OPTIMALIZACE CLOUDOVÝCH INFORMAČNÍCH SYSTÉMŮ S VYUŽITÍM AJAX

Ondřej Ryška
Ostravská univerzita, Trojanovice Lomná 551, 777987335, p12228@student.osu.cz

Abstrakt
Cloud computing je dnes velice populární pojem v oboru IT. Jde v podstatě o technologii, která zajišťuje přístup k aplikacím, které jsou ve skutečnosti umístěny jinde než v počítači nebo zařízení uživatele aplikace. Webové aplikace dnes často využívají technologii AJAX pro komunikaci klienta a serveru na pozadí stránky. Diplomová práce na toto téma se zabývá hledáním vhodného využití technologie AJAX co se týče výkonnosti aplikace. Nejprve je měřen běh aplikací různě využívajících technologii AJAX a na základě těchto měření jsou jednotlivé implementace analyzovány a porovnávány.

Klíčová slova: *cloud computing, AJAX, informační systém, optimalizace.*

Úvod

„Cloud computing je v zásadě koncepce, která umožňuje přistupovat k aplikacím, jež jsou ve skutečnosti umístěny jinde než v místním počítači nebo zařízením připojeném k Internetu.“ [1, str. 24]. Podle typu poskytování prostředků pro cloud computing jej dělíme na tři základní typy: PaaS, IaaS, SaaS [2]. Mezi nejznámější služby a firmy cloud computingu patří například Google Apps, Google App Engine, Windows Azure, Amazon, VMware.

Technologie AJAX je dnes u webových aplikací velmi populární. Pomocí funkcionality prohlížeče je možné komunikovat se serverem a tak například odesílat nebo přijímat data na pozadí webové stránky. Není tedy třeba vykonat potřebnou komunikaci klasickým postupem přes požadavek GET nebo POST. Proto může být aplikace snadněji a pohodlněji ovládána.

Cloud a AJAX využívá a je spojen s několika standardy a technologiemi. Jsou to například HTML, JavaScript, SSL, JSON, XML.

Tento článek představuje výsledky měření běhu tří aplikací využívajících AJAX nebo klasický postup pomocí GET nebo POST. Měření je realizováno softwarem JMeter [3]. Programem JMeter jsou generována množství požadavků a JMeter poskytuje různé užitečné statistiky. Měření je v diplomové práci prováděno na desktopu a mikropočítači Raspberry PI [4]. V tomto příspěvku je však kvůli omezení rozsahu prováděno měření pouze na desktopu.

Testovaná aplikace

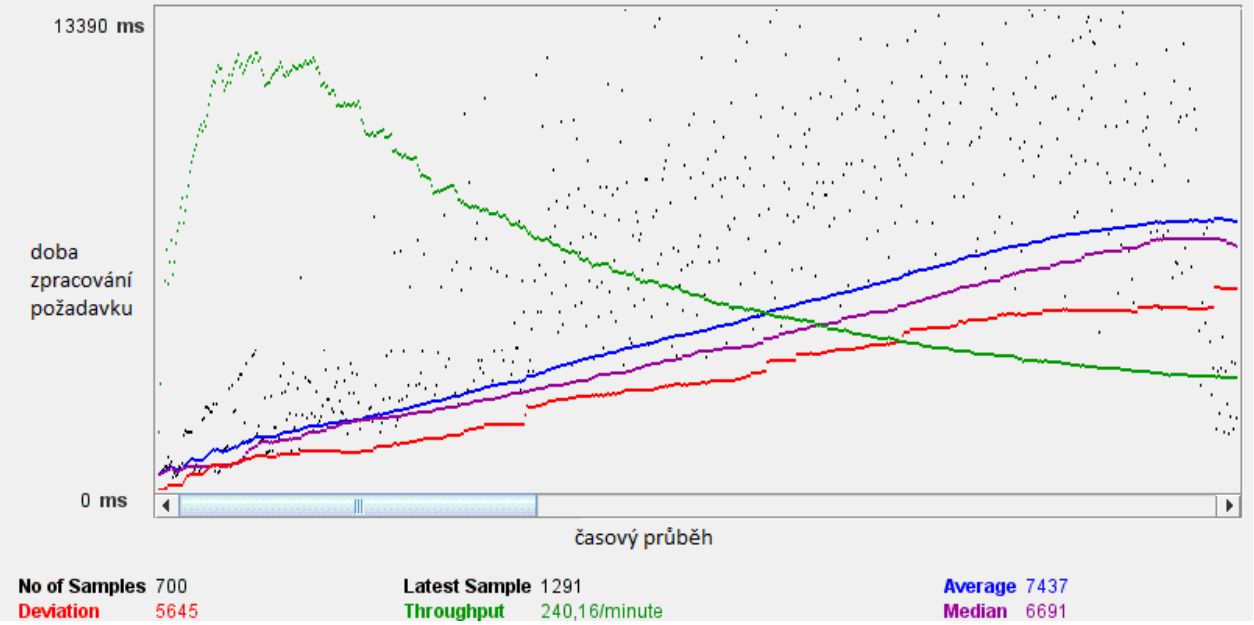
Testované aplikace jsou napsány v jazyce Java a využívájí Spring MVC Framework. Měření probíhá na jednoduché funkcionalitě - uložení osoby do databáze a jejich následné přidání do seznamu na stránce. Tato funkcionalita je realizována pomocí třech různých přístupů. Prvním je odeslání klasického požadavku typu POST, následné uložení osoby do databáze a vyrenderování celé webové stránky s již obnoveným seznamem osob.

Druhý typ aplikace již AJAX využívá. Data se serveru odešlou přes AJAX, avšak samotné přidání osoby do seznamu je realizováno okamžitým připojením osoby do seznamu pomocí jQuery, aniž by bylo nutné čekat na odpověď serveru.

Poslední řešení již využívá AJAX jak pro uložení osoby, tak také pro získání seznamu osob ze serveru. Data jsou přes AJAX odeslána na server, tam uložena, poté server odešle klientovi celý aktualizovaný seznam osob a nahradí jím na stránce seznam stávající.

Měření a analýza výsledků

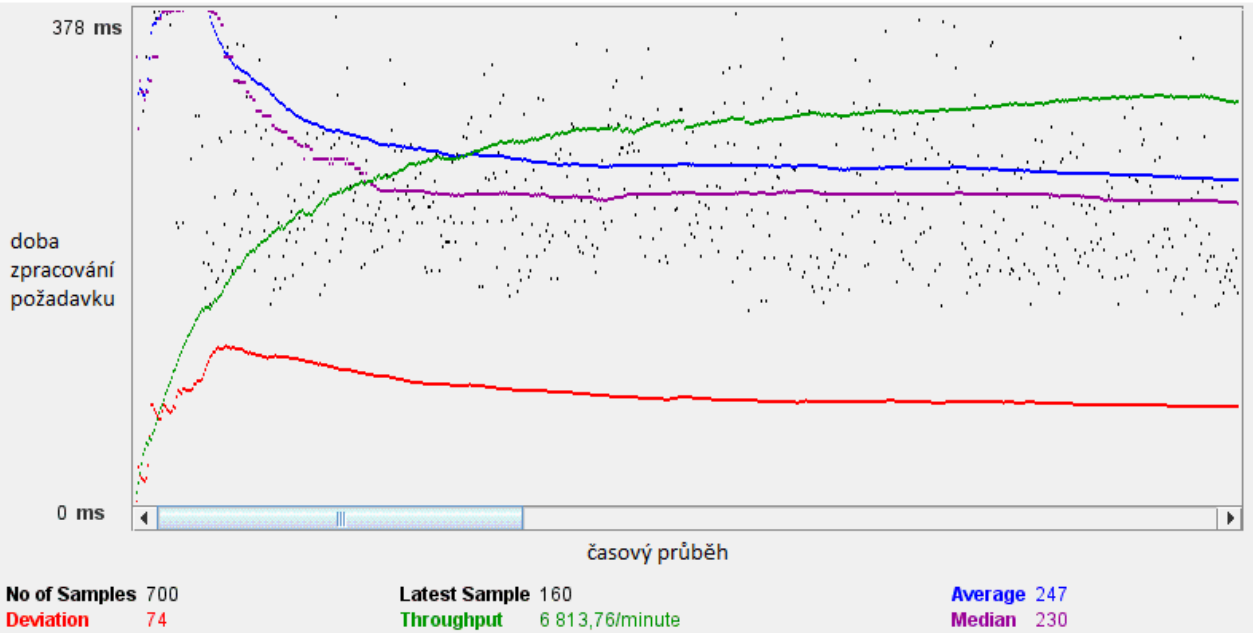
Měřící test simuluje 35 uživatelů odesílajících požadavek v jednom okamžiku. Toto se opakuje 20 krát. Dohromady tedy proběhne pro měření 700 komunikací se serverem. První měření proběhlo na implementaci bez technologie AJAX, tedy klasický požadavek typu POST. Následuje výsledný graf.



Obrázek 1. Výstup měření – požadavek POST

Asi nejpodstatnější jsou udáje *Average* a *Throughput*. Hodnota *Average* definuje průměrnou dobu zpracování požadavku. *Throughput* udává propustnost, tedy množství zpracovaných požadavků za minutu. Výše uvedený graf má hodnoty *Average* 7437 a *Throughput* cca 240 za minutu, což při porovnání s ostatními implementacemi, není příliš dobrý výsledek. V grafu lze pozorovat pozvolný nárust hodnoty *Average*. To je způsobeno narůstající velikostí vraceného seznamu osob. Čím je větší seznam osob, tím déle trvá jeho přenos a tím je tedy odezva pomalejší. Logicky naopak klesá propustnost, tedy *Throughput*. S narůstajícím seznamem osob se snižuje počet obsloužených požadavků za minutu. Propustnost na začátku průdce stoupá, což je způsobeno postupným generováním čím dál většího počtu uživatelů a požadavků.

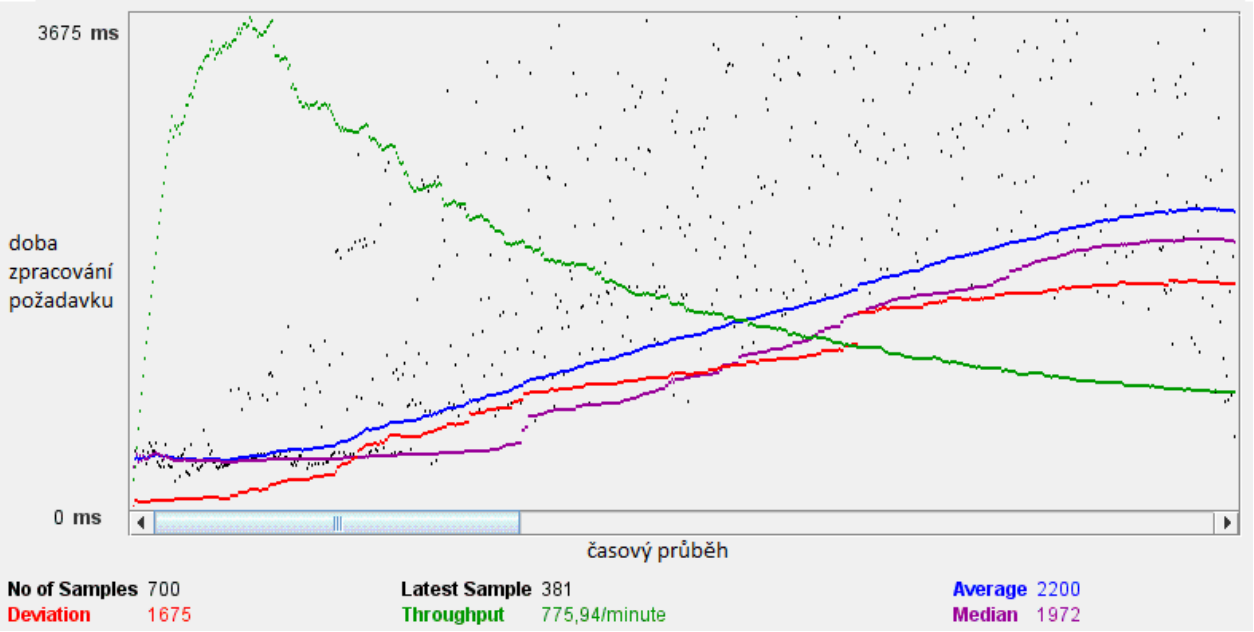
Následuje graf měření implementace s využitím AJAX jen pro odeslání dat na server. Seznam se doplní o novou osobu připojením osoby na konec seznamu pomocí JQuery.



Obrázek 2. Výstup měření – AJAX jednosměrně

Hodnoty jsou *Average* a *Throughput* jsou nyní v porovnání především s prvním grafem mnohonásobně lepší. V tomto ohledu tedy vypadá tato implementace jako rychlejší. Dokonce i nehrozí problém se zvyšováním *Average* s přibývajícím počtem požadavků z důvodu nárustu seznamu osob. V tomto případě se totiž tento seznam neodesílá ze serveru, jelikož je dynamicky aktualizován u klienta.

Následuje graf měření implementace plně využívající AJAX. Tedy jak pro uložení seznamu, tak pro získání aktuálního seznamu ze serveru.



Obrázek 3. Výstup měření – AJAX s odpovědí

Hodnoty *Average* a *Throughput* jsou sice lepší než u první implementace pomocí POST, avšak nedosahují hodnot pro druhou implementaci s jednosměrnou komunikací přes AJAX. Opět lze pozorovat nárust *Average* a logicky pokles *Throughput*. Opět je to z důvodu zvětšování seznamu osob.

Závěr

Dle výše vedených grafů lze jednoznačně prohlásit, že nejvhodnější implementace co se týče rychlosti zpracování, je druhá implementace využívající AJAX pouze pro odeslání dat na server. Naopak nejpomalejší je první aplikace využívající požadavek POST.

Velký vliv na rychlost měl také v první a třetí implementaci narůstající seznam osob odesílaný ze serveru. Druhá aplikace tímto ovlivněna není. Byla testována i varianta, kdy se po přidání osoby neaktualizuje seznam osob. Pouze se inkrementuje hodnota definující počet osob v databázi. Tím odpadá odesílání dlouhého seznamu klientovi a s tím spojené postupné zpomalování odezvy. Měření ukázalo, že první a třetí implementace se výrazně zrychlily. Avšak první aplikace využívající POST zůstává stále mnohonásobně nejpomalejší.

Lze tedy konstatovat že AJAX může výrazně zrychlit běh webových aplikací.

Poděkování

Děkuji panu RNDr. Jaroslavu Žáčkovi, Ph.D. za vedení diplomové práce a pomoc při tvorbě tohoto příspěvku.

Literatura

[1.] **Anthony T. Velte, Toby J. Velte, Robert Elsenpeter. 2011.** *Cloud computing - praktický průvodce.* [překl.] Jakub Goner. Brno : Computer Press, 2011. ISBN 978-80-251-3333-0.

[2.] IaaS, PaaS, SaaS (Explained and Compared). *Apprenda.* [Online] [Citace: 3. 4 2014.] http://apprenda.com/library/paas/iaas-paas-saas-explained-compared/.

[3.] *Apache JMeter.* [Online] [Citace: 31. 3 2014.] http://jmeter.apache.org/index.html.

[4.] *Raspberry PI.* [Online] [Citace: 20. 3 2014.] http://www.raspberrypi.org.

Abstract

Cloud computing is very popular theme in IT today. It is technology, that ensure access to application, which are placed elsewhere than in users pc or device. Web application often use AJAX for comunation client and server on background of web page. Thesis be occupied finding right way of using AJAX for performance. At first is measures run applications with different AJAX using. Results of measures are use for analysis different implementation AJAX technology.