



Performance Test Analýza

by Canarytrace

PTA

Performance Test Analýza (PTA) je souhrn požadavků a rizik pro performance testy (PT) , včetně detailu nastavení performance testů, závislostí, požadavků na testovací data, prostupy, monitoring testovacího prostředí, popis testovacích případů a jejich priorit, zúčastněné osoby, provoz performance testu, plánování a dalších prerekvizit pro úspěšné provedení performance testu. Protože jsou PT při přípravě složitou disciplínou je jejich životní cyklus rozdělen do několika fází a každá z nich si vyžaduje pozornost a dostatek pečlivosti. Aby výsledky PT byly vypovídající, je potřeba součinnost zúčastněných týmů / garantů. Proto je potřeba vytvořit PTA ještě před vývojem PT, která performance testera provede sběrem důležitých informací a poukáže na možná rizika spojená s test designem a test exekucí PT.

Přečtěte si naše studie: <https://medium.com/@canarytrace>



Sběr vstupů

Oblast	Popis	Hotovo, popř. kým a do kdy
Objednávateľ / Zadavateľ		
Technický garant	Při nejasnostech se na něj lze obrátit pro další nasměrování. Zpravidla jde o dev či architekta, který poskytne high-level pohled nad problémem.	
Realizační tým	Pro zjištění detailů fungování programové funkce.	
Cíl Performance Testu	Očekávaný přínos PT.	
Školení	Zaškolení do testované aplikace a dalších systémů, včetně interpretace logů a grafů.	
Specifikace a Testovací případy	Business specifikuje business cases, které má PT zatěžovat a přiřadí jim prioritu. Verifikuje se architekty a vývojáři. Zároveň je potřeba znát produkční profil + včetně budoucích odhadů.	
Časování	Deadline dodání výsledků za PT, doporučená / domluvená okna pro spuštění PT s ohledem na zajištění podpory a bližícího se termínu release. Test exekucí PT může být více pokud se PT vytváří během vývoje, nebo pokud jde o retests.	
Analýza a návrh Performance Testu	Technická specifikace způsobu provedení PT včetně doprovodného SW. Co přesně a jak bude PT zatěžovat.	
Rizika při přípravě PT	Pro PT existují rizika spojená se security, certifikáty, chybějící dokumentací, testovací data, data servery atp.	
Nastavení modelu zátěže PT	Ze zadání vyplýne, jak bude PT zatěžovat testovanou aplikaci, kolik transakcí se má paralelně provést po jakou dobu a to včetně Think Time. Zde je potřeba přihlédnout i na doprovodný background noise.	
Testovací data	Jak získat testovací data, jejich popis, mechanismus vytvoření, znova použitelnost, recyklaci, uchovávání, degradaci v čase	



Oblast	Popis	Hotovo, popř. kým a do kdy
Akceptační kritéria / NFR	Úspěšnost PT hodnotíme podle toho, zda naplnil očekávání, tedy zatěžování business scénářů podle zadání a přitom jsou NFR a využité zdroje akceptovatelné.	
Prostupy a přístupy	Generátory záťaze potřebují prostupy na IP a port, kde je nasazena testovaná aplikace, stejně tak jsou potřeba přístupy do dalších nástrojů pro monitoring či vyčítání logů.	
Logy a Monitoring	Testovaná aplikace a další prvky v cestě správně logují a stejně tak jsou nasazené sondy pro měření zdrojů.	
Testovací prostředí a podpora	Testovací prostředí je adekvátní pro PT nebo lze s produkčním korelovat a zdroje jsou neměnné. Je zajištěna podpora pro případný pád testované aplikace či infrastruktury pod záťaze PT, nebo pro investigaci zatěžovaných oblastí.	
Odhady	Jsou hotovy odhady a schváleny	
Reporting	Report bude ve zvoleném rozsahu (zpravidla standardizovaný test report), live-reporting do elasticsearch, existuje confluence page, kam ukládat reporty a víme komu report zaslat	
Omezení	Jsou známi rizika a omezení týkající se vývoje a exekuce PT. Zadavatel s omezeními souhlasí a pokračuje se s vývojem a exekucí PT.	
Model záťaze pro Web Performance Testy (WPT)	Pokud jde o PT webové aplikace, je potřeba paralelně spouštět WPT pro monitoring a měření nefunkcionálních charakteristik browseru.	

K zamýšlení (při přípravě PTA)

- Je dostupný nějaký benchmark / jiné testy a pokud ano, jaké jsou funkcionální a popř. nefunkcionální výsledky?
- Na jakém prostředí proběhne test exekuce PT? Budou potřeba prostupy?
- Je aplikace již dostupná, pro exploratory, její analýzu, poukázaní na rizika a přípravu odhadů?
- Jaký je deadline a kdy proběhne test exekuce PT?
- Bude se provádět test exekuce PT vícekrát s rozdílným nastavením? Nebo proběhne test exekuce baseline PT za účelem komparace?
- Je prostředí na kterém proběhne test exekuce PT sdílené s jinými aktivitami systémů, které mohou ovlivnit klidový stav testovacího prostředí?
- Jaký je cíl PT? Může jít o akceptaci, nebo třeba porovnání nového testovacího prostředí s původním, či změna aplikačního serveru?
- Jsou logy aplikace dostupné? Jaké je URL a jak logy interpretovat? (Jak vypadá vyhledávací syntaxe, či popis dashboardů. Krom aplikačního serveru je dobré myslit i na logy například proxy serveru / balanceru, databází atp.)
- Jaké jsou SLA/NFR pro PT? TTFB, ElapsedTime, propustnost, chybovost, využití zdrojů aplikačního serveru, databází atp.
- Jaké jsou NFR pro WPT? Využitý JS Heap, ResponseTimes, TTFB, Web Vitals metriky atp.
- Jaké jsou Test Cases a jaké rozhraní pokrývají (Web, Rest-Api, SOAP, MQ, DB, atp.)



- Jak vypadají requesty a očekávané odpovědi. Je dostupný popis rozhraní včetně možnosti ručního provolání zamockovaných endpointů.
- Jaké jsou požadavky na mix zátěže, jak mají být nadefinovány ThreadGroups včetně VU/req./s + ThinkTime? Jak dlouho má test trvat a jak má vypadat nárůst zátěže?
- Je potřeba před nebo po PT (zvlášť pokud jde o POSTové operace) uvést aplikaci či prostředí nebo testovací data do výchozího stavu? Je možné, že při smoke testech se použitá testovací data znehodnotí? Lze stále dokola používat stejná testovací data?
- Mohou smoke testy způsobovat omezení pro další uživatele aplikace na testovacím prostředí (například testeři) nebo nadměrným logováním aplikace?
- Je potřeba před PT či během něj spustit nějakou dávkovou operaci (někam nalít data, nebo například stahovat nějaký soubor?)
- Jak vypadá background noise a jak se bude simulovat při PT?
- Používá testovaná aplikace, testovací prostředí či předřazené prvky ochrany proti DDoS, F5 či cache? Lze jí před PT vypnout? Jak ověřit, že došlo k vypnutí?
- Je potřeba pro PT při paralelních požadavcích použít vždy unikátní uživatel nebo je možné použít jen několik uživatelů a testovaný systém lze zatěžovat stejným uživatelem najednou ve více vláknech?
- Jaké technologie je potřeba ještě nastudovat? Je možné využít interního školení?
- Jsou potřeba nějaké certifikáty či při přihlášení projít v PT nějakým autorizačním kolečkem? Jakou má expiraci access token a lze jeho platnost prodloužit na celou dobu běhu PT?
- Je potřeba zohlednit licence komponent, které bude na přímo či nepřímo PT zatěžovat?



- Jak vypadají klidové stavy testovacího prostředí, testované aplikace, databází, load balanceru atp. Jak lze před PT klidové prostředí ověřovat? (CPU, RAM, HEAP, TCP Connections, I/O atp.)
- Je vyžadována součinnost s podporou, bude probíhat oznamování o test exekuci PT, půjde o dohledový PT či bez dohledu a při test exekuci PT bude současně přítomno fyzicky / vzdáleně více lidí pro kooperaci s investigací či hledání zdroje incidentu? A s tím souvisí, zdali bude vyžadována notifikaci při selhání prostředí a při ukončení PT?
- Jaké jsou komunikační kanály pro kooperaci či podporu? Email: jaké skupiny, Slack: jaký channel atp. popř. telefonní čísla.
- Neprobíhá na testovacím prostředí nějaká dávková operace či nejsou spuštěny joby u kterých je nutné počkat na doběhnutí, aby testovací prostředí bylo stabilní a nebylo skrytě vytěžováno? Například přelejváním databází.
- Jaký typ PT se bude vytvářet? Performance Test, Stress Test, Soak Test, Failover Test atp.
- Jaké jsou pro WPT definovány user journey?

Architektura (při přípravě PTA)

- Je k dispozici blokový diagram architektury, business scénářů či flow volání služeb?
- Jaké prvky software / hardware jsou v cestě k testované aplikaci a jaká je jejich konfigurace / jak fungují?
- Existuje možnost zjistit aktuální stav prostředí a testované aplikace a jak bude probíhat notifikace o plánované změně konfigurace HW či plánované změně chování testované aplikace?



Konfigurace HW / SW (při přípravě PTA)

- Jaké jsou zdroje aplikacního serveru a jaké je jeho nastavení, podů v případě Kubernetes? Jaké je nastavení GC? To stejné platí o loadbalancerech, databázích, služeb atp.
- Je k dispozici benchmark pro síťové prvky, paketová analýza, I/O operace atp.



Server	Testovací prostředí	Produkční prostředí
Aplikační server	Popis prostředí IP / URL CPU RAM HEAP / GC OS Další software a nastavení Certifikáty	Popis prostředí IP / URL CPU RAM HEAP / GC OS Další software a nastavení Certifikáty
Databáze	Popis prostředí IP / URL CPU RAM HEAP / GC OS Další software a nastavení Certifikáty	Popis prostředí IP / URL CPU RAM HEAP / GC OS Další software a nastavení Certifikáty
Služby	Popis prostředí IP / URL CPU RAM HEAP / GC OS Další software a nastavení Certifikáty	Popis prostředí IP / URL CPU RAM HEAP / GC OS Další software a nastavení Certifikáty



Odhady (při přípravě PTA)

- Prvotní odhady se mohou připravovat již po úvodním sběru informací pro přípravu PTA a zároveň je potřeba počítat s možnými změnami v závislosti na odkrývání náročnosti přípravy PT.
- Složitost odhadů se může lišit podle dostupnosti informací.
- Při přípravě odhadů můžeme vycházet z již předem získaných znalostí testované aplikace a prostředí firmy.

Oblast	MD
Analýza rizik PT	
Příprava testovacích dat	
Příprava a vývoj PT	
Příprava a vývoj WPT	
Příprava smoke testu	
Příprava běhového prostředí pro automatizované testy	
Školení	
Příprava flow pro obsluhu doprovodných nástrojů	
Test Analýza	
Test Exekuce	
Prvotní investigace a prvotní reporting	
Incident / defekt	
Hlubší investigace, příprava reportu a reporting	
Retest	
Celkem	



Analýza rizik při přípravě PT (při přípravě PTA)

- V rámci exploratory testované aplikace je potřeba se zaměřit na její testovatelnost, složitost při vytváření požadavků, mechanismus dynamicky dosazovaných hodnot, funkcionálních incidentů, stability, stavu dokumentace, schopnost spolupráce PT dotčených týmů / garantů.
- Výsledkem může být fakt, že PT nemá smysl provádět, nebo že není možné je vytvořit podle zadání v plném rozsahu, ale pouze v omezené podobě - například pouze PT komponentové úrovně.
- Stejně tak může být výsledkem informace, že výsledky PT vzhledem k testovacímu prostředí mohou být zavádějící a proto je potřeba změnit strategii a zaměřit se pouze na metriky TTFB při 1VU při zamockovaných službách.

Kontext	Detailní popis	Ticket / Kontaktovaná osoba / či progress řešení	Výsledek

Doporučení (při přípravě PTA)

- Pro PTA vytvořte samostatnou stránku v Confluence pod projektem, kam PT spadá, nebo pod stránku s PT.
- Do PTA zahrňte odkazy webů, měření, studijních materiálů, přílohy s grafy a prezentacemi, odkazy na Git repozitáře, linky na test reporty PT komponentové úrovně, pokud jde o E2E PT.
- Jakékoli nejasnosti ved'te v komentářích pod PTA a patřičné lidi assignujte. Pokud o PT proběhne důležitá komunikace mimo komentáře v PTA, například v emailu, slacku atp., překopírujte je také do komentářů.
- Do PTA stejně tak patří nezdary (neumíme vytvořit dataServer, neumíme najít chybu v testované aplikaci, neumíme interpretovat grafy zdrojů z databáze atp.) a upozornění na



nespolupráci = to jsou také rizika, které správné provedení PT také ovlivní a musí se o nich vědět.

- Při přípravě PTA neberte od garantů jednotlivých oblastí odpovědi typu "jo, to dodáme" nebo "to bude potřeba připravit" ale chtějte datum a jméno osoby, která pro PTA dodá informace. Po úvodním sběru vstupů pro PTA zašlete hromadný email na zúčastněné osoby s body, které jsou hotovy a kdo co slíbil a do kdy. Pokud sběr dodatečných informací trvá delší dobu, průběžně pošlete hromadný email se stavem PTA a kdo co slíbil a stále ještě nedodal.
- Pro správné nastavení modelu zátěže nelze použít již existující logy - informace v nich mohou být zavádějící, neúplné a mohou obsahovat chyby. V logu zároveň nenajdeme business požadavky, není v nich k dispozici ani použitelné assertace = chtějte vždy znát produkční profil = to, jak se uživatelé skutečně chovají + předpoklad, jak se budou chovat.

Takto detailní PTA se Vám bude hodit při objasňování nastavení PT, důvodů a způsobu test exekuce a při znovu oživení po delší době.



Radim Daniel Pánek
CEO, SDET & Performance Tester

rdpanek@canarytrace.com
+420 731 011 200

<https://www.performance-testing.cz/>
<https://canarytrace.com>

Performance Testy neprovozujte bez měření rychlosti načítání webových aplikací a bez monitoringu chování prohlížeče v době, kdy je infrastruktura pod zátěží. Využijte bezúdržbový Canarytrace stack pro monitoring a měření webových aplikací z pohledu uživatele.