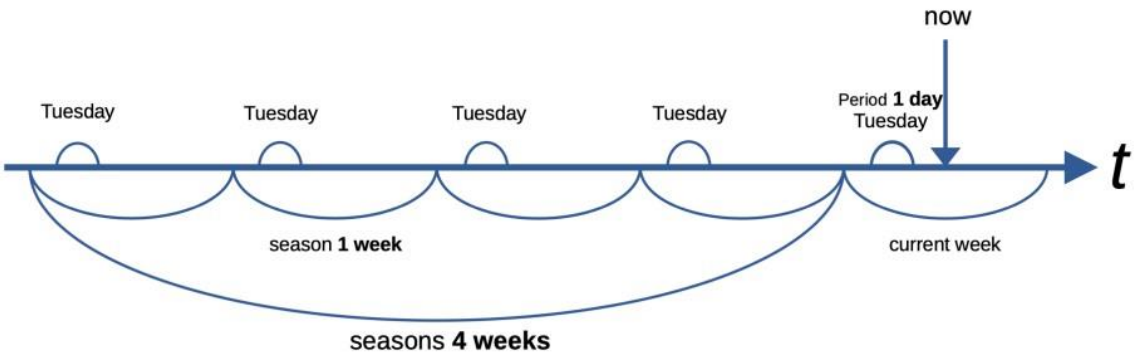
Dünaamilise seire teostamiseks on vajalikud funktsionaalsused, mis suudavad kalkuleerida trende ning teha nende põhjal prognoose tulevaste väärtuste kohta. Nendele kalkulatsioonidele toetudes on võimalik loobuda senisest praktikast alarmide tekitamisel staatilisi piirväärtusi kasutada. Trendide ja prognooside põhjal saab tuvastada vajalike ressursside olemasolu ja ka olukorrad, kus süsteemi tõrgeteta tööks vajalik ressurss kasutatakse täiel määral ära.

2.5.1 Trendifunktsioonid

Trendifunktsioonid Zabbix rakenduses kasutavad kalkulatsioonide tegemiseks trende, mitte ei päri andmebaasist kõiki andmepunkte vajaliku perioodi ulatuses. Trend on Zabbix rakenduse kontekstis ühe tunni agregeeritud andmestik, mis talletatakse vahemälus. Trendifunktsioonide kalkuleerimiseks kasutatakse STL dekompositsioonimeetodit [14], [15]. Kuna päringuid ei tehta andmebaasi ning andmeid hoitakse agregeeritud kujul, on seda tüüpi andmestik eelistatud variandiks mõõtepunktide puhul, mis vajavad andmeid pikast ajaperioodist [16].

Zabbix rakenduse poolt pakutavatest funktsioonidest [16] testiti kõiki ning siinkohal tuuakse välja mõned, mis leidsid kasutust antud töö testimise faasis või lõplikus raamistikus.

Testimisel kasutati funktsiooni baselinewma, mis kalkuleerib andmestikust määratud ajaperioodi alusväärtuse. Seda funktsiooni saab kasutada võrdluste tegemisel ning otsustamisel, kas praegused andmed on alusväärtusest erinevad. Antud funktsioon kasutab season (vaatlusperiood) jaotust, mille abil saab võrdlusesse võtta vaid olulise ajaperioodi. Season tähendab ühe tsükli pikkust, mida võib olla rohkem kui üks ning millele lisatakse võrdlustes kasutatav ajaaken. Joonisel 4 on näitena nelja nädalane vaatluse periood, milles iga season kestab ühe nädala ning võrdluses kasutatav ajaaken kestab ühe päeva. Tuvastamiseks kas teisipäevased andmed on alusväärtusega võrreldes erinevad, võrreldakse andmeid eelmise nelja nädala teisipäevaste andmetega. Selline jaotus aitab tuvastada tsüklilisi koormusi ning võib olla abiks valepositiivsete alarmide vähendamisel.



Joonis 4. Season kasutamine baselinewma funktsioonis

Keskmiste väärtuste kalkuleerimiseks kasutati funktsiooni trendavg, mis arvutab määratud ajaperioodi keskmise väärtuse. Antud funktsioonil puudub season võimalus, aga selle asendamiseks on võimalik arvutatud mõõtepunktidesse kirjutada valemi, mis võrdleks praeguseid andmeid 24 tunni või 7 päeva vanuste andmetega ning sellisel meetodil tuvastataks muudatused varasemaga võrreldes.

2.5.2 Prognoosifunktsioonid

Prognooside tegemiseks ei kasutata trende ega agregeeritud andmeid, vajalik andmestik talletatakse vahemälus. Seda peab arvesse võtma antud funktsioone kasutades, kuna kasutades pikki perioode prognooside täpsuse tõstmiseks, tekib suur vahemälu kasutus. Soovi korral peab andmetest arvutatud mõõtepunkte kasutades keskmisi väärtusi kalkuleerima, mis omakorda loob uue keerukuse andmete korrektsete väärtuste säilitamisel. Prognoose tegevaid funktsioone on Zabbix rakenduses kaks tükki.

Forecast funktsioon prognoosib varasemate väärtuste põhjal tulevasi väärtusi. Seda funktsiooni teostatakse nelja sammuna:

- Andmete pärimine vahemälust
- Andmete asetamine valitud funktsiooni valemisse
- Vähimruutude meetodi rakendamine [17]
- Väärtuse kalkuleerimine [9]

Timeleft funktsioon kalkuleerib varasemate andmete põhjal seadistatud piirmäärani jõudmise aja sekundites. Kalkulatsioon teostatakse forecast'i funktsioonile sarnase nelja sammuna, kasutades neljanda sammu juures timeleft'i jaoks mõeldud valemit [9].

2.6 Trigerite optimeerimine valealarmide vähendamiseks

Trigerite abil defineeritakse tingimused, mille peale aktiveeritakse alarm ning käivitatakse teavitamise protsess või käivitatakse trigeri aktiveerimise juurpõhjuse kõrvaldav skript. Trigerite aktiveerimise tingimusi on reeglina üks, aga neid võib kombineerida ühe trigeri alla ka mitmeid. Triger aktiveeritakse kui sellele määratud loogiline tehe osutub tõeseks. Trigeri tingimused võivad olla lihtsast võrdlusest keerukamad ning sisaldada funktsioone. Funktsioonid võivad kontrollida varasemaid väärtusi andmebaasist, teha väärtustega kalkulatsioone või prognoosida nende põhjal tulevasi väärtusi.

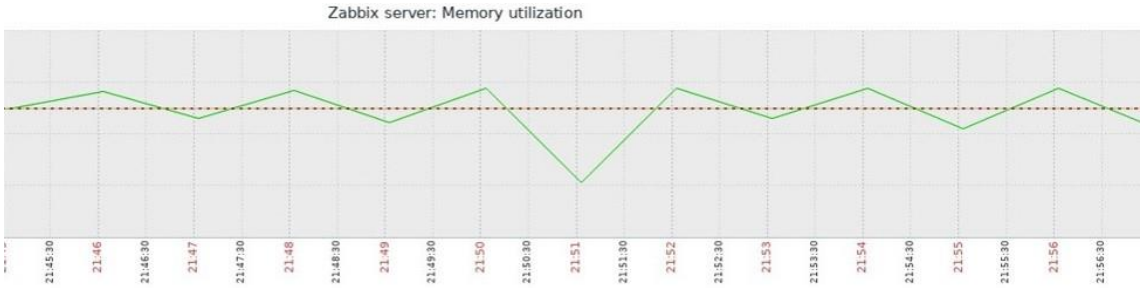
Trigeri seadistamisel peab kaaluma selle jaoks tehtavate päringute kogust ning kalkulatsioonide poolt tekitatavat koormust seiresüsteemile. Ajaloolisi andmeid kogudes saab süsteemi tööst hea ülevaate, samas kulub selleks palju päringuid ning sellega võib kaasneda andmete hoidmine vahemälus. Optimaalse vahemiku leidmine on oluline, et saada adekvaatsed andmed ning hoida koormust minimaalsel tasemel.

Ressursside kasutus seireraamistiku loomisel on oluline aspekt, millega tuleb arvestada. Samas on oluline koguda piisavalt palju andmeid, et kalkuleeritud prognoosid oleksid võimalikult täpsed. Mida suurem on andmepunktide kogus, seda edukamalt saab nende põhjal tulevasi väärtusi prognoosida.

Valealarmideks seires loetakse teavitusi, mis tekivad ajutise koormuse peale ning mis ei vaja inimese poolset sekkumist. Üks piirväärtusest kõrgem tulemus ei tohiks tekitada teavitust, kui selleks pole spetsiifilist vajadust. Hea praktika on trigerite aktiveerimiseks kontrollida mitmeid väärtusi, mitte ainult ühte ja viimast. Oodates trigeri aktiveerimiseks mitut ebastandardset väärtust, saab minimeerida valealarmide tekkimist seires [18].

Oluline osa trigeri seadistamisest on selle mitteaktiivseks tegev tingimus. Triger peab püsima aktiivsena, kuni saab suure tõenäosusega väita, et trigeri aktiveerimise tinginud olukord on normaliseerunud. Liiga leebed tingimused trigeri mitteaktiivseks tegemise seadistuses, võivad tingida olukorra, kus triger aktiveeritakse korduvalt ühe probleemse

episoodi jooksul. Joonisel 5 on näide sellest, kuidas ebaoptimaalselt seadistatud trigeri aktiveerimine ning desaktiveerimine tekitab seiresüsteemis valealarme. Jälgimisel on mälukasutus, millele on seatud fikseeritud piirväärtus, mis on joonisel kuvatud tumepunase punktiirina. Roheline joon tähistab ressursi kasutuse määra. Kui triger on seadistatud aktiveerima ühe piirväärtusest kõrgema väärtuse peale ning desaktiveerima samuti ühe piirväärtusest madalama väärtuse peale, tekitatakse joonisel kuvatud 12 minuti pikkuse perioodi jooksul sama sisuga alarm kuuel korral.



Joonis 5. Näide valealarme tekitavast andmestikust

Prognoosivad funktsioonid, eriti lühikese tagasivaatamise perioodiga, reageerivad toimunud muudatustele andmestikus hüppeliselt [9] ning suure tõenäosusega prognoosivad tuleviku väärtusi reaalsusest oluliselt suurema või väiksemana, sõltuvalt sellest, millises suunas trendiga tegemist on. Seda asjaolu peab arvesse võtma trigerite loomisel ning arvestama sisse puhvri, mille jooksul saab prognoos end korrigeerida, ilma trigerit aktiveerimata.

Trigerite seadistamise keerukust tõstab asjaolu, et valealarmid ei ole ainus aspekt, millega arvestada seireraamistiku loomisel. Valealarmide vähendamise kõrval on tähtsaks eesmärgiks trigerite aktiveerimine võimalikult kiiresti, mis ennetaks intsidentide teket või minimeeriks nende kestvust. Loodud seadistus peab mõlemad tingimused efektiivselt täitma.

2.7 Dünaamilist seiret toetavad protsessid ja seadistused Zabbix rakenduses

Zabbix rakendus koosneb paljudest protsessidest, mille käitumist saab serveri konfiguratsioonifaili muutmisega mõjutada [19]. Suuremate keskkondade või

ebastandardse seire ülesehitamise korral on vajalik ka Zabbix serveri seadistuse muutmine. Dünaamilise seireraamisliku loomisel peab arvestama suurenenud vahemälu kasutusega ning korduvate päringutega andmebaasi poole.

Ajaloolisi väärtusi haldav protsess teostab päringuid, mis vajavad andmebaasi ühendust [20]. Antud protsess leiab kasutust arvutatud mõõtepunktide puhul. Dünaamilist seiret teostatakse arvutatud mõõtepunktidega, seega vaikeväärtustega konfiguratsioonifail ei taga suure tõenäosusega piisavalt ressursse, et täita vajalikke toiminguid ilma järjekorda tekitamata. HistoryCacheSize ja StartHistoryPollers väärtused võivad vajada vastavalt seire keskkonna NVPS (*new values per second*) väärtusele suurendamist.

Trendide puhvrisse talletatakse kalkuleeritud andmed, et hilisem andmete pärimine toimuks kiiremini [20]. Trendide kasutamine kalkulatsioonide tegemiseks toob endaga kaasa vahemälu kasutuse tõusu, mille tõttu võib CacheSize muutuja vajada suurendamist.

Väärtuste puhvris hoitakse ajaloolisi andmeid, et nende pärimine oleks kiirem kui alternatiiv andmebaasi päringuid kasutada [8]. Sõltuvalt seireraamistiku suurusest võib ValueCacheSize vaikeväärtus olla liiga väike ning vajada suurendamist.

3 Seireraamistiku loomine

Seireraamistiku all mõeldakse Zabbix rakenduse mõõtepunkte ja trigereid, mille koostöös teostatakse kindla skoobiga seiret. Iga andmeid koguv mõõtepunkt peaks omama vähemalt ühte trigerit. See tähendab, et kogutav andmestik on süsteemi heaolu aspektist oluline ning trigerit aktiveeriva tingimuse täitmise peale tõstatatakse alarm. Erandina võib raamistik sisaldada mõõtepunkte, mille põhjal alarme ei tekitata, aga nendele olemasolu aitab mõne teise alarmi tõrkeanalüüsi teostada.

Seiretegevus tekitab palju andmeid. Raamistiku loomisel peab arvesse võtma andmebaasi kirjutatavate sissekannete kogust ning ettevõtte eeskirjadega määratud perioodi andmete säilitamiseks. Zabbix rakendus jagab andmed talletamisel kahte kategooriasse: ajalugu (sissekanne, mis tehakse andmebaasi muutmata kujul) ja trend (sissekanne, mille puhul ei kasutata igat andmepunkti, vaid kalkuleeritakse tunni keskmine väärtus).

Ajaloolisi andmeid on soovituslik alles hoida minimaalselt kaua [21]. Reeglina on need andmed kasutusel selleks, et tuvastada mõnda ebaregulaarsust süsteemi näitajates reaalajas või lähiminevikus. Parima ülevaate andmestikust annavad just võimalikult väikese intervalliga kogutud andmed. Iga mõõtepunkti puhul saab määrata, kui pikalt selle väärtusi andmebaasis hoitakse ning kui pikalt trende kalkuleeritakse. Talletamise piirmäära ületanud andmebaasi sissekannetest ja kasutajaliidese kaudu kustutatud mõõtepunktide andmetest vabaneb Zabbix seadistustes määratud regulaarsusega, et hoida andmebaasi mahtu kontrolli all. Vaikeväärtus andmete kustutamiseks on kord tunnis.

Trendi andmestikku kalkuleeritakse iga mõõtepunkti tunni keskmine, suurim ja väikseim väärtus. Lisaks talletatakse andmete koguarv iga tunni kohta [21]. Kuna trendi andmeid kogutakse suure intervalliga, võib seda tüüpi andmeid alles hoida pikema perioodi vältel. Trendidele tuginedes saab teha järeldusi süsteemide koormuse muutustest pikema perioodi vältel ning teostada muid vaatlusi, mis ei nõua väikese intervalliga korjatud andmeid.

Zabbix arvestab serverite ja puhverserverite koormust näitajaga NVPS. Kindlat piirväärtust pole Zabbix dokumentatsioonis välja toodud, millest alates võib tekkida rakenduse töös häireid, kuna seda muutujat mõjutavad liiga palju erinevad tegurid nagu andmebaasi ja operatsioonisüsteemi valik, süsteemi ressursid, andmestiku tüüp jne. Küll

aga on parim praktika koguda vaid olulisi andmeid ning mitte lasta NVPS väärtusel tõusta ilma mõjuva põhjuseta.

Mainitud eripärasid peab arvestama seireraamistiku loomisel. Kokkuvõtvalt ei tohiks otse andmebaasi talletada liiga palju andmeid, analüüsi jaoks vajalikud andmed tuleb talletada trendidena ning hoida NVPS väärtust võimalikult madalana.

3.1 Metoodika analüüs

Diplomitöö esimeses faasis luuakse kaks malli, kuhu lisatakse arvutatud mõõtepunktid, et testida nende võimekust vajalikke prognoose teostada. Arvutatud mõõtepunkti väärtuse leidmise käigus on reeglina vaja mitmeid erinevaid ajaloolisi andmeid pärida. Arvukate päringute tõttu ei ole optimaalne esimesi katsetusi teha toodangukeskkonnas, kuna tekkiv koormus on prognoosimatu ning võib hüppeliselt tõsta keskkonna koormust ja seeläbi toodangukeskkonna töös häireid tekitada.

Testimiseks luuakse arvutatud mõõtepunktide elemendid ning neile määratakse Zabbix dokumentatsioonist leitud soovituslikud väärtused [9]. Testimise käigus analüüsitakse, millised neist on kõige täpsemalt prognoosinud mõõtepunktide tulevasi väärtusi. Ebatäpselt prognoosivad arvutatud mõõtepunkti elemendid kustutatakse ära, rahuldava tulemusega elemendid jäetakse alles. Alles jäänud elementide seadistuste põhjal tehakse järeldused, millised parameetrid paremini toimivad ning sellest lähtuvalt luuakse uusi, vähesel määral muudetud parameetritega arvutatud mõõtepunkte, mida võrreldakse omakorda varasemate elementide tulemustega. Seda protsessi korratakse, kuni jõutakse arvutatud mõõtepunktideni, mis on suutnud erinevate koormuste ja anomaaliate korral edukalt prognoose kalkuleerida.

Testimist tehakse korraga kahel meetodil. Üheks meetodiks on reaalajas andmete kogumine ning nendele tuginedes prognooside kalkuleerimine. Antud lähenemise suurimaks miinuseks on suur ajakulu. Plussiks on see, et andmed tulevad töötavast süsteemist ning on selle tõttu reaalsete keskkondade andmeid paremini esindavad.

Teine meetod on Python programmeerimiskeeles loodud skriptid, mis genereerivad andmeid ning saadavad need Zabbix serverisse. Sellise meetodiga saab oluliselt kiiremini mustreid testida, kuna andmeid võib serverisse saata tavapärasest suurema sagedusega. Keerukuse loob reaalsust edukalt peegeldavate väärtuste skriptidesse kirjutamine.

Skripte kasutades saab simuleerida erinevas tempos vähenevat vaba kettamahtu. Samuti saab luua ka täiesti juhuslikke väärtusi ning seeläbi testida, kas prognoosivates funktsioonides tekitab see mingeid anomaaliaid. Skriptidega loodavate väärtuste meetod on kiire, aga see ei pruugi olla tõetruu meetod jäljendama reaalse keskkonna tööd. Seetõttu võetakse skriptid kasutusele anomaaliate tuvastamiseks ning kindlate mustrite ja prognoosivate funktsioonide esmaseks testimiseks.

3.2 Erinevate kasutusmustritega serverite analüüs

Seireraamistiku loomisel on oluline roll teooriate testimise hõlbustamisel. Märkimisväärselt lihtsam on analüüsida prognooside efektiivsust grupeerides seireobjektid ressursside kasutuse järgi ning võimalikult vähesel määral kontrollides seires olevaid objekte individuaalselt.

Seireobjektide ressursside kasutuse graafikutelt on võimalik tuvastada erinevaid mustreid. Mustritele keskendumine on oluline, kuna see aitab seireobjekte grupeerida erinevatesse kategooriatesse. Grupeerimise eesmärgiks on tekitada arusaam erinevatest kasutusjuhtudest ning seeläbi tagada, et uuringu skoobis oleks võimalikult palju erinevate kasutusmustritega objekte. Grupeerimine lihtsustab analüüsi, luues selgema arusaama milliste kasutusjuhtude korral prognoosid ootuspäraselt käituvad ning lihtsustab ebatäpseid prognoose tekitavate kasutusmustrite tuvastamist.

Kettamahu ning mälukasutuse puhul grupeeriti seireobjektid kolme gruppi: vähesel määral muutuv vaba ressurss, lühikese aja jooksul või suurel määral muutuv vaba ressurss ja objektid, mille vaba ressurss kasutatakse regulaarselt kas maksimaalsel või selle lähedasel määral ära.

Kõik kolm gruppi seireobjekte tekitavad prognoosivates funktsioonides erinevaid tulemusi ning sellest lähtuvalt vajavad erinevat lähenemist. Korrektsete trigerite seadistamiseks, võeti kõiki erinevaid gruppe arvesse ning lisati trigeritesse või kalkuleeritud mõõtepunktidesse antud grupi spetsiifikat arvesse võtvad tingimused.

3.3 Esmased katsetused

Esmased katsetused teostati kolme seireobjekti kasutades. Zabbix serverisse loodi kaks malli, üks neist mälukasutuse ning teine kettamahu kasutuse seireks. Mallidesse lisati

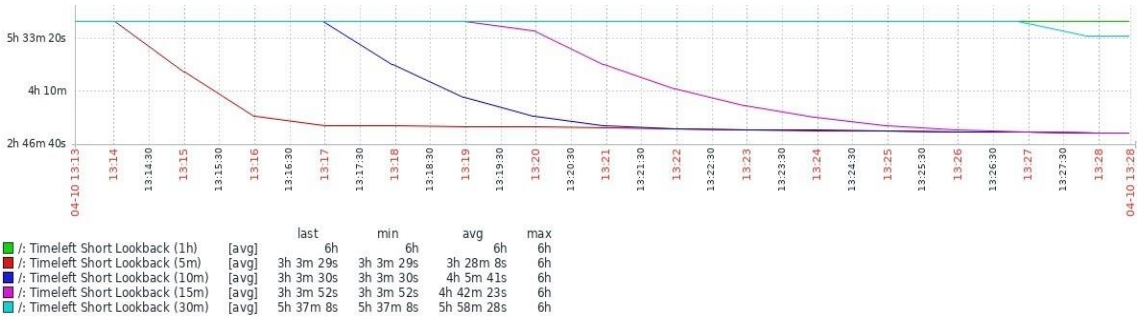
arvutatud mõõtepunktid ning vaadeldi nende võimekust erinevate kasutusmustrite korral korrektseid prognoose kalkuleerida.

Seireobjektid jaotati kolmeks ning nende eesmärgiks oli imiteerida erinevaid käitumismustreid. Üks objektidest oli tööjaam, mille ressursside kasutust ei mõjutatud mingil moel, et saada võimalikult tõetruud kasutuse andmed.

Teine objektidest oli virtuaalne Linux server, mis loodi Virtualbox virtualiseerimise tarkvara kasutades. Serveri peal teostati manuaalselt tegevusi, mis vajasid mäluressurssi ning kirjutati ja kustutati andmeid, millega testiti kettamahu kasutuse seiret. Tegevuse eesmärk oli tekitada koormust reaalajas ning vaadelda raamistiku jaoks loodud trigerite efektiivsust võrreldes staatilise seire trigeritega ja hinnata mõõtepunktide prognooside täpsust võrreldes reaalse ressursside kasutusega.

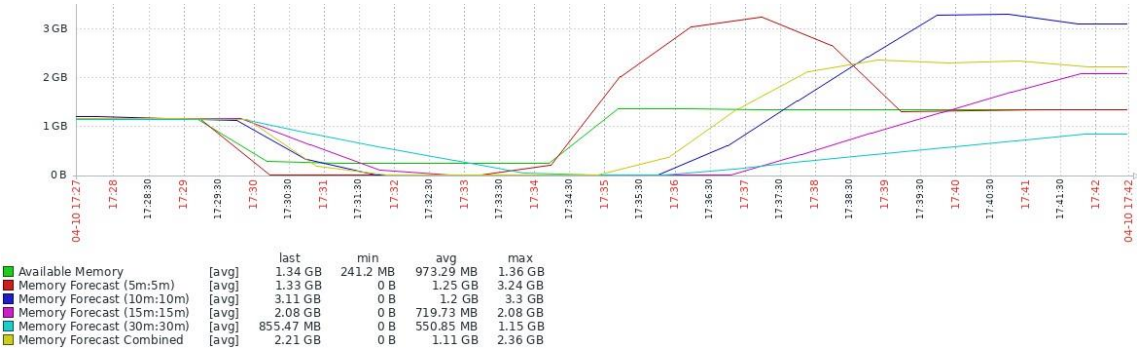
Kolmas objektidest oli virtuaalne Linux server. Serverile laeti Zabbix Agent agentprogramm ja Zabbix Sender programm, millega saab Zabbix serverile või puhverserverile andmeid edastada. Serveri andmestik genereeriti Python skripte kasutades. Andmestiku imiteerimise üheks eesmärgiks oli aja säästmine. Zabbix Sender seadistati uusi andmeid edastama iga sekund. Teiseks imiteeritud andmete eeliseks on vabadus luua täpselt selline ressursside kasutus, nagu erinevate oletuste testimiseks vajalik oli.

Katsetuste käigus vaadeldi andmeid ning nende põhjal kalkuleeritud prognoose. Joonis 6 kujutab timeleft prognoosiva funktsiooni [7] erinevate seadistuste puhul saadud tulemusi. Graafikul kujutatakse olukorda, kus juurkataloogi kirjutati suures mahus andmeid. Graafikult on tuvastatav, et kõik mõõtepunktide väärtusi kujutavad jooned algavad sarnase prognoosiga graafiku ülemises osas ning mingis punktis muudetakse prognoosi ja joon liigub graafikus alla, tähistamaks uut prognoosi. Mida rohkemate andmete põhjal prognoose tehakse, seda pikem saab olema kalkulatsioonide reaktsiooniaeg toimunud muudatuste peale, sellest ka erinevate värvustega joonte erinev reageerimise kiirus. Antud erinevus omab suurt tähtsust õigeaegselt aktiveeruvate trigerite loomisel.



Joonis 6. Näide erinevatest timeleft reageerimisaegadest

Joonis 7 kujutab olukorda, kus vaba mälu maht vähenes neljaks minutiks ning seejärel taastus tavapärase nivoo peal. Rohelisena on joonisel kujutatud vaba mälu maht ning kõik teised värvused on erinevate seadistustega prognoosid. Jooniselt on tuvastatav, kuidas erinevate seadistuste korral võib tulemuseks saada teineteisest suurel määral eristuvaid prognoose. Joonisel nähtust on võimalik järeldada, et optimaalse seadistuse leidmine on väga oluline aspekt seireraamistiku loomisel.



Joonis 7. Näide erinevatest forecast prognoosidest

3.3.1 Testkeskkonna katsetuste järeldused

Katsetused jaotati kahte erinevasse kategooriasse: mälukasutus ja kettamahu kasutus. Katsetuste varajases faasis sai selgeks, et sarnaselt seadistatud funktsioonid ei suuda mõlema andmestiku puhul efektiivselt prognoose teha. Sellest järeldati, et andmestikud ei sarnane piisaval määral ning vajavad teineteisest sõltumatut lähenemist.

Esimese eelistusena prognooside teostamiseks oli timeleft funktsionaalsus. Timeleft on intuitiivsem kui selle alternatiivid, kuna tagastab iga ressursi puhul ajalise väärtuse selle lõppemiseni. Selliste väärtuse korral puudub vajadus seiratava süsteemi spetsiifikaga tutvuda, enne kui saab piisava ülevaate aktiveeritud trigeri kriitilisuse astmest.

Mälukasutuse seire puhul saavutati parimad tulemused kasutades lühikese analüüsiperioodiga forecast funktsioone. Pikemat perioodi analüüsiv forecast oli täpsem, aga selle miinuseks oli aeglane reaktsioon koormuse tõusule ning aeglasem alarmi trigeri aktiveerimine, võrreldes staatilise seire trigeriga.

Alarmi tõstatamise kiirusele lisaks, on oluline tuvastada, kas andmestik viitab selgele vajadusele triger aktiveerida. Trigerisse kirjutati ajaloolisi andmeid kontrolliv funktsionaalsus, mis võrdleb viimaseid andmeid 24 tunni ning 7 päeva vanuste andmetega. Selline võrdlus aitab tuvastada olukorra, kus mingil perioodil on regulaarselt kõrgem koormus ning selle peale ei tuleks trigerit aktiveerida.

Mälukasutuse seireks ei suudetud edukalt kasutusele võtta timeleft funktsionaalsust, kuna see on liialt tundlik ka väikestele muutustele andmestikus. Funktsiooni tundlikkusest tingituna tekkisid ebaühtlased prognoosid ning trigerite aktiveerimine ja desaktiveerimine muutus kontrollimatuks. Ressurss võis olla otsakorral, kuid minimaalse vaba ressursi tõusu peale, reageeris timeleft viisil, nagu probleemne olukord oleks lahenenud, kuna vaba ressurssi tekkis ebaoluliselt vähesel määral juurde. Sellist käitumist üritati muuta parameetreid optimeerides, kuid ei suudetud viia edukaks seireks vajaliku täpsuseni.

Kettamahu kasutuse seire puhul olid andmestikus väärtuste muutused ajas väiksemad ning sellest tingituna suutsid timeleft funktsioonid suurema täpsusega korrektseid prognoose teha. Seire jaotati omakorda kaheks: kiireloomuliselt muutuvad väärtused ning aeglaselt täituv salvestuspind. Need kaks tingimust said loodud eesmärgiga tuvastada lühikese aja jooksul tekkinud probleeme ning lisaks prognoosida lõppevat salvestuspinda mitu päeva ette.

Kiires tempos vähenevat vaba kettamahtu tuvastati 10 minuti andmete põhjal, et vajadusel oleks võimalik triger varakult aktiveerida. Korduvaid mustreid tuvastatakse võrreldes praegusi andmeid 24 tunni taguste andmetega, et tuvastada ajutiselt kõrgemat koormust, mille peale ei tohi trigerit aktiveerida.

Lauges tempos täituv kettamaht tuvastati kolme päeva andmete põhjal. Kuna kalkulatsioonides kasutatav periood on pikem kui kiireloomuliste muudatuste jaoks kasutatava arvutatud mõõtepunkti korral, siis see silus ära ajutised väärtuste muutused ja registreeris pikaajalisemad trendid. Korduvate mustrite tuvastamiseks kasutatati võrdlust 24 tunni taguste andmetega.

3.4 Zabbix toodangukeskkonnas raamistiku seadistamine

Toodangukeskkond, milles seireraamistik kasutusele võetakse, koosneb ühest serverist ning viiest puhverserverist. Puhverserverite eesmärgiks on jaotada koormust erinevate keskkondade vahel ning tagada võimalikult ohutu ligipääs erinevatest võrkudest. Raamistikku lisatakse kokku 897 seireobjekti, keskkonna keskmine NVPS väärtus on 729. Agentrakendused on seadistatud andmeid edastama puhverserveritesse, mis omakorda edastavad andmed serverisse.

Kasutusele võetud raamistiku trigerid on kujutatud Lisa 2 ja Lisa 3 joonistel. Kokku pandi andmeid koguma neli arvutatud mõõtepunkti. Kaks mõõtepunkti seadistati mälukasutuse seireks ning kaks kettamahtude seireks, mille puhul iga haakepunkti jaoks luuakse eraldi mõõtepunkt. Mälukasutuse pikka perioodi analüüsiv mõõtepunkt arvutas prognoose viimase 24 tunni andmete põhjal ning lühikest perioodi analüüsiv mõõtepunkt kasutas viimase 7 minuti andmeid. Kettamahtude seire jaoks kasutati 72 tunni ja 10 minuti andmeid. Ajalised parameetrid mõõtepunktide jaoks valiti testkeskkonnas teostatud katsetuste tulemusel, arvestades prognooside täpsust ning valealarmide kogust.

Trigereid loodi neli tükki. Kaks trigerit loodi mälukasutuse seire ja kaks kettamahtude seire jaoks. Mõlema andmetüübi jaoks loodi üks triger kiiresti reageerivaks ning üks pikemat perioodi jälgiv ning hoiatavat laadi, mis teavitab ressursi lõppemisest aegsasti ette.

Selleks, et ühe mõõtepunkti kohta ei aktiveeritaks mitu erinevat alarmi, loodi trigeritele omavaheline sõltuvus. Korraga saab aktiivsena olla vaid üks ja kõige kriitilisem triger. Sellise seadistuse eesmärk on vähendada alarmide kogust seires ning tekitada ühe juurpõhjuse kohta vaid üks, kõige kriitilisem ning ühtlasi võimalikult informatiivne alarm.

Madalaimal kohal hierarhias on pikka perioodi kontrolliv triger. Antud triger reageerib kiireloomulistele muudatustele aeglaselt ning selle tõttu tuleb mitme trigeri aktiveerimise korral eelistada teavitusteks kiiremini reageerivat trigerit, kuna selles sisalduvad andmed on kõige uuemad ja annavad parima ülevaate olukorra kriitilisusest.

Trigerites kasutatakse mõõtepunktide erinevate ajaperioodide minimaalseid ja maksimaalseid väärtusi. Selliselt trigereid üles ehitades on väiksem tõenäosus, et üksikud tavapärasest erinevad väärtused ei aktiveeri trigerit.

3.5 Tulemuste analüüs

Seireraamistiku analüüsiks filtreeriti aktiveeritud alarmide seast välja kõik lõputöö skoobis olnud mõõtepunktid. Eraldi võrreldakse mälukasutuse ning kettamahtude seiret, et tuvastada erinevate andmestike puhul dünaamilise seire efektiivsus. Võrdlemiseks kasutatud kriteeriumid osutusid valituks sellest lähtuvalt, et seire peab alarmi tõstatama võimalikult varakult, kuid mitte liiga kergekäeliselt, et hoiduda valealarmide tekitamisest.

Filtreeritud andmetest loendatakse kokku aktiveeritud alarmid dünaamilise ja staatilise seire poolt. Loendatakse kokku unikaalsed alarmid, mille üks raamistik suutis tuvastada ning teine mitte. Valenegatiivseid alarme käesolevas töös eraldi ei käsitleta.

Kokku loendatakse valealarmideks kvalifitseeruvad alarmid. Valealarmideks loetakse 15 minuti jooksul taastunud alarm, mis ei vajanud inimese poolset sekkumist ning 24 tunni jooksul korduvad alarmid ühe mõõtepunkti kohta.

Viimase muutujana arvestatakse seda, kumb seireraamistik suutis kiiremini tõstatada alarmi. Alarmi tõstatamise kiirust võrreldakse vaid olukorras, kus mõlemad raamistikud suutsid olukorra tuvastada ning tegemist ei olnud valealarmi kriteeriumit täitva alarmiga.

Kuna andmeid kogutakse korra minutis ning selle järgselt algatatakse trigerite kontroll, siis tekkis olukordi, kus mõlemad raamistikud registreerisid alarmi samal ajal. Sellest lähtuvalt võib esineda olukordi, kus alarmide koguarv ei võrdu ülejäänud väljade summaga, kuna polnud võimalik eristada kiiremini reageerivat trigerit.

Tabelites 1, 2 ja 3 on kujutatud lõplikud tulemused peale 14 päeva pikkust testimise perioodi. Tabelis 1 on kujutatud mälukasutuse seire tulemused. Tabelisse lisati nii pikka kui ka lühikest perioodi jälgivate mõõtepunktide poolt tõstatatud alarmid.

Tabel 1. Mälukasutuse seire tulemused

Raamistik	Alarmide kogus	Unikaalsed alarmid	Valealarmid	Esimesena tõstatatud alarmid
Dünaamiline	12	3	6	2
Staatiline	11	3	6	1

Ligi pooled alarmidest taastusid peatselt peale tõstatamist ise ning ei vajanud inimese poolset sekkumist. Trigerid suutsid kinni püüda vaid kiireloomulised muudatused. Testperioodi jooksul ei suudetud tuvastada ühtegi aeglases tempos tõusvat trendi, mis viitaks liigsele koormuse lisandumisele või mälulekkele. Raamistikku kirjutati pikema perioodi trendide jälgimine just sarnaste olukordade tuvastamiseks.

Tabel 2 kujutab kettamahtude seire tulemusi.

Tabel 2. Kettamahtude seire tulemused

Raamistik	Alarmide kogus	Unikaalsed alarmid	Valealarmid	Esimesena tõstatatud alarmid
Dünaamiline	49	4	19	0
Staatiline	54	12	41	1

Lühikese perioodi jooksul toimuvatele muudatustele suutsid mõlemad raamistikud reageerida sarnase kiirusega. Enamasti tekitasid alarme väikeste mahtudega haakepunktide vaba mahu lõppemised. Vaba maht kirjutati täis liiga kiiresti, et mõõtepunktidel oleks olnud aega täitumise trendi tuvastamiseks. Selle tõttu ei suutnud dünaamiline raamistik kiiremini trigereid aktiveerida.

Dünaamilise raamistiku puhul võrreldi praegusi andmeid 24 tunni taguste andmetega ning tänu sellele suudeti tuvastada korduvad mustrid. Antud informatsiooni kasutati trigerite

aktiveerimisel ning selliselt toimetades vähendati valealarmide kogust. Valdav enamus staatilise seire alarmidest olid valealarmid ning ei vajanud sekkumist.

Tabel 3 kujutab pikka perioodi jälgivat prognoosi. Kuna staatilise seire puhul puuduvad võrreldavad andmepunktid, siis antud tabelisse koguti info vaid dünaamilise seire kohta.

Tabel 3. Dünaamilise seire prognooside tulemused

Raamistik	Alarmide kogus	Valealarmid
Dünaamiline	24	0

Pika ajaperioodi põhjal tehtavad prognoosid suutsid ilma valealarme tekitamata teavitada olukordadest, kus vaba maht haakepunktil oli lähitulevikus lõppemas. Kuna analüüsiti pikka perioodi, ei suutnud ajutised muudatused andmestikus prognoosi segada ning valepositiivseid alarme ei tekkinud.

3.6 Võimalikud arengusuunad

Lõputöö analüüsi tulemustest saab järeldada, et suurt tähelepanu tuleks pöörata valealarmidele seires. Lõputöö raames aktiveeritud trigeritest ei vajanud reageerimist staatilise raamistiku korral 72% ja dünaamilise raamistiku korral 41% juhtudest, kuid siiski registreeriti seirekeskkonnas ning nende põhjal tekitati teavitus seireobjekti haldajale. Selliseid olukordi tuleks täpsemalt analüüsida ning testida, kas oleks võimalik valealarme viia madalamale tasemele nii, et sellega ei väheneks reaktsiooni kiirus reaalsete intsidentide korral.

Valealarmide vähendamisele saaks kaasa aidata metoodikad, mida kasutati dünaamilise seire raamistiku loomisel. Üheks võimaluseks on tuvastada tsüklilised koormused, võttes võrdlusesse varasemad väärtused mõõtepunkti andmestikus. Trigeri tingimustesse ajalooliste andmete kontrolli lisamine, aitab välja filtreerida kindla regulaarsusega toimuvad sündmused, mille sarnaseid tuvastati antud lõputöö raames korduvalt.

4 Kokkuvõte

Lõputöö eesmärgiks oli dünaamilise seire funktsionaalsuste testimine Zabbix rakenduses. Lõputöö raames testiti dünaamilist seiret toetavaid funktsionaalsusi ning nendest loodi seireraamistik. Loodud raamistik võeti kasutusele toodangukeskkonnas 897 objekti seireks. Dünaamiline raamistik pandi võrdlusesse seireobjektidel juba kasutuses olnud staatilise raamistikuga ning nendega teostati samaaegselt seiret 14 päeva jooksul. Testperioodi lõpus analüüsiti mõlema raamistiku poolt aktiveeritud trigereid.

Trigerite aktiveerimise kiirus oli mõlema raamistiku puhul sarnane ning töö käigus ei tuvastatud meetodeid, mille abil saaks teavituse protsessi kiirendada, ilma valealarme juurde tekitamata.

Dünaamiline seireraamistik suutis vähendada valealarmide tõstatamist. Staatilise raamistiku puhul oli valealarme 72% ning dünaamilise raamistiku puhul 41%. Selle erinevuse eelduseks on varasemate andmete kaasamine trigeri aktiveerimise tingimustesse. Võttes arvesse, et trigerite aktiveerimise vahe oli minimaalne ning dünaamiline raamistik kulutab töö käigus rohkem ressursse, pole hetkel head põhjust staatilise seire välja vahetamiseks. Ajalooliste andmete võrdluse võiks kirjutada staatilise seire trigeritesse.

Testimise käigus seati eesmärk prognoosida ennetavalt kettamahu lõppemist, et vähendada alarme töövälisel ajal ning triviaalseid tegevusi valvepersonali poolt. Pikaajaliste andmete prognoos oli edukas ning seda funktsionaalsust saab kasutada ennetavate ning madala prioriteediga teavituste loomiseks, mis annavad märku kettamahu lõppemisest. Pikka perioodi kontrollivad mõõtepunktid sobiksid olemasoleva staatilise seire täiendamiseks. Antud funktsionaalsusel on otsene kasu ettevõtetele, kuna varajase teavituse korral saab probleemiga tegeleda ennetavalt tööajal ning see aitab vältida väljakutseid ja ületundide tasusid.

Kasutatud kirjandus

[1] Paul Parker, “Network Monitoring Overload and How to Survive,” [www] https://orangematter.solarwinds.com/2018/02/27/network-monitoring-overload-and-how-to-survive/ Kasutatud: 15.03.2023

[2] Jess Frame, Anthony Lenton, Steven Thurgood, Anton Tolchanov, Nejc Trdin ja Carmela Quinito, “Site Reliability Engineering book,” [www] https://sre.google/workbook/monitoring/ Kasutatud 15.03.2023

[3] “System and method for statistical performance monitoring,” by Yiping Ding, Kenneth W. Newman [www] https://patents.google.com/patent/US8000932 Kasutatud: 25.03.2023

[4] GNU. General Public License, version 2. [www] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html Kasutatud: 15.03.2023

[5] Zabbix. Roadmap. [www] https://www.zabbix.com/roadmap Kasutatud: 16.03.2023

[6] Dmitry Lambert, “Zabbix Agent: Active vs. Passive,” [www] https://blog.zabbix.com/zabbix-agent-active-vs-passive/9207/ Kasutatud: 16.03.2023

[7] “Zabbix Documentation, Prediction functions,” [www] https://www.zabbix.com/documentation/6.0/en/manual/appendix/functions/prediction Kasutatud: 23.03.2023

[8] “Zabbix Documentation, Value cache,” [www] https://www.zabbix.com/documentation/6.0/en/manual/config/items/value_cache Kasutatud: 23.03.2023

[9] Zabbix, „Forecasting trigger functions,“ [www] https://www.zabbix.com/documentation/5.0/assets/en/manual/config/triggers/prediction_docs.pdf Kasutatud: 23.03.2023

[10] Kartik Patel, “What Are Data Trends and Patterns, and How Do They Impact Business Decisions,” [www] https://www.dataversity.net/data-trends-patterns-impact-business-decisions/ Kasutatud 22.03.2023

[11] “The Linux Information Project, Mount Point Definition,” [www] http://www.linfo.org/mount_point.html Kasutatud: 11.04.2023

[12] Jon Martindale, “What is RAM? Here’s everything you need to know,” [www] https://www.digitaltrends.com/computing/what-is-ram/ Kasutatud: 26.03.2023

[13] “All about Linux swap space” [www] https://www.linux.com/news/all-about-linux-swap-space/ Kasutatud: 11.04.2023

[14] “Forecasting: Principles and Practice, STL decomposition,” [www] https://otexts.com/fpp2/stl.html Kasutatud: 10.04.2023

[15] Raul Kangro „Aegridade analüüs“ [www] https://courses.ms.ut.ee/MTMS.01.023/2016_fall/uploads/Main/aegread.pdf Kasutatud: 10.04.2023

[16] “Zabbix Documentation, Trend functions” [www] https://www.zabbix.com/documentation/6.0/en/manual/appendix/functions/trends Kasutatud: 10.04.2023

[17] “Least Squares Fitting,” [www] https://mathworld.wolfram.com/LeastSquaresFitting.html Kasutatud: 10.04.2023

[18] “Zabbix Blog, No more flapping. Define triggers the smart way,” [www] https://blog.zabbix.com/no-more-flapping-define-triggers-the-smart-way/1488/ Kasutatud: 12.04.2023

[19] “Zabbix Documentation, Server,” [www] https://www.zabbix.com/documentation/6.0/en/manual/concepts/server Kasutatud 11.04.2023

[20] “Zabbix Documentation, Process Configuration, Zabbix Server,” [www] https://www.zabbix.com/documentation/6.0/en/manual/appendix/config/zabbix_server Kasutatud: 11.04.2023

[21] „Zabbix Documentation, History and Trends,“ [www] https://www.zabbix.com/documentation/6.0/en/manual/config/items/history_and_trends kasutatud 22.04.2023

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

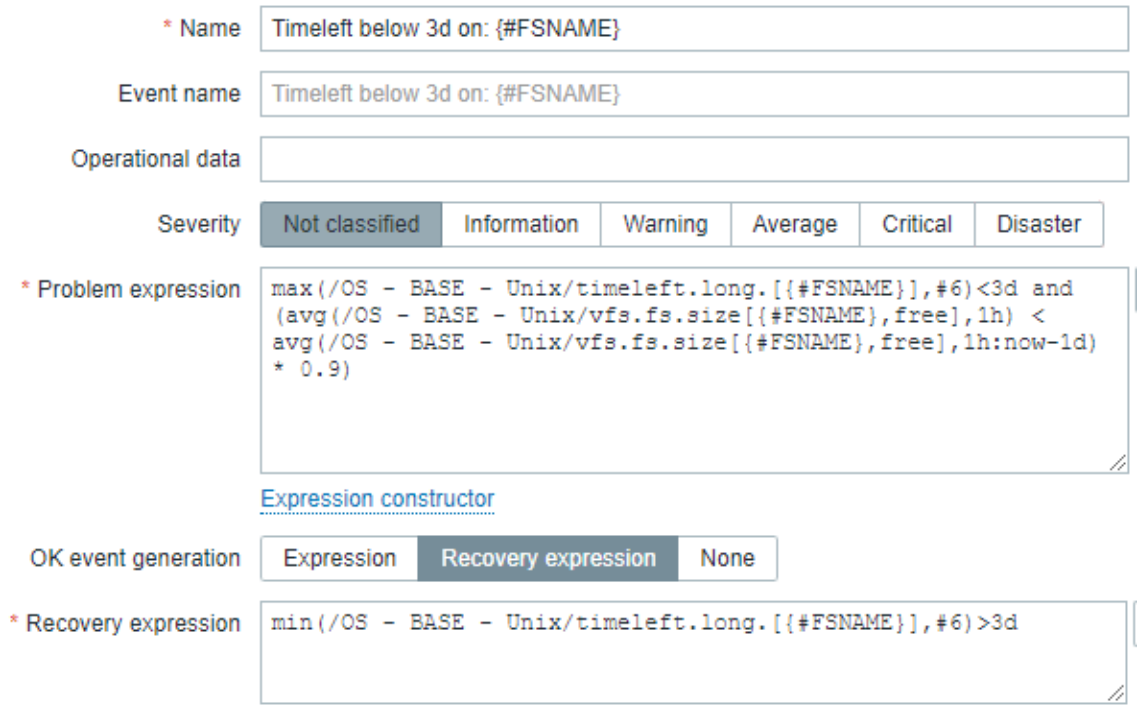
Mina, Marek Ott

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Dünaamiliselt seadistatud seire ning häireteavitused kettamahu ja mälukasutuse näitel“, mille juhendaja on Lauri Anton.
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

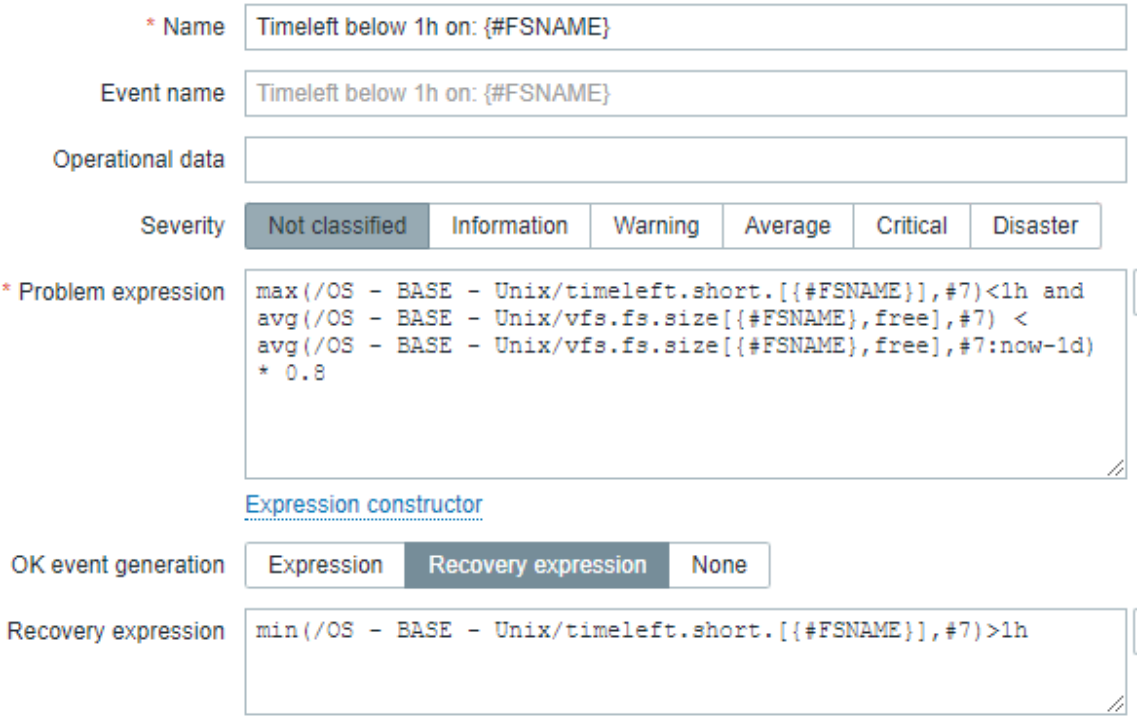
24.04.2023

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Lõpliku raamistiku trigerid kettamahtude seireks

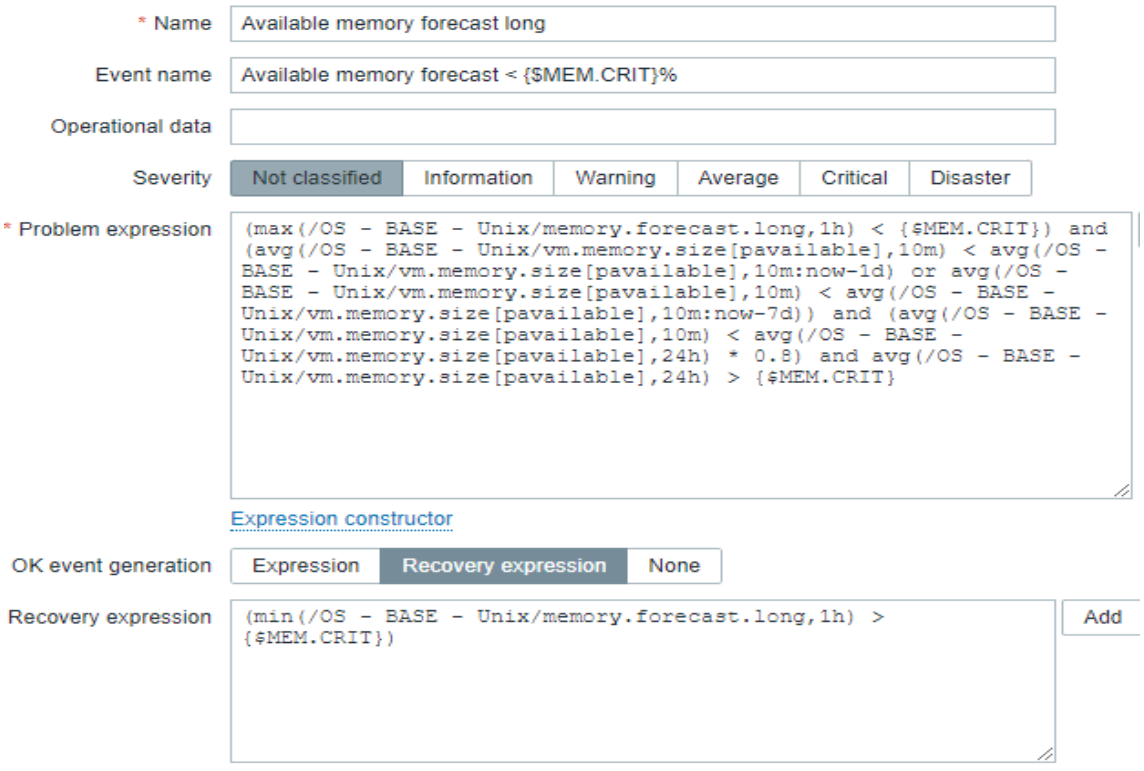


Joonis 8. Pikka perioodi kontrolliv triger kettamahtude seireks

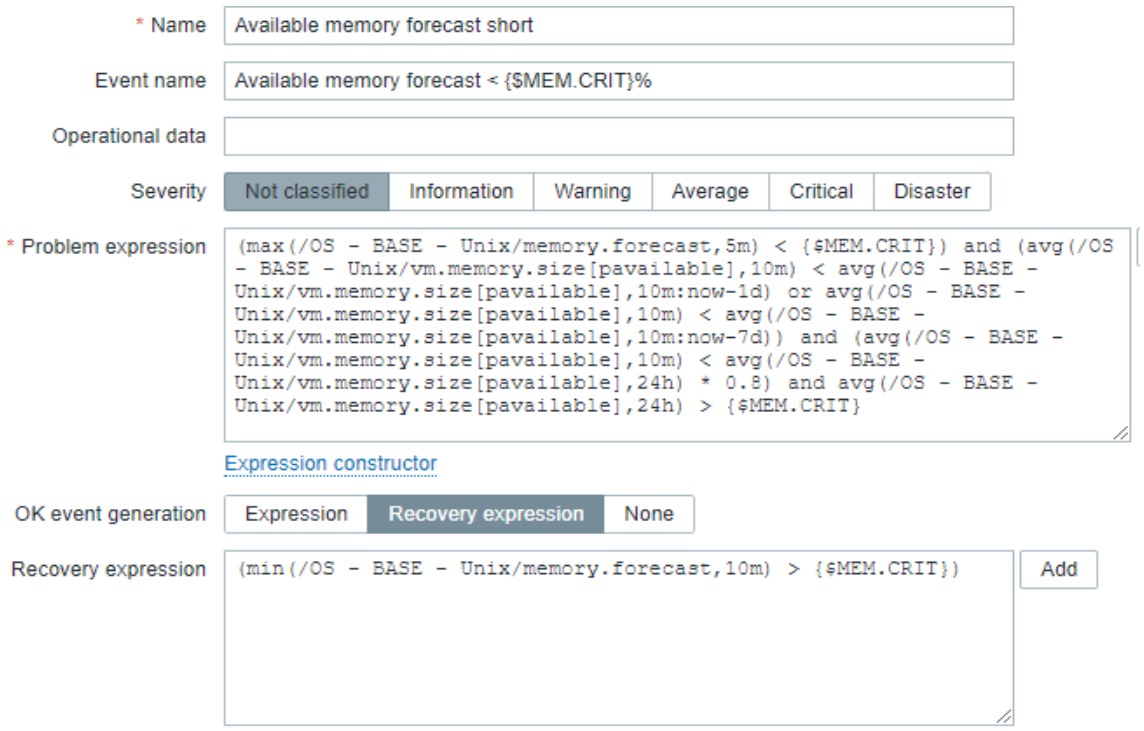


Joonis 9. Lühikest perioodi kontrolliv triger kettamahtude seireks

Lisa 3 – Lõpliku raamistiku trigerid mälukasutuse seireks



Joonis 10. Pikka perioodi kontrolliv triger mälukasutuse seireks



Joonis 11. Lühikest perioodi kontrolliv triger mälukasutuse seireks