

Univerzita Pavla Jozefa Šafárika v Košiciach

Prírodovedecká fakulta

MONITOROVANIE INFORMAČNÝCH SYSTÉMOV

BAKALÁRSKA PRÁCA

Študijný odbor:

Informatika

Školiace pracovisko:

Ústav informatiky

Vedúci práce:

Doc. RNDr. Gabriel Semanišin PhD.

Konzultant:

RNDr. Erik Bruoth PhD.

Košice 2015

Pavol Kozák

Pod'akovanie

Chcem sa poďakovať vedúcemu tejto práce Doc. RNDr. Gabrielovi Semanišinovi PhD., konzultantovi RNDr. Erikovi Bruothovi PhD. a Mgr. Slavomírovi Varchulovi za cenné rady, odbornú pomoc a konzultácie, ktoré mi poskytli pri vypracovaní tejto bakalárskej práce.

Osobitné poďakovanie patrí mojej rodine a priateľom za podporu a pochopenie.

Abstrakt

Hlavným cieľom našej bakalárskej práce je navrhnúť spôsob monitorovania informačného systému za účelom zbierania užitočných informácií ohľadom výkonu systému a používateľských preferenciách. Zaoberáme sa najmä otázkami aké informácie zbierať a akými nástrojmi ich zbierať. Existuje množstvo nástrojov z časti riešiacich našu problematiku. Zameriame sa na open-source nástroje. V práci si rozoberieme tie najpoužívanejšie, ich výhody a nevýhody a vybrané nástroje integrujeme do akademického informačného systému AIS2. Pomocou nich sa pokúsime zbierať a zobrazovať metriky, ktoré nám pomôžu identifikovať problémy a funkcie, ktoré je potreba vylepšiť.

Obsah

| | | |
|-------|---|----|
| 1 | Úvod | 5 |
| 1.1 | Motivácia | 5 |
| 1.2 | Informačný Systém | 5 |
| 1.3 | Web Aplikácia | 5 |
| 1.4 | Metrika | 6 |
| 1.4.1 | Rozdelenie Metrík | 6 |
| 1.4.2 | Aká je Dobrá Metrika? | 7 |
| 1.4.3 | Typy Metrík | 7 |
| 1.5 | Monitorovanie systému Ais2 | 7 |
| 1.6 | Problémy | 8 |
| 1.7 | Ilustračný Príklad | 8 |
| 1.8 | Prehľad Súčasného Stavú | 9 |
| 1.9 | Ciele Práce | 9 |
| 2 | Algoritmus monitorovania informačného systému | 10 |
| 3 | Typy monitorovania | 13 |
| 3.1 | Monitorovanie systému v reálnom čase | 13 |
| 3.2 | Dlhodobé monitorovanie systému | 13 |
| 3.3 | Monitorovanie systému za účelom hĺbkovej analýzy | 13 |
| 3.4 | Dlhodobé monitorovanie systému s použitím logovania | 14 |
| 4 | Nástroje na Zber Metrík | 15 |
| 4.1 | Nástroj Metrics | 15 |
| 4.2 | Nástroj Statsd | 15 |
| 4.3 | Metrics vs Statsd | 16 |
| 5 | Nástroje na Ukladanie a Zobrazenie Metrík | 18 |
| 5.1 | Nástroj Graphite | 18 |
| 5.1.1 | Výhody nástroja Graphite | 18 |
| 5.2 | Nevýhody nástroja Graphite | 19 |
| 5.3 | Nástroj Zabbix | 19 |
| 5.3.1 | Výhody nástroja Zabbix | 20 |
| 5.3.2 | Nevýhody nástroja Zabbix | 20 |
| 5.4 | Nástroj Ganglia | 21 |
| 5.4.1 | Výhody nástroja Ganglia | 21 |
| 5.4.2 | Nevýhody nástroja Ganglia | 23 |

| | | |
|-----|---|----|
| 5.5 | Graphite vs Zabbix vs Ganglia | 23 |
| 5.6 | Monitorovanie systému AIS2 v reálnom čase | 23 |
| 5.7 | Dlhodobé monitorovanie AIS2 | 23 |
| 5.8 | Vlastná práca | 23 |
| 6 | Záver | 24 |

1 Úvod

1.1 Motivácia

S informačnými systémami sa stretávame v dnešnej dobe čoraz častejšie. Mnohé systémy sa presúvajú z papierovej formy do elektronickej. Môže to byť spôsobené mnohými výhodami tohto prechodu, ako napríklad jednoduchší prístup k informáciám, alebo len modernizáciou systému kvôli pokroku doby.

V tejto informačnej spoločnosti každý túži po informáciách. S informačnými systémami sa stretávame na školách, na univerzitách, na pracoviskách, v zdravotníctve a v rôznych iných oblastiach.

Vývojári informačných systémov sa snažia poskytnúť svojim zákazníkom informácie, ktoré hľadajú, kvôli ktorým navštívili ich informačný systém. Aby vedeli, či ich systém funguje spoľahlivo, potrebujú ho nejakým spôsobom odmerať.

Na meranie systémov slúžia metriky. Dávajú nám komplexnú informáciu o systéme, o jeho nedostatkoch, o tom, čo vyžaduje zlepšenie, o rýchlosti a spoľahlivosti systému. Pri zlepšovaní systému sú metriky veľmi užitočné. Odhalia slabé miesta systému, ktoré by sme pri bežnom testovaní nespozorovali.

1.2 Informačný Systém

Informačný systém je zostavený z ľudí a počítačov, ktorí spracúvajú a interpretujú informácie. Niekedy sa tento termín používa na označenie softvéru, ktorý využíva počítačovú databázu alebo len počítačového systému.

V tejto práci sa budeme zameriavať na informačné systémy, ktoré sú vo forme webovej aplikácie.

1.3 Web Aplikácia

Webová aplikácia je aplikácia typu klient-server vytvorená pre prostredie internetu alebo intranetu. V softvérovom inžinierstve je aplikácia poskytovaná užívateľom z webového servera cez počítačovú sieť Internet, alebo jej vnútropodnikovú obdobu (intranet). Webové aplikácie sú populárne predovšetkým pre všadeprítomnosť webového prehliadača ako klienta. Ten sa nazýva tenkým klientom, pretože sám o sebe logiku aplikácie nepozná. Výhodou webových aplikácií je rovnaké užívateľské rozhranie kdekoľvek bez nutnosti inštalácie špeciálneho softvéru. Veľkou výhodou pre užívateľa

ale aj pre prevádzkovateľa aplikácie je jednoduchá aktualizácia. Tá sa vykonáva len na jednom mieste - na serveri, na ktorom webová aplikácia beží. Nevýhodou je nutnosť internetového pripojenia.

1.4 Metrika

Slovo metrika je často používaný pojem v oblasti monitorovania systémov. Metriky v oblasti monitorovania sú spôsoby a nástroje merania, vyjadrujú stav určitého systému, napríklad jeho kvality, efektívnosti a nadobúdajú pri tom rôzne hodnoty. Pri riadení sa používajú indikátory aj pre definíciu a dosahovanie cieľov prípadne ich žiaducich hodnôt. Používajú sa tiež špecializované pojmy Performance indicator alebo Key performance indicator. Metriky nám pomáhajú “odmerať” systém a tým nám dávajú komplexnú informáciu o systéme a pomáhajú odhaliť jeho slabé miesta. Vhodne zvolené metriky dokážu nasmerovať vývojárov správnym smerom a poukázať na možné problémy a tým v konečnom dôsledku pomôžu zlepšiť monitorovaný systém.

1.4.1 Rozdelenie Metriek

Medzi základné delenia metrík patrí delenie na kvalitatívne a kvantitatívne metriky. Kvantitatívne metriky udávajú číselnú hodnotu, zatiaľ čo kvalitatívne metriky nečíselnú hodnotu. Kvantitatívne metriky sa ľahšie zobrazujú, interpretujú a sú zrozumiteľnejšie.

Metriky môžeme rozdeliť podľa druhu na:

- výkonnostné metriky
- softvérové metriky
- aplikačné metriky
- systémové metriky
- používateľské metriky

V tejto práci nás budú zaujímať aplikačné, systémové a používateľské metriky.

1.4.2 Aká je Dobrá Metrika?

Aby metrika plnila svoj cieľ, teda podala správnu informáciu o systéme, mala by byť:

- jednoduchá
- relevantná
- aktuálna
- okamžite ukončiteľná

Metrika by mala byť jednoduchá, aby bola ľahko interpretovateľná a pochopiteľná. Relevantná metrika je taká, ktorá pomôže splniť cieľ, dosiahnuť to, na čo bola určená. Neaktuálne metriky nám poskytnú možno už nepravdivé dáta a preto by metriky mali byť aktuálne. Len aktuálne dáta nám popíšu systém spoľahlivo. Od metriky požadujeme, aby bola okamžite ukončiteľná, aby hneď po jej získaní bola pripravená na spracovanie a tak viedla k novým poznatkom.

1.4.3 Typy Metriek

Medzi základné typy metriek patria údaje o počte (counting), údaje o čase (timing), iné číselné hodnoty (celočíselné alebo desatinné), alebo niekedy aj nečíselné hodnoty (gauges).

Údajmi o počte môžeme získavať metriky ako napríklad počet požiadaviek, ktoré prišli na server za nejaký časový interval, počet volaní metódy v kóde, počet kliknutí na tlačidlo v informačnom systéme. Metriky tohto typu zbierame inkrementovaním počítadla v nástroji ako metrics alebo statsd v klientskej časti.

Často potrebujeme zmerať časový údaj ako napríklad čas pripojenia k databáze, čas obsluhy požiadavky alebo čas strávený vykonávaním určitého kusu kódu. Zmeraný čas zaznamenáme nástrojom na zbieranie metriek.

Niekedy je užitočné zbierať hodnotu, ktorá nezávisí od predchádzajúcich hodnôt napríklad počet objektov v rade, veľkosť dát odoslanej odpovede, alebo zaťaženie procesora. Tieto hodnoty môžu byť číselné alebo nečíselné. Tieto hodnoty pravidelne zbierame a odosielame na spracovanie a zobrazenie.

1.5 Monitorovanie systému Ais2

V našej bakalárskej práci chceme navrhnúť systém monitorovania akademického informačného systému AIS2. Problémom je to, že vývojári systému AIS2 nie sú správcami

systému. Nemôžu preto získavať metriky ako napríklad zaťaženie serverov, voľná operačná pamäť, či iné systémové metriky. Aby zabezpečili hladký chod systému, musia sa spoľahnúť na aplikačné metriky. Tie nám dajú informácie o aktuálnom stave aplikácie a o jej správaní.

Keďže systém AIS2 beží vo viacerých inštaláciách, je potrebné monitorovať viacero inštalácií tohto systému samostatne, aj ako celok. Chceme ich vedieť porovnať medzi sebou, aby sme vedeli odlíšiť problémy v konkrétnej inštalácii a globálne problémy týkajúce sa systému samotného. Na porovnanie inštalácií potrebujeme získané metriky mať na jednom mieste, na jednom serveri, čo môže výrazne skomplikovať zber metrík. Potrebujeme zobrazovať agregované metriky zo všetkých inštalácií. Bude jednoduchšie inštalácie porovnať a analyzovať nazbierané metriky.

Ďalším cieľom monitorovania AIS-u je optimalizácia interakcie systému s používateľom. Chceme detekovať zle navrhnuté dialógy, ich neprehľadné umiestnenie v systéme, a analyzovať interakciu s používateľmi, napríklad aj dynamickým upravovaním používateľského rozhrania.

1.6 Problémy

Hlavným cieľom monitorovania informačného systému je identifikovať problémy skôr ako používateľ. Ak zbadá problém používateľ, už je neskoro. Medzitým, ako nahlási problém vývojárom informačného systému, tento problém už mohlo zaznamenať množstvo ďalších používateľov. Vývojári potom musia rýchlo problém identifikovať a opraviť, čo môže byť netriviálna záležitosť. Ak by sme problém zaznamenali hneď ako nastane, mali by sme viac času na jeho riešenie.

Problém môže nastať, ak máme metrík veľa. Monitorovanie systému nesmie systém výrazne zaťažovať, lebo by monitorovanie nebolo efektívne.

1.7 Ilustračný Príklad

Dobрым príkladom informačného systému je akademický informačný systém. Monitorovanie takéhoto systému môže výrazne pomôcť k jeho vylepšeniu a rozvíjaniu.

Ak by mal vývojár informáciu o prehliadačoch, ktorými sa používatelia pripojili do systému, vedel by sa lepšie rozhodnúť, ktoré prehliadače má podporovať, aby pokryl čo najväčšiu skupinu používateľov.

Zaujímavou metrikou je aj rozlíšenie obrazoviek klientov, lebo s takouto informáciou sa dá lepšie rozvrhnúť používateľské rozhranie a zabezpečiť, aby klienti našli

informácie, ktoré potrebujú rýchlejšie. Ak by napríklad väčšina používateľov systému k nemu pristupovala s malou obrazovkou cez mobilný telefón, vývojár by investoval viac času a prostriedkov do prispôsobenia rozhrania pre mobilné telefóny.

1.8 Prehľad Súčasného Stav

V tomto čase je množstvo informačných systémov, ktoré sú neustále vylepšované a zdokonaľované a tak je potreba ich monitorovať, aby sa dali ešte viac zlepšiť a prispôbiť zákazníkovi. Každý systém je špecifický, a tak aj jeho monitorovanie je jedinečné. Vývojári sa snažia upraviť monitorovanie ich systému a prispôbiť ho tak, aby im poskytol čo najviac vedomostí o ich systéme.

Existuje niekoľko nástrojov, ktoré ponúkajú riešenia našich problémov či už ide o zbieranie metrík alebo ich vizualizáciu. Patria tu jak platené (Datadog, Stackdriver, Librato), tak aj open-source nástroje (Statsd, Metrics, Cacti, Graphite, Zabbix, Ganglia, Naggios). My sa zaoberáme open-source nástrojmi.

1.9 Ciele Práce

1. Navrhnuť spôsob monitorovania informačného systému v reálnom čase za účelom zhromažďovania informácií o výkonne systému a používateľských preferenciách.
2. Navrhnuť dátovú štruktúru pre uložené metriky z jednotlivých inštalácií a vytvoriť jednoduché štatistické prehľady nad zozbieranými údajmi v závislosti od charakteru údajov.
3. Analyzovať možnosti využitia monitorovania pre účely optimalizácie interakcie s používateľom.

2 Algoritmus monitorovania informačného systému

Ak chceme získať užitočné dáta o systéme aby sme ich mohli analyzovať, musíme ich najprv zbierať. Pred zbieraním metrík si treba najprv dobre premyslieť, čo chceme zbierať, aké metriky sú pre nás vhodné a aký cieľ chceme zberom metrík dosiahnuť. Treba si dobre a jasne definovať metriky, aby sme z nich mali čo najväčší ošoh.

Monitorovanie informačného systému od prípravy na monitorovanie až po analýzu výsledkov monitorovania sa dá popísať nasledujúcim algoritmom:

1. Určenie cieľov monitorovania informačného systému.
2. Výber vhodných metrík.
3. Výber nástrojov na monitorovanie a zobrazovanie metrík.
4. Integrácia nástrojov a metrík do informačného systému.
5. Samotný zber metrík.
6. Analýza získaných metrík.
7. Vyvodenie dôsledkov monitorovania.

Určenie cieľov monitorovania informačného systému je dôležité. Musíme si dobre premyslieť, na čo chceme monitorovanie zamerať, čo chceme zlepšiť, čo potrebujeme odmerať. Taktiež je dôležité premyslieť si, čo urobíme s výsledkom monitorovania a nakoľko bude výsledok merania relevantný a pravdivý. To závisí aj od metrík, ktoré si zvolíme, od času a obdobia, v ktorom systém budeme monitorovať. Treba si ujasniť problémy, ktoré chceme vyriešiť pomocou monitorovania.

Na to, aby sme monitorovaním dosiahli stanovené ciele, musíme si zvoliť vhodné metriky, ktoré nám poskytnú odpovede na naše otázky. Každý systém je odlišný a tak neexistuje univerzálny návod, aké metriky si zvoliť. Treba sa zamyslieť nad tým, aké údaje by nám pomohli, keby sme ich mali k dispozícii. Ak potrebujeme získať informácie o používateľoch a o ich správaní v systéme, pomohli by nám používateľské metriky. Ak potrebujeme získavať údaje o stave aplikácie, potom je pre nás vhodné zbierať aplikačné metriky. Keď chceme zistiť, aký je výkon systému a ako sa správa systém pri zaťažení a pod akou záťažou je v konkrétnych obdobiach, potom by sme sa mali zamerať na systémové metriky.

Ak už máme definované metriky, ktoré chceme zbierať, v ďalšej časti prípravy na monitorovanie si zvolíme vhodné nástroje na zber metrík a na ukladanie a zobrazovanie metrík. Môžeme si zbieranie, ukladanie a zobrazovanie metrík navrhnuť a implementovať aj po svojom, ale keďže je monitorovanie informačných systémov rozšírené, existujú už špecializované nástroje, ktoré nám pomôžu s monitorovaním a zabezpečia nám kompatibilitu s ďalšími nástrojmi. Nástrojov, ktoré máme k dispozícii, je niekoľko. V tejto práci si rozoberieme open-source nástroje na zbieranie metrík Statsd a Metrics a nástroje na zobrazovanie metrík Graphite, Zabbix a Ganglia. Všetky nástroje sú odlišné a majú svoje výhody a nevýhody, ktoré si podrobnejšie popíšeme. Nástroje si vyberieme podľa funkcionality, náročnosti integrácie a podľa veľkosti informačného systému.

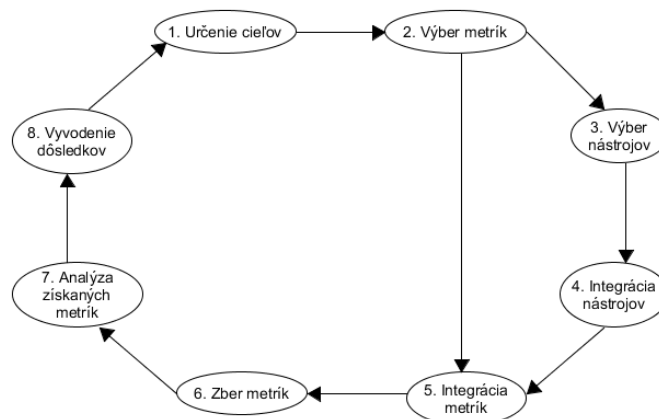
Po výbere nástrojov na monitorovanie nasleduje integrácia vybraných nástrojov a metrík do systému. Zložitosť integrácie a konfigurácie závisí od nástrojov, ktoré sme si zvolili a od komplexnosti monitorovaného systému. Metriky zbierame v podobe counterov, timerov a gauges, prípadne im podobným alebo inak pomenovaným alternatívnym spôsobom podľa vybraných nástrojov.

Ak máme nakonfigurované nástroje na zber metrík a funkčné prepojenie nástroja na zber metrík s nástrojom na ukladanie a zobrazovanie metrík, nasleduje fáza samotného zbierania metrík a monitorovania informačného systému. V tejto fáze sledujeme metriky, ktoré nám nástroj na zobrazovanie metrík vykresľuje v prehľadných grafoch. Niektoré zobrazovacie nástroje umožňujú aj agregáciu grafov pre jednotlivé metriky, to znamená, že súvisiace metriky si zobrazíme v jednom spoločnom grafe.

Niektoré nástroje, alebo ich jednoduché prepojenie s ďalším nástrojom ponúkajú možnosť notifikácie administrátora o udalostiach, ktoré nastali v systéme. Môžeme tak byť notifikovaný napríklad o prívelkej záťaži na systém, o nedostatku voľnej operačnej pamäti alebo o prídlhom čase čakania klienta na odpoveď zo servera. To nám pomôže identifikovať aktuálne nastávajúci problém, ktorý je potrebné čo najskôr vyriešiť. Takisto nám to pomôže odhaliť problém skôr, ako ho odhalí používateľ nášho systému.

Ak už zbierame metriky dlhšiu dobu, máme dostatok údajov na ich analýzu a ďalšie spracovanie. Môžeme analyzovať, ako sa systém správa v určitých obdobiach, ako sa správajú používatelia v systéme a ako sa správa systém, keď je pod záťažou. To nám pomôže pripraviť sa na ďalšiu záťaž, alebo zistíme, že potrebujeme zbierať ďalšie metriky, na ktoré sme na začiatku nemysleli, alebo sa nám zdali irelevantné.

V ďalšej fáze vyvodíme dôsledky monitorovania. Zamyslíme sa, či nám monitoro-



Obr. 1: Diagram algoritmu pre monitorovanie informačného systému

vanie pomohlo splniť ciele, ktoré sme si na začiatky stanovili a ako nám to pomohlo zlepšiť systém. Porozmýšľame, čo nám ešte treba zlepšiť, aké závery nám plynú z monitorovania a určíme ďalšie kroky k ešte lepšiemu monitorovaniu. Monitorovanie informačného systému je dlhodobý proces, a počas monitorovania sa môže meniť a upravovať s meniacím sa systémom a už získanými poznatkami z monitorovania. Môžeme pridávať nové metriky aj s novou funkcionalitou alebo tie nepodstatné metriky zmazať. V tejto fáze upravíme systém podľa výsledkov, ktoré sme získali monitorovaním. Napríklad ak sme zistili, že náš systém potrebuje optimalizáciu dopytov do databázy, optimalizujeme ich a sledujeme, aký to má dopad na systém a na rýchlosť odpovede na požiadavky od klientov.

Na obrázku 1. môžete vidieť diagram algoritmu pre monitorovanie informačného systému.

3 Typy monitorovania

3.1 Monitorovanie systému v reálnom čase

Monitorovanie systému v reálnom čase nám pomôže detekovať problém skôr, ako ho zbadá používateľ. Prehľadné grafy zobrazujúce nazbierané metriky nám ukážu dôležité informácie o stave systému. Niektoré nástroje nás dokonca upozornia, ak v systéme nastane problém. Po vyriešení problému zas z grafov vyčítame, či sme detekovaný problém vyriešili a ako dobre sme ho vyriešili.

3.2 Dlhodobé monitorovanie systému

Krátkodobé merania systému nám podajú užitočnú informáciu o aktuálnom stave systému a o jeho používateľoch. Nás však zaujímajú aj dlhodobé merania. Tie nám povedia, kedy v priebehu roka je systém najviac zatažený, kedy najmenej, a aké funkcionality používateľa využívajú v konkrétnych obdobiach. To sa nám pomôže pripraviť na najväčšiu záťaž.

Po vylepšení niektorej funkcionality v systéme vieme porovnať nové hodnoty metrick s historickými hodnotami metrick pre danú funkcionality. Dozvieme sa tak, či sme funkcionality naopak nezhoršili a či bude schopná vydržať záťaž.

3.3 Monitorovanie systému za účelom hĺbkovej analýzy

Dlhodobými meraniami vieme zaznamenávať vzory správania používateľov v našom informačnom systéme. Takýmto vzorom môže byť v akademickom informačnom systéme napríklad:

- Používatelia si najčastejšie po prihlásení do aplikácie prečítajú nové správy.
- Po prečítaní správ sa odhlásia alebo:
 - Ak je začiatok semestra, pozrú si rozvrh hodín.
 - Ak je stred semestra, pozrú sa na priebežné hodnotenie.
 - Ak je koniec semestra, prihlásia sa na skúšku.

Ak získame takéto vzory, môžeme napríklad dynamicky meniť používateľské rozhranie počas priebehu semestra a sprístupniť tak najčastejšie hľadané informácie podľa vzorov na viditeľnejšie miesta v našom systéme.

Dlhodobými meraniami systému vieme získať aj ďalšie zaujímavé informácie. Nad množstvom vyzbieraných metrík sa dá robiť hĺbková analýza v podobe dolovania dát. No na hĺbkovú analýzu budeme potrebovať väčšinou odlišné dáta od tých, ktoré pre nás majú zmysel pri monitorovaní v reálnom čase. Budú nás zaujímať iné metriky a aj ciele takejto analýzy budeme mať odlišné.

3.4 Dlhodobé monitorovanie systému s použitím logovania

4 Nástroje na Zber Metriík

Ak už máme jasne definované metriky, môžeme ich začať zbierať. Zber metriík je často riešený problém, v zbere metriík nám môžu pomôcť rôzne nástroje. Medzi najpopulárnejšie open-source nástroje na zber metriík patria Metrics a Statsd. Sú to nástroje určené na odosielanie nameraných metriík do nástrojov na uloženie a zobrazenie metriík v podobe grafov.

Integrácia oboch nástrojov si však vyžaduje určité zmeny v kóde systému. Každý systém je jedinečný a teda aj jeho monitorovanie je odlišné. To si však žiada odlišné metriky a tie sa zbierajú na odlišných miestach v kóde systému.

Oba nástroje sú odlišné z hľadiska integrácie a architektúry. Naším čiastočným cieľom je porovnať ich. V nasledujúcich odstavcoch si popíšeme ich výhody a nevýhody.

4.1 Nástroj Metrics

Metrics je open-source java knižnica na zber metriík. Získané metriky posielajú do nástrojov na zobrazovanie metriík ako napríklad Graphite alebo Ganglia, alebo ukladá do databázy prostredníctvom logovacích nástrojov.

Keďže Metrics je java knižnica, integrácia nástroja do java webovej aplikácie je záležitosť pridania niekoľkých závislostí. Následne definujeme registre, v ktorých budú metriky uložené. Ďalej je potrebné nastaviť reportovanie do ďalších nástrojov. Potom na vhodných miestach v aplikácii získame metriky pomocou counterov, timerov a gauges, ktoré chceme zbierať. Nástroj Metrics nám pomôže dostať namerané metriky do ďalších nástrojov.

Asi najväčšou konkurenciou k Metrics je nástroj Statsd.

4.2 Nástroj Statsd

Open-source nástroj Statsd je určený na zber metriík a následné preposlanie do nástrojov na spracovanie a zobrazenie. Má klientov pre množstvo programovacích jazykov ako napríklad node.js, java, python, ruby, perl, php, c, cpp, .net, či dokonca plugin pre wordpress.

Statsd klient na rozdiel od Metrics posielajú získané metriky cez sieťovú komunikáciu na otvorený UDP port na oddelený statsd server, ktorý ich prijíma, akumuluje a v nastavených časových intervaloch odosiela do ďalších nástrojov. Služí ako rýchla cache

pre metriky. Umožňuje posielat' dáta do nástrojov ako napríklad Graphite, Ganglia, Librato, Zabbix, či Mongo, Mysql, Datadog a mnoho ďalších.

Architektúra tohto nástroja je trochu zložitejšia.

4.3 Metrics vs Statsd

Obe nástroje slúžia na zbieranie metrík, majú však rozdielnú architektúru. Ponúkajú rôzne možnosti integrácie s ďalšími nástrojmi. Kvôli prehľadnosti sú základné rozdiely popísané v tabuľke 1.

Tab. 1: Porovnanie nástrojov Metrics a Statsd

| Atribút porovnania | Metrics | Statsd |
|--|--|--|
| Architektúra | Server totožný s klientom | Server a klient oddelený |
| Integrácia s ďalšími nástrojmi | Ganglia, Graphite, Librato, Spring, Sematext, Wicket, Guice, Scala, Clojure, Cassandra, Elasticsearch, Statsd, Datadog, Influxdb, Cdi, Aspectj, Apache Camel | Amqp, Ganglia, Librato, Socket.io, Statsd, Mongo, Mysql, Datadog, Monitis, Instrumental, Hosted graphite, Zabbix, Opentsdb, Influxdb, Stackdiver, Couchdb |
| Implementácia Servera | Iba Java | Node.js, Python, Ruby, C, Go, Clojure, Perl |
| Implementácia Klienta | Klient totožný so serverom | Node.js, Java, Python, Ruby, Perl, Php, Clojure, C, Cpp, .Net, Go, Apache, Io, Varnish, PowerShell, JavaScript, Cocoa, ActionScript, Wordpress |
| Licencia | Apache 2 | MIT |
| Zložitosť integrácie do java web aplikácie | Import knižnice, definovanie metrík, nastavenie reportovania | Inštalácia Node.js, Statsd servera, import Statsd klientskej knižnice, definovanie metrík, inštalácia reportovacích backendov, konfigurácia Statsd servera |
| Počet prispievateľov | 127 | 154 |
| Počet releasov | 51 | 11 |
| Prvý commit do repozitára | 27.4.2011 | 30.12.2010 |
| Počet commitov do repozitára | 1958 | 823 |

5 Nástroje na Ukladanie a Zobrazenie Metrík

Nástroje na zbieranie metrík majú často za úlohu agregovať metriky a preposielať ich do ďalších nástrojov. Existujú rôzne nástroje na ukladanie a zobrazenie metrík. Niektoré z nich sú komerčné, iné open-source. Medzi často používané open-source nástroje patria nástroje Graphite, Zabbix a Ganglia. Tieto nástroje vedia prijať dáta z iných nástrojov, uložiť ich v databáze a vhodným spôsobom ich zobrazíť, aby sa dali ľahšie interpretovať.

Rôzne nástroje na zbieranie metrík majú rôznu architektúru a ponúkajú rozličnú funkcionality. Záleží od veľkosti, komplexnosti systému a od požadovanej funkcionality, ktorý nástroj si vyberieme.

Metriky sa najčastejšie zobrazujú v podobe histogramov alebo čiarových diagramov cez používateľské rozhranie vo forme webovej aplikácie.

5.1 Nástroj Graphite

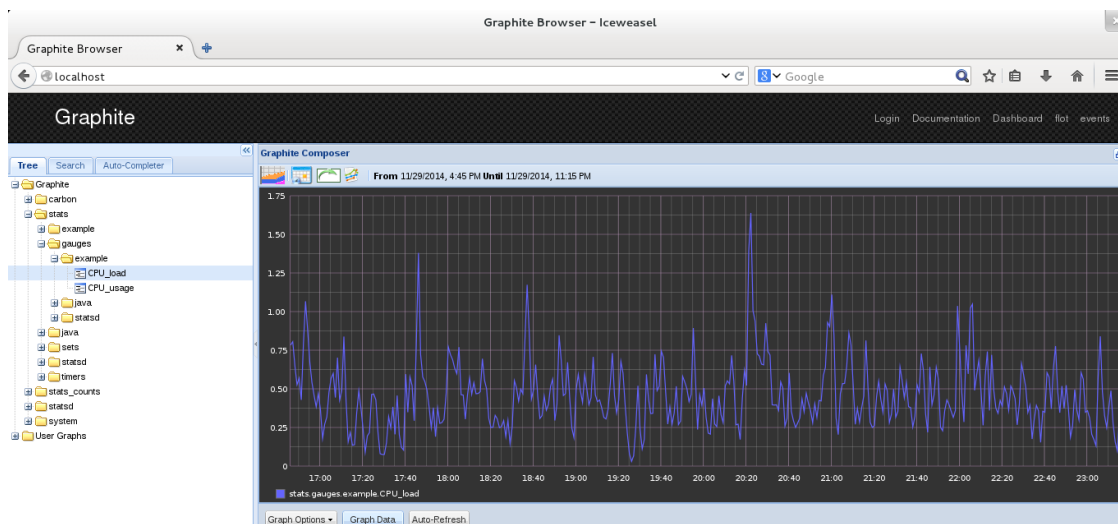
Nástroj Graphite je veľmi obľúbeným nástrojom v oblasti zobrazovania metrík. Nástroje ako napríklad Statsd alebo Metrics pošlú získané metriky do Graphitu a ten ich uloží pomocou integrovaného nástroja Carbon do Whisper databázy. Whisper databáza je veľmi podobná RRD (round robin database) databáze, ale je prispôbená a napísaná v pythone. Kód graphitu je napísaný v programovacom jazyku python. Metriky stačí poslať do nástroja Graphite a on ich sám zaradí do stromu metrík podľa prefixu. Po inštalácii sa nevyžaduje žiadna dodatočná konfigurácia.

Na obrázku 2 môžete vidieť používateľské rozhranie nástroja graphite a čiarový diagram pre metriku zaťaženie procesora.

Nástroj Graphite je jednoduchý open-source nástroj určený na ukladanie a zobrazovanie metrík v podobe grafov. Na vykresľovanie grafov využíva python knižnicu PyCairo. V praxi sa používa pri jednoduchých projektoch, ak nám postačuje získané metriky zobrazovať. Pri monitorovaní komplexnejších systémov sa používajú zložitejšie nástroje ako napríklad Zabbix alebo Ganglia.

5.1.1 Výhody nástroja Graphite

Graphite je jednoduchý nástroj určený pre monitorovanie malých jednoduchších systémov. Hlavnou výhodou je, že po inštalácii nástroja sa nevyžaduje žiadna konfigurácia metrík. Ďalšou výhodou je jednoduché zobrazenie viacerých metrík v jednom grafe.



Obr. 2: Používateľské rozhranie nástroja Graphite

Graphite je zameraný na monitorovanie systémov v reálnom čase.

5.2 Nevýhody nástroja Graphite

Nástroj Graphite je zameraný len na vykresľovanie metrík a neponúka žiadnú dodatočnú funkcionálnosť. Jeho nevýhodou je, že sa má komplikovanú inštaláciu.

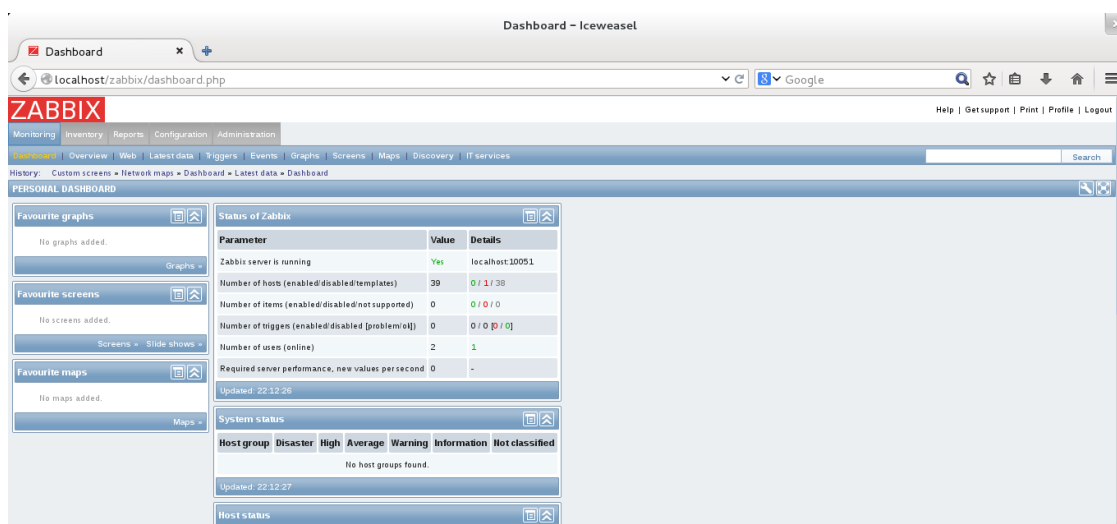
5.3 Nástroj Zabbix

Ďalším používaným nástrojom je Zabbix. Jeho kód je napísaný v jazyku C a používateľské rozhranie v podobe webovej aplikácie je napísané v jazyku PHP. Na ukladanie metrík používa Mysql databázu. Na obrázku 3 môžete vidieť používateľské rozhranie nástroja Zabbix.

Zabbix je komplexný nástroj určený aj pre väčšie systémy. Okrem vizualizácie dokáže zbierať vlastné metriky, pomáha detekovať problém a vie o ňom notifikovať administrátorov. Je to omnoho mohutnejší nástroj ako Graphite.

K informačnému systému patrí aj hardvér, na ktorom systém beží. Informácie o hardvéri a rôzne iné systémové metriky môžeme zbierať pomocou modulu Zabbix agent. Zabbix agent pravidelne posiela získané údaje zo stroja, kde je nasadený na Zabbix server. Môžeme tak jednoducho monitorovať napríklad výkon systému, sieťové zariadenia, virtuálne stroje, databázy, Java virtual machine alebo Java aplikačné servery.

Zabbix tiež umožňuje na zber dát použiť SNMP a IPMI agentov alebo Zabbix



Obr. 3: Používateľské rozhranie nástroja Zabbix

sender nástroj, ktorým do Zabbixu vieme poslať ľubovoľné metriky z nástrojov na zbieranie metrík.

Zabbix môže slúžiť aj na detekciu problémov so systémom. Detekcia problémov v Zabbixe funguje pomocou definovania triggerov, ktoré odhalia, že niečo nefunguje. Môže to byť napríklad výpadok stroja, málo voľného miesta na disku monitorovaného zariadenia alebo veľa pripojených používateľov. V triggeroch sa nastaví prah, ktorým určíme maximálnu akceptovateľnú hodnotu metriky.

Ak sa táto hodnota prekročí, Zabbix nás o tom môže notifikovať alebo vykonať nastavenú akciu. Môže nás notifikovať prostredníctvom e-mailu, SMS správy alebo správy cez Jabber. Môžeme si tiež nastaviť, aké dáta sa nám v notifikácii zobrazia. Okrem notifikačnej správy Zabbix vie po detekcii problému vykonať napríklad shell príkaz cez ssh.

5.3.1 Výhody nástroja Zabbix

Veľkou výhodou nástroja Zabbix je to, že na ukladanie metrík používa Mysql databázu. Dáta tak máme uložené v relačnej databáze a vieme sa na ne jednoducho dopytovať.

5.3.2 Nevýhody nástroja Zabbix

Hlavná nevýhoda Zabbixu je to, že každá metrika, ktorú chceme monitorovať musí byť dopredu nakonfigurovaná. Ak máme metrík veľa, úvodná konfigurácia je pracná.

Pri každej novej funkcionalite informačného systému s predpokladom, že ju chceme monitorovať, je potrebné nakonfigurovať nové metriky aj v Zabbixe.

5.4 Nástroj Ganglia

Dalším obľúbeným nástrojom je Ganglia. Ganglia je open-source nástroj zameraný na distribuované monitorovanie celých clustrov či gridov. Je naozaj rozšírený, používajú ho napríklad známe spoločnosti ako Twitter, flickr, National Aeronautics and Space Administration (NASA), Wikipedia, CERN, Cisco, HP, Microsoft, Dell, nVidia, U.S. Air Force.

Metriky do Ganglie môžeme poslať pomocou Gmond modulov, Gmetric modulu alebo inými nástrojmi, ako napríklad statsd. Ak posielame metriky do Ganglie, nemusíme ich mať dopredu nakonfigurované. Gmetad vytvorí nový RRD súbor pre každú novú metriku, čo značne uľahčí konfiguráciu, ak metrík máme veľa. Web používateľské rozhranie potom zobrazí základný graf pre každú metriku.

Ganglia na posielanie metrík na centrálny server používa dvoch démonov a to Gmond a Gmetad. Gmond môže posilať alebo prijímať metriky, ktoré si drží v pamäti. Gmetad v rámci gridu ťahá dáta z jedného gmond démona v rámci gridu. Gmond moduly so sebou komunikujú cez UDP, do Gmetad posielajú XML cez TCP. Gmetad si medzi sebou posielajú metriky cez TCP vo forme XML. Gmond medzi sebou komunikujú buď unicastom alebo multicastom. Architektúra pomocou unicastu je popísaná na obrázku 4.

Ganglia priamo nepodporuje detekciu problémov so systémom. Umožňuje však prepojenie s nástrojom Naggios, ktorý je na to určený.

5.4.1 Výhody nástroja Ganglia

Asi najväčšou výhodou nástroja Ganglia je to, že dokáže monitorovať celé clustre a gridy. Je určená na agregáciu metrík z viacerých serverov a z komplexných informačných systémov. Na ukladanie metrík používa RRD databázu, takže metriky nemusíme v Ganglii konfigurovať. Výhodné je aj jednoduché prepojenie s nástrojom Naggios.

5.4.2 Nevýhody nástroja Ganglia

5.5 Graphite vs Zabbix vs Ganglia

Ako sme už písali, nástroj Graphite je vhodný na monitorovanie menších systémov. Pre väčšie informačné systémy je lepšie použiť komplexnejší nástroj, ako napríklad Zabbix alebo Gangliu. Tie nám ponúkajú viac užitočných funkcií, ktoré môžeme využiť. Zabbix vie detekovať problémy a tie jednoduchšie aj vyriešiť (napríklad reštartom zlyhanej služby), Ganglia zas dokáže monitorovať celé clustre a gridy a ponúka nám jednoduchú agregáciu grafov. Záleží od komplexnosti systému, ktorý nástroj si vyberieme.

5.6 Monitorovanie systému AIS2 v reálnom čase

5.7 Dlhodobé monitorovanie AIS2

5.8 Vlastná práca

6 Záver

V tejto práci sme sa zaoberali monitorovaním informačných systémov, prečo je monitorovanie výhodné a ako nám môže pomôcť pri jeho rozvíjaní. Rozobrali sme si open-source nástroje na zbieranie a vizualizáciu metrík a to Statsd, Metrics, Graphite, Zabbix a Ganglia.