

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Hodnocení flexibility provozu aplikací v prostředí cloud computingu

Miroslav Brzobohatý

Vedoucí: Ing. Pavel Nápava Ph.D.

Studijní program: Softwarové inženýrství a technologie

Květen 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Brzobohatý** Jméno: **Miroslav** Osobní číslo: **465947**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Hodnocení flexibility provozu aplikací v prostředí cloud computingu

Název bakalářské práce anglicky:

Flexibility evaluation of applications running in a cloud computing environment

Pokyny pro vypracování:

Seznamte se s problematikou nasazování aplikací v prostředí Cloud Computingu a metodou hodnocení flexibility, založenou na výpočtu hodnoty Reálné opce. Využijte princip generování binomického stromu, který tato metoda využívá, navrhnete, implementujte a pomocí uživatelských testů vyhodnoťte výpočet flexibility, založený na generování a simulaci náhodných průchodů stromem. Porovnejte získané hodnoty s hodnotami, vypočtenými pomocí metody Reálných opcí. V rámci možností implementujte a ověřte využití trinomického stromu místo binomického.

Seznam doporučené literatury:

- [1] MASTNÝ, Martin. Real Option Analysis and cloud computing investments (draft). Praha, 2017. České vysoké učení technické v Praze.
- [2] TRNKA, Václav. Nástroj pro podporu výpočtu hodnoty Reálných opcí. Praha, 2017. České vysoké učení technické v Praze, Fakulta elektrotechnická.
- [3] NÁPLAVA, Pavel. REÁLNÉ OPCE A JEJICH VYUŽITÍ V PROSTŘEDÍ IT TECHNOLOGIÍ. Disertační práce. Praha, 2017. České vysoké učení technické v Praze, Fakulta elektrotechnická.
- [4] SCHOLLEOVÁ, Hana. Hodnota flexibility: Reálné opce. Praha: C. H. Beck, 2007. ISBN 978-80-7179-735-7.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Ph.D., katedra ekonomiky, manažerství a humanitních věd FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **05.02.2019**

Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Pavel Náplava, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

_____ Datum převzetí zadání

_____ Podpis studenta

Poděkování

Tímto bych rád poděkoval vedoucímu práce, Ing. Pavlu Náplavovi Ph.D., za to, že se mi při vypracování bakalářské práce aktivně věnoval a pomáhal mi. Také bych chtěl poděkovat rodině a přátelům, kteří mi byli oporou po celou dobu studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s metodickým pokynem o etické přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mé práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen "Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze, 24. května 2019

Abstrakt

Cílem této práce bylo zjistit využitelnost simulací při výpočtu flexibility cloudu. Jejím základem je teorie popisující rozdíly mezi on-premise a cloudem v IT infrastruktuře. Z ní pak vychází praktická část, která zahrnuje průběh návrhu, implementace a testování podpůrného experimentálního nástroje, který je výstupem této práce. Nástroj podporuje výpočty založené na binomickém modelu spojené se simulacemi včetně samotné metody reálných opcí.

Klíčová slova: Reálné opce, Nástroj, IT investice, Cloud, On-premise, Java, Cloud computing, Binomický model, Trinomický model, Flexibilita

Vedoucí: Ing. Pavel Nápava Ph.D.

Abstract

The aim of this work was to find out the usability of simulations in the calculation of cloud flexibility. It is based on a theory describing the differences between on-premise and cloud in IT infrastructure. It is based on the practical part, which includes the design, implementation and testing of the supporting experimental tool, which is the output of this work. The tool supports calculations based on a binomial model associated with simulations, including the real option method itself.

Keywords: Real option, Tool, IT investments, Cloud, On-premise, Java, Cloud computing, Binomial model, Trinomic model, Flexibility

Title translation: Flexibility evaluation of applications running in a cloud computing environment

Obsah

1 Úvod	1	6.3.5 Generování reportů	25
2 IT infrastruktura a Cloud Computing	3	6.4 Použitá technologie	26
2.1 Definice Cloud Computingu	3	7 Implementace	29
2.2 Modely služeb	4	7.1 Průběh implementace	29
2.2.1 IaaS, Infrastructure as a Service	5	7.1.1 Sestavení stromu	30
2.2.2 PaaS, Platform as a Service . .	5	7.1.2 Výpočet hodnoty IT investice	30
2.2.3 SaaS, Software as a Service . . .	5	7.1.3 Výpočet hodnoty opce	30
2.3 Výhody a nevýhody Cloud Computingu	6	7.1.4 Simulace	30
2.3.1 Výhody Cloud Computingu . .	6	7.1.5 GUI	31
2.3.2 Nevýhody Cloud Computingu	7	7.1.6 Generování reportů	31
2.4 Shrnutí	7	7.2 Zhodnocení	32
3 Rešerše existujících způsobů hodnocení flexibility Cloudu pomocí statistických metod	9	8 Testování	33
3.1 Způsob provedení rešerše	9	8.1 Experimentální testování	33
3.1.1 Vyhledávání	9	8.1.1 Různá volatilita	34
3.2 Vyhodnocení provedené rešerše .	10	8.2 Různá délka životnosti a počet období	35
4 Metody hodnocení flexibility Cloud Computingu	11	8.3 Různé investice do HW	36
4.1 Flexibilita	11	8.4 Různé investice do HW a různé poplatky do Cloudu	37
4.2 Metody zohledňující flexibilitu . .	11	9 Aktuální stav a možnosti dalšího vývoje	39
4.2.1 Metoda reálných opcí	11	9.1 Aktuální stav	39
4.2.2 Metoda Monte Carlo	12	10 Závěr	41
4.3 Zhodnocení	12	Literatura	43
5 Simulace průchodů binomickým stromem	13	A Seznam použitých zkratk	45
5.1 Binomický model při výpočtu reálné opce	13	B Obsah příloženého CD	47
5.2 Simulace průchodů binomickým modelem	17		
5.3 Trinomický strom	17		
5.4 Závěr	19		
6 Analýza a návrh řešení	21		
6.1 Cíle práce	21		
6.2 Analýza požadavků	21		
6.2.1 Funkční požadavky	21		
6.2.2 Nefunkční požadavky	22		
6.3 Návrh	22		
6.3.1 Datová struktura	22		
6.3.2 Simulace	24		
6.3.3 Generování průchodů	25		
6.3.4 Grafické rozhraní	25		

Obrázky

2.1 Model služeb Cloud Computingu [1]	5
5.1 Vývoj ceny podkladového aktiva dle binomického modelu [autor] ...	14
5.2 Simulace optimistického a pesimistického průchodu stromem [autor]	18
5.3 Vztah binomického a trinomického modelu [autor]	19
6.1 Datová struktura [autor]	23
6.2 GUI pro metodu založenou na simulacích [autor]	26
6.3 GUI pro metodu reálných opcí [autor]	27
6.4 Ukázka reportu - počet cest vedoucích do aktuálního uzlu ze spodního předchůdce [autor]	27

Tabulky

8.1 Experiment s různou volatilitou	34
8.2 Experiment s různou délkou a počtem období	36
8.3 Experiment s různými investicemi	37
8.4 Experiment s různými poplatky	38



Kapitola 1

Úvod

V dnešní době, kdy se investuje hodně a do všeho, je hodnocení investic téma, které stále častěji zaznívá při rozhodování o investici. Hodnocení investic není jednoduchá věc. Když se ale podaří správně stanovit, jak bude průběh investice vypadat, tak to může člověku nebo firmě ušetřit spoustu financí a taky času.

Cílem práce je seznámit se s problematikou hodnocení investic a na základně těchto informací navrhnout, implementovat a také otestovat podpůrný nástroj, který by nabízel nový způsob, jak počítat hodnocení investic. Návrh čerpá z již existujících metod a je postaven na průchodech binomickým modelem a měla by být zodpovězena otázka využitelnosti trinomického stromu namísto binomického.

Kapitoly teoretické části jsou věnované srovnání cloudu a on-premise a dále přiblížení si základních pojmů a vzorců z ekonomie. V kapitolách je vysvětleno, jak funguje cloud a jaké možnosti nabízí oproti on-premise včetně srovnání těchto dvou možností, dále bude přiblížen binomický a trinomický model oceňování opcí, bude představena metoda reálných opcí a v neposlední řadě simulace průchodů binomickým stromem. Kapitoly praktické části se pak už věnují samotnému návrhu aplikace. První část bude tvořit analýza a návrh řešení. Na základě získaných výsledků je posléze navržen podpůrný nástroj. V dalších kapitolách se pak věnují průběhu implementace nástroje. Na to následuje kapitola věnující se testování nástroje. Na závěr je kapitola věnující se vyhodnocení, kde se podíváme na naplnění cílů práce a možnosti případného dalšího zkoumání.

Motivací pro mě bylo, že se můžu seznámit s něčím novým a můžu posunout někam dál to, co začali mí předchůdci.

Kapitola 2

IT infrastruktura a Cloud Computing

Cloud Computing není nová věc. Existuje už několik desetiletí. Poprvé se myšlenky o dostupnosti výpočetního výkonu odkudkoliv začaly objevovat v 60. letech minulého století. Více se tato myšlenka začala šířit a zpracovávat až v 90. letech, kdy se poprvé objevuje pojem *Cloud computing*. Přesněji v roce 1997 ho použil Ramnath Chellappa [2] na své přednášce.

2.1 Definice Cloud Computingu

Přeneseně můžeme říci, že se v určitém slova smyslu jedná o službu, ke které má přístup každý počítač, který má přístup k síti. Služba v tomto případě představuje určitou vrstvu, ve které se skrývá software, hardware, datová uložení, databáze a aplikace. Zákazník, tedy člověk nebo organizace, která bude tuto službu využívat, si podle vlastního uvážení a potřeb pronajme určitý kus samotného hardwaru, nebo případně kompletní balík pro správu infrastruktury.

Za první obecnou definici Cloud Computingu, lze považovat definici poradenské společnosti Gartner, Inc. “*Cloud Computing is a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external customers using Internet technologies*” [3]. Ta ve stručnosti říká, že Cloud Computing je způsob sdílení výpočetní kapacity zákazníkům formou služby napříč sítí. Obdobně, jako společnost Gartner, se pokoušely i ostatní společnosti postavit vlastní definici Cloud Computingu. Důkazem, že sestavit jednoznačnou definici se ukázalo být těžším úkolem, než se na první pohled zdálo, je existence desítky dalších definic, které se v mnohém liší, ale všechny mají společných několik základních principů, zbytek definice se pokaždé liší podle potřeb a úhlu pohledu autorů definice. Místo jednoznačné definice, na které se společnosti nebyly schopny domluvit, se tak začaly stanovovat společné atributy pro všechny definice, které charakterizují Cloud Computing a zároveň ho jednoznačně odlišují od ostatních technologií.

Jeden z prvních, kdo tyto atributy stanovil, byla opět společnost Gartner, Inc. Samozřejmě i definice jednotlivých atributů charakterizujících Cloud Computing se potkaly s protichůdnými názory, ale i navzdory těmto názorům se tyto definice považují za obecné výchozí vlastnosti *Cloud Computingu*.

Nakonec bylo definováno pět základních atributů [4]:

1. Princip služby (Service-Based)

Zákazník a poskytovatel jsou navzájem jednoznačně odděleni pomocí přesně definovaného rozhraní, které se nazývá službou. Tato služba staví technickou stránku realizace pro zákazníka do postavení černé skříňky (black box), která ukazuje směrem ven k zákazníkovi pouze rozhraní připravené přijímat požadavky, aniž by bylo nutné se starat o to, jak je nástroj realizovaný uvnitř. Samotný provoz a nastavení této "černé skříňky" je z pohledu zákazníka zcela automatické.

2. Škálovatelnost a elasticita (Scalable and Elastic)

Škálovatelnost dává zákazníkovi možnost upravovat automaticky nebo ručně požadavky na výkon a kapacitu, kterou aktuálně potřebuje a je schopen využít. Elasticita popisuje schopnost sdílených prostředků se přizpůsobovat požadavkům zákazníka, a to jak směrem dolů, tak i směrem nahoru. Dalo by se říci, že se jedná o strukturu, která se automaticky přizpůsobuje podle toho, jak ji zákazník využívá, do ideálního vyrovnaného stavu.

3. Sdílení (Shared)

Základním kamenem celé služby je velmi robustní IT infrastruktura postavená za účelem sdílení a uspokojení potřeb co nejvíce zákazníků. Infrastruktura je sdílená mezi velkým počtem různých zákazníků. S nevyužitými zdroji lze libovolně manipulovat a docílit tak co největší optimalizace výkonu a využití dostupných zdrojů na maximum.

4. Placení za využití (Metered by Use)

Poskytovatel služeb sleduje a měří využití zdrojů zákazníkem a podle definovaných modelů pak vypočítává měsíční sazbu. Vždycky platí, že zákazník platí jen za to, co opravdu využil.

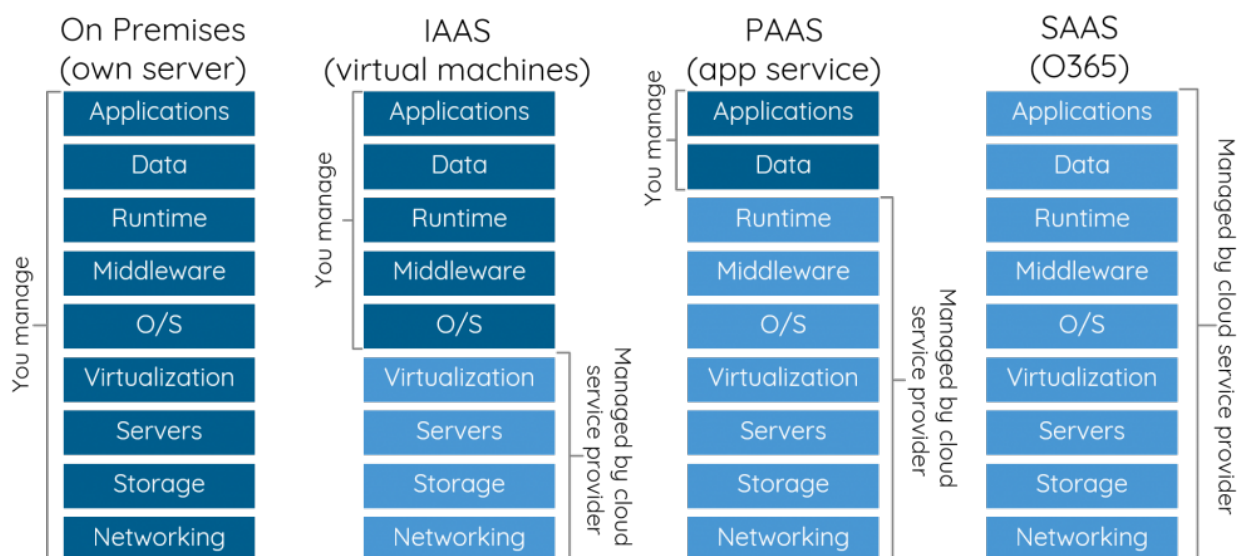
5. Používání prostřednictvím internetu (Uses Internet Technologies)

Ze samotného názvu atributu vyplývá, že jde o to, jak se zákazník ke službě dostane. Zákazník se ke službě dostane prostřednictvím internetu a síťových technologií, jako jsou například protokoly HTTP a HTTPS.

2.2 Modely služeb

Na obrázku 2.1 můžeme vidět *Model služeb*, který uvádí rozsah infrastruktury, kterou v rámci *Cloud Computingu* poskytovatelé nabízí zákazníkům jako službu. Výběr máme ze tří možných modelů, které jsou zobrazeny v porovnání s tradičním modelem infrastruktury (on-premise) na obrázku 2.1.

Tmavě modře zbarvené části infrastruktury jsou ty, které si zákazník spravuje ve svém *on-premise* prostředí, na druhé straně barvou světle modrou jsou zbarveny ty části, které si zákazník pronajímá v prostředí *Cloud Computingu*. V případě, že má zákazník všechny části infrastruktury u sebe, tedy provozuje



Obrázek 2.1: Model služeb Cloud Computingu [1]

on-premise řešení, tak si musí všechny potřebné komponenty zakoupit. V opačném případě, když využívá službu SaaS, tak nemusí kupovat vůbec nic, protože si všechno pronajímá. Typické věci pro jednotlivé služby jsou stručně popsány v následujících podkapitolách.

2.2.1 IaaS, Infrastructure as a Service

Infrastruktura jako služba je model, který většinou využívají firmy, které se nechtějí zabývat správou hardwaru, ale zároveň si samy zajistí software a jiné potřebné komponenty. Jednoduše řečeno si zákazník tímto modelem pronajímá hardware. Nad samotným hardwarem nemá žádnou kontrolu a prakticky vzato nemusí o něm vědět vůbec nic. Kdykoliv samozřejmě může navyšovat či ponížovat požadovaný výkon.

2.2.2 PaaS, Platform as a Service

Platforma jako služba je model, který se využívá převážně pro vývoj aplikací a jejich následné testování. Zákazník získá přístup k připravenému prostředí, ve kterém může začít vyvíjet a nemusí se zabývat jinými věcmi. Výhodou je výrazné snížení složitosti samotného vývoje aplikace. Tato možnost nabízí škálovatelný výpočetní výkon, který je potřebný pro vývoj jako takový.

2.2.3 SaaS, Software as a Service

Software jako služba je model, který představuje úplný opak *on-premise* řešení. Zákazník si pronajímá úplně všechno od hardwaru až po aplikace. Zákazník nemá kontrolu jak nad hardwarem, tak ani nad systémem s aplikacemi. Předpokládáný je provoz ve webovém prostředí.

2.3 Výhody a nevýhody Cloud Computingu

Tak jako všechno, tak i Cloud Computing má své výhody, pro které ho lidé využívají, a své nevýhody, pro které ho kritizují. V následujících dvou podkapitolách si ve stručnosti představíme ty nejznámější z nich.

2.3.1 Výhody Cloud Computingu

1. Žádné či malé investice

Jelikož celá infrastruktura se nachází u poskytovatele, tak není potřeba investovat do vlastní infrastruktury. Maximálně se zde mohou nacházet výdaje na různé licence.

2. Žádné opakované investice do infrastruktury

V budoucnosti nehrozí opakované investice do nových hardwarových komponent, které se většinou každých 3 až 5 let obnovují v závislosti na podpoře výrobce. U Cloud Computingu se tato starost přesouvá na poskytovatele služby. Samostný zákazník se o stav infrastruktury nestará, pro něj se tváří stále ve stejném stavu.

3. Žádné skryté výdaje

Nehrozí, tak jako u *on-premise* infrastruktury, že se náhodně rozbije disk, že bude potřeba reinstalovat server a další jevy, které se náhodně objevují.

4. Není potřeba správa a údržba

Veškerou starost o stav, zabezpečení a hladký běh přebírá poskytovatel Cloud Computingu.

5. Nezávislost na vlastních zaměstnancích

S možností přemístit celou IT-infrastrukturu k někomu jinému souvisí nadbytečnost mít vlastní IT oddělení, které by se o infrastrukturu staralo.

6. Možnost dynamicky měnit kapacitu

Pravděpodobně největší předností Cloud Computingu je možnost dynamicky měnit velikost požadovaného výkonu, tak aby nezůstávala část výkonu nevyužitá. Jako ukázkou můžeme vzít obchody, které mají své webové stránky. Je jasné, že v létě bude navštěvovat web méně lidí než v prosinci o svátcích, a tudíž pouze v prosinci potřebuje vyšší výkon než ve zbylých měsících.

7. Rychlost nasazení

Rychlost nasazení oproti *on-premise* infrastruktuře, kde se musí pořídit všechny komponenty a následně se musí sestavit, je prakticky okamžitá, protože v pronajmuté infrastruktuře je už vše nastavené a může se rovnou začít pracovat.

2.3.2 Nevýhody Cloud Computingu

1. Někdy vyšší ceny než za realizaci svépomocí

Může se stát, že cena za pronajmutou infrastrukturu může být vyšší než pořízení a spravování vlastní infrastruktury, a to hlavně z dlouhodobějšího hlediska u specifických modelů Cloud Computingu, kde jsou přírážky za určité benefity, jako je např. velmi vysoká dostupnost služby.

2. Zákazníková data jsou uschována na cizí infrastruktuře

I přesto, že data jsou šifrována a na serverech poskytovatele jsou pravděpodobně zákazníková data lépe zabezpečena, než by byla na vlastních serverech, tak se stále nachází na cizí infrastruktuře, kterou spravuje někdo jiný.

3. Zákazníková data putují internetem

Nesmíme zapomenout, že data cestují napříč internetem, vícekrát než by bylo nezbytně nutné a může se stát, že při větším útoku, i přesto, že jsou zašifrována, se k nim může někdo dostat.

4. Volba hardware a software je limitována poskytovateli Cloud Computingu

Není možnost měnit základní konfiguraci a nastavení hardwaru. To znamená, že zákazník nemůže ovlivnit to, jaké aplikace či jaké nastavení komponent by chtěl, ale musí se přizpůsobit nabídce poskytovatele.

5. Někdy pomalejší reakční doba

Jelikož poskytoval Cloud Computingu se může nacházet prakticky kdekoliv na světě a tím pádem se kdekoliv na světě nachází i hardware a software, který nabízí. V důsledku může docházet místy k výpadkům či prodlevám. Je rozdíl, když se připojujete k serveru v České republice z České republiky nebo z USA.

2.4 Shrnutí

V této kapitole jsme se dozvěděli, co je vlastně *Cloud Computing*, k čemu se používá, jaké poskytuje výhody a nevýhody v oblasti vytváření IT infrastruktury. Tou hlavní výhodou je schopnost se flexibilně přizpůsobovat aktuálním požadavkům a minimalizovat tím tak nadbytečné výdaje. Dnes, když se vytváří IT infrastruktura, tak se příliš neřeší, jak jí udělat efektivně. Mnozí nejspíš ani nevědí, že je i jiná možnost než *on-premise* IT infrastruktura, a v případě, že vědí, že existuje možnost *Cloud Computingu*, tak nejsou schopní porovnat, jaká varianta je pro ně výhodnější a vybírají na základě jednoduchých nástrojů, které nezohledňují všechny faktory. Nejvíce opomíjeným faktorem je právě flexibilita cloudu. Na tuto problematiku už vznikají práce [5] [6] [7] [8], které se zabývají právě tím, jak zohlednit při výpočtu faktor flexibility. V této práci se dále budeme zabývat hledáním a návrhem

alternativní metody výpočtu založené na statistických metodách. Podklady a veškeré informace jsem hledal v databázích závěrečných prací vysokých škol, případně v odborné literatuře v databázích IEEE Xplore a SpringerLink, a to pod klíčovými slovy *cloud*, *cloud computing v IT*, *ROA*, *binomial*, *flexibility*, *statistical method*.

Kapitola 3

Rešerše existujících způsobů hodnocení flexibility Cloudu pomocí statistických metod

V předchozí kapitole jsme si definovali, co je Cloud Computing a jaké má vlastnosti. V této kapitole se budeme zabývat hledáním již existujících implementací/způsobů hodnocení flexibility Cloud Computingu pomocí statistických metod. Výsledek hledání by nám měl zodpovědět následující otázky:

1. Jsou nějaké statistické metody, které se používají pro hodnocení flexibility Cloud Computingu?
2. Existuje nějaké propojení mezi metodou Monte Carlo a hodnocením flexibility Cloud Computingu?
3. Je možné vypočítat flexibilitu cloudu pouze pomocí statistické metody?

3.1 Způsob provedení rešerše

Pro vyhledání relevantních a důvěryhodných odpovědí byly využity tyto informační databáze: IEEE Xplore, SpringerLink.

Jako vyhovující byly označeny pouze dokumenty, jejichž plný text byl k dispozici online a byly psány v jazyce českém nebo anglickém. Důležitou roli hrál i rok vytvoření, aby byla zajištěna aktuálnost hledaných prací, tak jako vyhovující byly brány pouze ty, které vznikly v roce 2012 a později. Nalezené práce byly pročteny a ty, které se ukázaly jako použitelné pro tuto práci, byly přidány do seznamu literatury.

3.1.1 Vyhledávání

V rámci vyhledávání byla použita tato klíčová slova: “**statistical method**”, “**Monte Carlo**”, “**Cloud - Cloud computing**”, “**ROA**”, “**binomial**”, “**flexibility**”.

Pod klíčovými slovy “**Monte Carlo**”, “**ROA**” bylo nalezeno 9 prací, s tím, že ani jedna neodpovídala tématem.

Pod klíčovými slovy “**Monte Carlo**”, “**ROA**”, “**Cloud**” byla nalezena 1 práce, která také neodpovídala tématem.

Pod klíčovými slovy “**Monte Carlo**”, “**flexibility**”, “**Cloud**”, “**binomial**” bylo nalezeno 4 práce, z nichž 1 odpovídala tématem.

Pod klíčovými slovy “**Monte Carlo**”, “**flexibility**”, “**Cloud computing**”, “**ROA**” byly nalezeny 2 práce, z nichž 1 odpovídala tématem.

Celkem tedy bylo nalezeno 7 prací a z toho 2 se blížily tématu této práce.

3.2 Vyhodnocení provedené rešerše

Rešerše byla provedena s cílem získat obecný přehled o realizovaných výzkumech týkajících se *metod hodnocení flexibility cloudu pomocí statistických metod*. Zajímalo nás, zda se už někdo o podobné téma zajímal a zda už nějaká metoda existuje.

1. Jsou nějaké statistické metody, které se používají pro hodnocení flexibility Cloud Computingu? - **Ne**.

Existují metody, které používají v nějaké části výpočtu statistické metody, ale nebyla nalezena žádná čistě statistická metoda, která by počítala flexibilitu cloudu.

2. Existuje nějaké propojení mezi metodou Monte Carlo a hodnocením flexibility Cloud Computingu? - **Ano**.

Našli jsme práce, které se zabývají flexibilitou cloudu a metodou Monte Carlo. Metoda Monte Carlo je v těchto pracích používána pro odhad nebo výpočet některých veličin náročných na výpočet normálním způsobem, jedná se například o volatilitu.

3. Je možné vypočítat flexibilitu cloudu pouze pomocí statistické metody? - **Ne**.

Nenalezli jsme žádnou práci, která by se zajímala o výpočet flexibility cloudu pouze pomocí statistické metody. Nepovedlo se nám najít zdroj inspirace, jak k tomu přistoupit, takže jsme se rozhodli, že použijeme vlastní metodu, která je popsána dále 5.2.

Kapitola 4

Metody hodnocení flexibility Cloud Computingu

V této kapitole se seznámíme s pojmem *flexibilita*. Vysvětlíme si, co znamená a co si pod ním představit.

4.1 Flexibilita

Pod pojmem flexibilita si můžeme představit schopnost se přizpůsobovat změnám. V oblasti IT to znamená schopnost se přizpůsobovat požadavkům na výpočetní výkon, paměť a dalším podobným parametrům, na které se zvyšují či snižují požadavky v rámci IT infrastruktury v průběhu fungování společnosti. Společnosti, která rychle reaguje na probíhající změny, říkáme, že je *“agilní”*. Pojmy *agilita* a *flexibilita* se dnes často zaměňují. Ten, kdo dnes je schopný rychle a vhodně reagovat na změny, které neustále probíhají, má značnou výhodu oproti ostatním.

“V souvislosti se strategickým investováním můžeme flexibilitu definovat jako schopnost změnit svá rozhodnutí a v reálném čase investici rozšířit, ukončit, změnit vstupy či výstupy, přemístit apod., a to s cílem maximalizovat svůj užitek z ní.” [7]

4.2 Metody zohledňující flexibilitu

4.2.1 Metoda reálných opcí

Ve stručnosti zmíním a přiblížím metodu reálných opcí navrženou v disertační práci Pavla Náplavy [5] a diplomové práci Jiřího Trnky [6], který ji ve své práci implementoval. Metoda se zabývá výpočtem flexibility Cloudu pomocí binomického modelu v kombinaci s prodejní a nákupní opcí, která představuje povýšení či ponížení požadovaného výkonu infrastruktury.

Flexibilita cloudu je zde reprezentována jako binomický strom, který s využitím parametru volatility reprezentuje budoucí vývoj požadavků na změnu systému. Růst požadavků představuje posun z uzlu, ve kterém jsme směrem nahoru, a v případě poklesu požadavků směrem dolů.

Kapitola 5

Simulace průchodů binomickým stromem

V první řadě si představíme binomický model při výpočtu reálné opce, ukážeme si, jak celý postup vypadá a jak probíhá samotný výpočet opce. V druhé části této kapitoly si ukážeme, jakým způsobem zkombinujeme binomický strom a simulaci různých průchodů tímto stromem. A nakonec si povíme něco o trinomickém stromě, který jsme původně zvažovali také zaimplementovat, ale z časových důvodů jsme to nakonec neudělali.

5.1 Binomický model při výpočtu reálné opce

V této kapitole si definujeme binomické model pro výpočet hodnoty opce, který je pro práci jakýmsi základem, na kterém bude postaven celý výpočet. Představíme si zde všechny důležité vzorce, které budou potřeba. Binomický model je jedním z nejjednodušších modelů oceňování opcí. Model poskytuje poměrně přesné výsledky s využitím jednodušších postupů než jiné přesnější modely. Pro popis binomického modelu byly použity práce [6] [9] [7] [10] [8]. Pro použití binomického modelu je nutné splnit předpoklady pro jeho použití. Předpoklady pro použití jsou následující:

1. Nelze dosáhnout bezrizikového zisku
2. Platnost zákona jedné ceny - Aktiva, která mají v budoucnu stejnou cenu, ji musí mít i dnes
3. Existence dokonalých trhů - Trhy, kde neexistují transakční náklady, daně či omezení na krátkodobý prodej
4. Výnosy aktiv jsou rovny bezrizikové sazbě

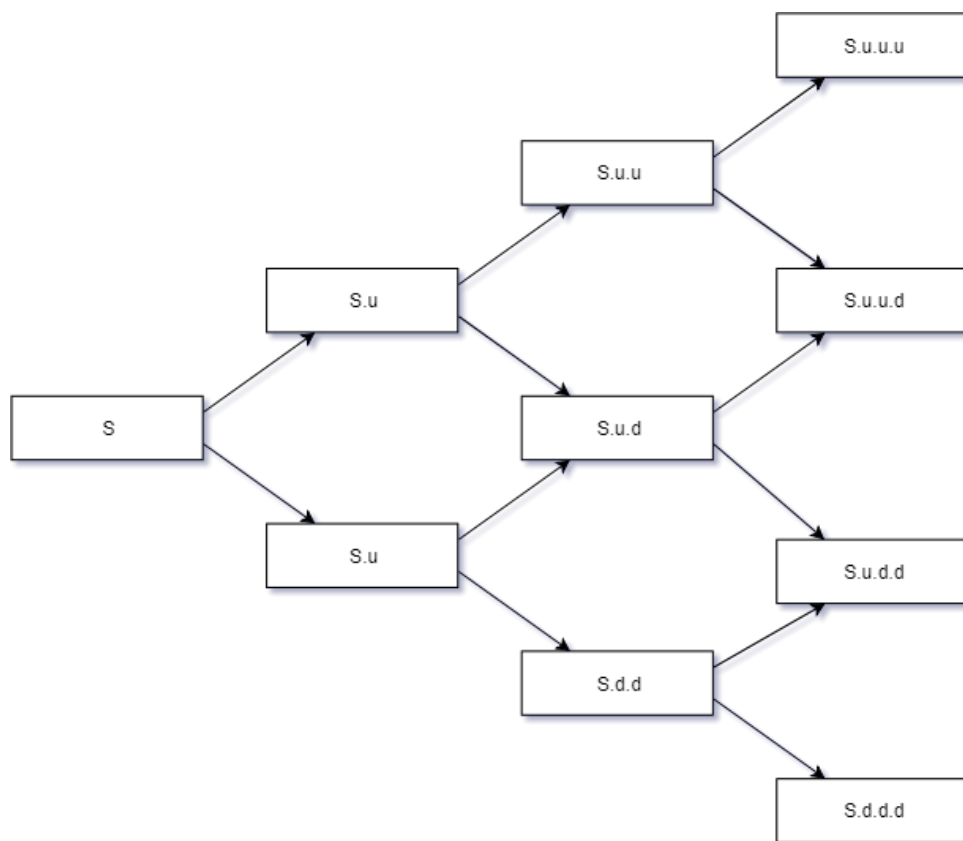
Podle typu dělíme opce na dva typy [6]:

- *Kupní* - Kupující má právo koupit podkladové aktivum za stanovenou cenu
- *Prodejní* - Prodávající má právo prodat podkladové aktivum za dohodnutou cenu

Podle doby dělíme opce na dva typy [6]:

- *Evropská* - opce může být uplatněna pouze v den expirace
- *Americká* - opce může být uplatněna kdykoliv od upsání opce až do doby expirace

Binomický model je postaven na tom, že cena podkladového aktiva S v čase T_i nabývá dvou různých hodnot, a to buď S_1^u , pokud cena aktiva vzrostla, nebo S_1^d , pokud cena klesla. Pravděpodobnosti růstu a poklesu se přitom v čase nemění. Vývoj spotové ceny podkladového aktiva dle modelu ilustruje obrázek 5.1.



Obrázek 5.1: Vývoj ceny podkladového aktiva dle binomického modelu [autor]

V uvedených vzorcích jsou použity následující parametry:

- u : Index růstu
- d : Index poklesu
- e : Základ přirozeného logaritmu

- S : Spotová cena
- X : Realizační cena
- r : Bezriziková úroková míra
- T : Čas zbývající do expirace
- n : Počet období v čase zbývajícím do expirace
- q : Míra volatility vyjádřená směrodatnou odchylkou
- p : Pravděpodobnost růstu
- $(1 - p)$: Pravděpodobnost poklesu
- C_u : Vnitřní hodnota kupní opce (call) na konci 1. období pro případ růstu
- C_d : Vnitřní hodnota kupní opce na konci 1. období pro případ poklesu
- P_u : Vnitřní hodnota prodejní opce (put) na konci 1. období pro případ růstu
- P_d : Vnitřní hodnota prodejní opce na konci 1. období pro případ poklesu
- C : Hodnota evropské kupní opce
- P : Hodnota evropské prodejní opce
- $(C_i)_n$: I-tá vnitřní hodnota americké kupní opce v období n
- $(P_i)_n$: I-tá vnitřní hodnota americké prodejní opce v období n

Aby bylo možné vývoj spočítat, je potřeba nejdříve znát hodnotu u a d .
Pro ty platí:

$$u = e^{q\sqrt{\frac{T}{n}}} \quad (5.1)$$

$$d = e^{-q\sqrt{\frac{T}{n}}} \quad (5.2)$$

Jelikož pokles a následný růst musí skončit původní hodnotou, musí platit také:

$$u \cdot d = e^{q\sqrt{\frac{T}{n}}} \cdot e^{-q\sqrt{\frac{T}{n}}} = e^0 = 1 \quad (5.3)$$

Pro výpočet hodnoty opce je dále nutné znát pravděpodobnost růstu p a pravděpodobnost poklesu $(1 - p)$:

$$p = \frac{(1+r)^{\frac{T}{n}} - d}{u - d} \quad (5.4)$$

$$1 - p = \frac{(1+r)^{\frac{T}{n}} - u}{d - u} \quad (5.5)$$

Pro které platí, že:

$$p + (1 - p) = 1 \quad (5.6)$$

Pro výhodné uplatnění opce musí být její vnitřní hodnota větší než nula. V opačném případě je její hodnota nulová, protože opce nebude uplatněna. Z tohoto důvodu se ve vzorcích využívá funkce $\max(x, 0)$, která vrací 0, pokud by bylo x záporné.

Pro vnitřní hodnotu kupní opce na konci prvního období platí:

$$C_u = \max((u \cdot S) - X, 0) \quad (5.7)$$

$$C_d = \max((d \cdot S) - X, 0) \quad (5.8)$$

Pro výpočet vnitřní hodnoty prodejní opce na konci prvního období je třeba prohodit S a X :

$$P_u = \max(X - S \cdot u, 0) \quad (5.9)$$

$$P_d = \max(X - S \cdot d, 0) \quad (5.10)$$

Pro hodnotu evropské kupní opce pro n období pak platí:

$$C = \frac{1}{(1+r)^n} \cdot \sum_{i=0}^n \frac{n!}{i! \cdot (n-i)!} \cdot p^i \cdot (1-p)^{n-i} \cdot \max(S \cdot u^i \cdot d^{n-i} - X, 0) \quad (5.11)$$

Analogicky pro hodnotu evropské prodejní opce pro n období platí:

$$C = \frac{1}{(1+r)^n} \cdot \sum_{i=0}^n \frac{n!}{i! \cdot (n-i)!} \cdot p^i \cdot (1-p)^{n-i} \cdot \max(X - S \cdot u^i \cdot d^{n-i}, 0) \quad (5.12)$$

Pro výpočet hodnoty americké opce je potřeba procházet rekurentně binomickým stromem zpět, kde i -tá hodnota (pro $i = 1 \dots n$) uzlu v období n je rovna vnitřní hodnotě kupní opce:

$$(C_i)_n = \max(S \cdot u^i \cdot d^{n-i} - X, 0) \quad (5.13)$$

a i -tá hodnota uzlu v periodě $n - 1$ se pak pro americkou opci počítá dle vzorce:

$$(C_i)_{n-1} = \max \left[\frac{1}{(1+r)^{\frac{T}{n}}} \cdot (p \cdot (C_i)_n + (1-p) \cdot (C_{i+1})_n), \max((S_i)_{n-1} - X, 0) \right] \quad (5.14)$$

Stejný princip platí i pro vnitřní hodnotu americké prodejní opce:

$$(P_i)_n = \max(X - S \cdot u^i \cdot d^{n-i} - X, 0) \quad (5.15)$$

a i -tá hodnota uzlu v periodě $n - 1$ se pak pro americkou prodejní opci počítá dle vzorce:

$$(P_i)_{n-1} = \max \left[\frac{1}{(1+r)^{\frac{T}{n}}} \cdot (p \cdot (P_i)_n + (1-p) \cdot (P_{i+1})_n), \max(X - (S_i)_{n-1}, 0) \right] \quad (5.16)$$

5.2 Simulace průchodů binomickým modelem

Z metody reálných opcí použijeme tvorbu binomického modelu, hodnota flexibility v podobě celkových ušetřených nákladů pak bude získána právě pomocí simulací různých scénářů nad vytvořeným binomickým stromem. Příklad binomického modelu se znázorněnými průchody zobrazuje obrázek 5.2.

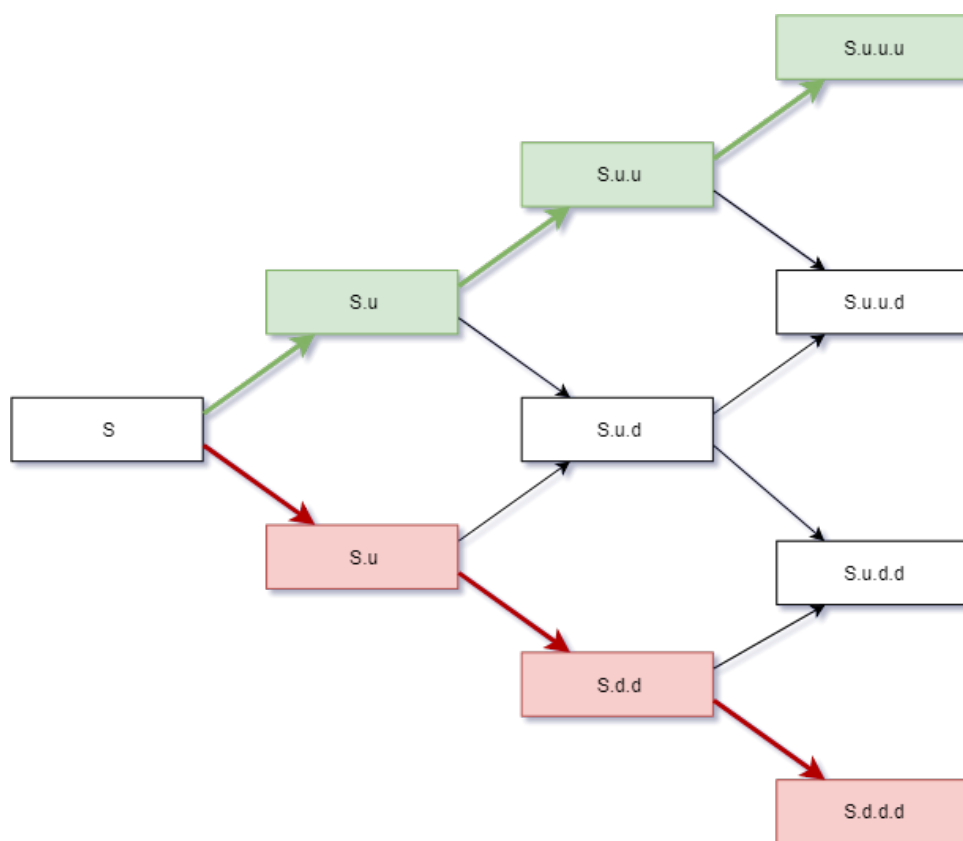
Možnou modifikací je použití trinomického modelu namísto modelu binomického a nad ním provádět simulace. Pro ověření správnosti výpočtu použijeme scénáře, které vychází z již existujících prací a na kterých tedy lze ověřit správnost samotného výpočtu.

Zelenou barvou je vyobrazen optimistický průchod a naopak červenou barvou je vyobrazen pesimistický průchod. Tyto příklady představují jednu z mnoha možných simulací v závislosti na velikosti binomického stromu. Cílem je simulacemi průchodů binomickým stromem zahrnout do výpočtu variabilitu budoucích změn, protože nelze předem říci, které změny nastanou a které nenastanou. Tento způsob výpočtu bere v potaz celou množinu situací, které mohou nastat.

Menší nevýhodou je omezení metod vycházející z toho, že změna mezi jednotlivými úrovněmi binomického stromu je omezená, to vyplývá ze samotné metody reálných opcí.

5.3 Trinomický strom

V této podkapitole si přiblížíme pojem trinomický model oceňování opcí, který se nabízí jako alternativa k binomickému modelu. Oproti binomickému



Obrázek 5.2: Simulace optimistického a pesimistického průchodu stromem [autor]

modelu nabízí další scénář, který může nastat. V binomickém modelu mohou nastat scénáře - povýšení požadavků nebo ponížení, v trinomickém modelu může nastat ještě třetí scénář - požadavky zůstanou stejné, nezvýší se ani se nesníží. To by nám umožnilo generovat přesnější simulace a celkově by výsledek měl být přesnější, neboť zahrnujeme možnost nicnedělání. K popisu trinomického stromu byly použity tyto zdroje [9].

■ Binomický vs Trinomický model

Pokud se blíže podíváme na binomický a trinomický model oceňování opcí, tak si můžeme všimnout, že ze dvou kroků binomického modelu se stane jeden krok v modelu trinomickém. Vztah mezi binomickým a trinomickým modelem je znázorněn na obrázku 5.3.

Vnitřní hodnota evropské kupní (call) opce je obdobně jako u binomického modelu:

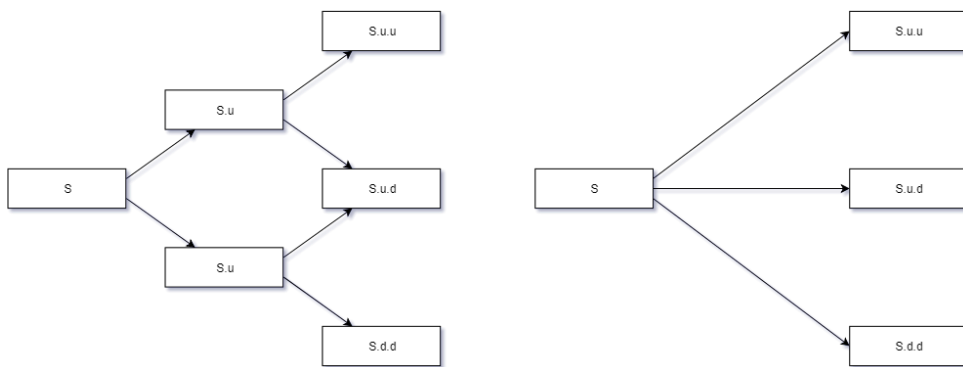
$$C = \max(0, S_t - RC) \quad (5.17)$$

přesněji

$$C_u = \max(0, uS - RC) \quad (5.18)$$

$$C_m = \max(0, S - RC) \quad (5.19)$$

$$C_d = \max(0, dS - RC) \quad (5.20)$$



Obrázek 5.3: Vztah binomického a trinomického modelu [autor]

5.4 Závěr

V teoretické části práce jsme se seznámili s problematikou hodnocení flexibility cloudu a s metodami, které se výpočtem hodnoty flexibility zabývají. Představili jsme si základní myšlenky binomického modelu a vzorce potřebné k vytvoření binomického modelu. Řekli jsme si výhody a nevýhody jednotlivých metod.

Představili jsme si simulace různých průchodů binomickým modelem, kterými se budeme dále zabývat v praktické části této práce.

Kapitola 6

Analýza a návrh řešení

V této kapitole se podíváme na samotný návrh podpůrného nástroje. V první řadě se podíváme na funkční požadavky definované zadáním. Po shrnutí požadavků si přiblížíme návrh implementovaného nástroje. Na to navazuje popis datové struktury a vyřešení grafického rozhraní nástroje. Předposlední kapitola se věnuje generování reportů a na samotný závěr kapitoly je popsána použitá technologie.

6.1 Cíle práce

Cílem této práce bylo navrhnout a implementovat podpůrný nástroj, který by pro výpočet hodnot IT investic využíval simulací a na jejich základě získával výslednou hodnotu flexibility cloudu 5.2. Jedná se primárně o experimentální aplikaci z toho důvodu, že nebylo předem jasné, jaké výsledky dostaneme a šlo hlavně o to zjistit, zda je možné pro výpočet hodnot IT investic využít simulace. Práce staví na základě již existujících prací [5] [6], z kterých využila základ, na kterém postavila novou metodu výpočtu hodnoty IT investice. Cílem bylo vyvinout funkční nástroj, který nabízí alternativní způsob výpočtu hodnoty IT investice vedle metody reálných opcí. Než přejdeme k samotnému návrhu, tak je potřeba si vyspecifikovat funkční požadavky.

6.2 Analýza požadavků

V této části se podíváme na požadavky, které byly kladeny na podpůrný nástroj, samotné požadavky můžeme rozdělit na funkční, požadavky na funkčnost aplikace, a nefunkční, které se týkají věcí jako je výkon a vzhled nástroje.

6.2.1 Funkční požadavky

Funkční požadavky byly specifikovány na samotném začátku a jedná se o tyto:

1. **Binomický model** - Hlavním funkčním nárokem, který byl po aplikaci vyžadován, je požadavek využít pro výpočet binomický model. Binomický

model má jasně definované vzorce, které jsou zmíněny v sekci 5.1 a na základě kterých se konstruuje binomický strom, který se pak využívá pro další výpočty.

2. **Výpočet hodnoty IT investice** - Mimo obecné reálné opce musí aplikace podporovat také výpočet rozvoje vnitřních hodnot a celkové hodnoty IT investice. Pro výpočet této metodiky je nutné použít opci s pamětí zmíněnou v pracích [5] [6].
3. **Generování reportů** - Jako důležitá část bylo nakonec přidáno i generování jednoduchých reportů.

6.2.2 Nefunkční požadavky

Funkční požadavky byly specifikovány na samotném začátku práce a jedná se o tyto:

1. **Desktopová aplikace** - Prioritním nefunkčním požadavkem je použitelnost nástroje, proto se má jednat o desktopovou aplikaci.
2. **Neomezující rychlost výpočtu** - Rychlost provedení výpočtu a odezvy by neměla být příliš vysoká, aby se nestala omezující podmínkou. Nejedná se však o prioritní požadavek.

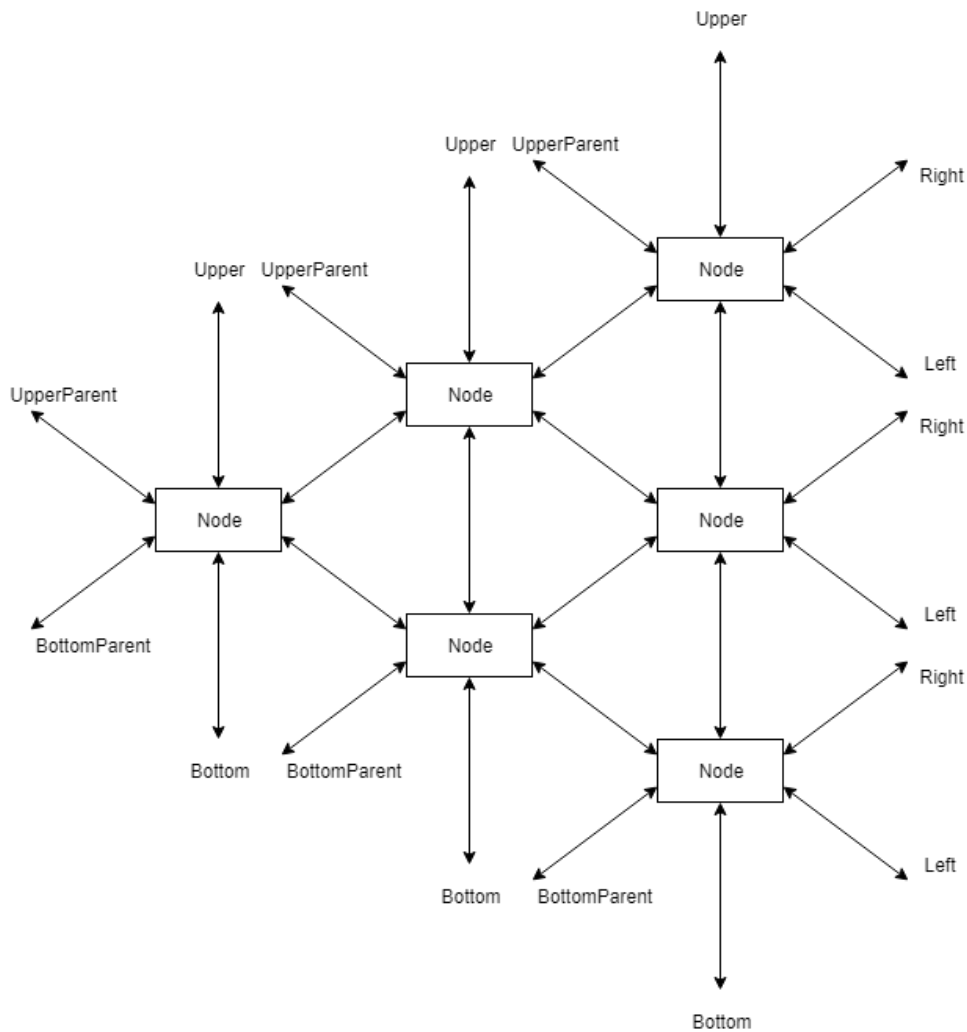
6.3 Návrh

Na základě získaných informací z prací [6] [11] byla navržena základní struktura aplikace tak, že nástroj bude podporovat výpočet založený na simulacích průchodů binomickým stromem a jako vedlejší možnost bude podporovat výpočet pomocí metody reálných opcí.

6.3.1 Datová struktura

V první řadě bylo důležité najít datovou strukturu, která by dokázala uchovávat parametry a důležité mezivýpočty pro správné fungování aplikace. U použité struktury bylo důležité, aby dokázala během výpočtu uchovávat data o průchodnosti uzlů a další informace potřebné pro generování stanovených reportů a zároveň aby byla použitelná pro implementaci vedlejší metody reálných opcí. Po krátké úvaze byla vybrána struktura, kterou navrhl a použil Václav Trnka ve své diplomové práci [6]. Jedná se o orientovaný graf, ve kterém si jednotlivé uzly uchovávají reference na všechny uzly v nejbližším okolí. Datová struktura je vidět na obrázku 6.1. Tato struktura je nejen vhodná ke klasickému výpočtu metody reálných opcí, ale dá se zároveň použít jako nosič informací, jako je průchodnost uzlů při simulacích, ke kterým jinak samotná datová struktura není příliš potřeba.

Každý uzel, jak je z obrázku vidět, má celkově šest referencí na okolní uzly.



Obrázek 6.1: Datová struktura [autor]

1. **UpperParent** slouží jako označení pro referenci na uzel, který je horním předchůdcem aktuálního uzlu. Reference je používána především pro schopnost přenášet seznam anuitních splátek.
2. **BottomParent** označuje referenci na uzel, který je spodním předchůdcem aktuálního uzlu. Slouží ke stejnému účelu jako reference na horního předchůdce.
3. **Right** označuje referenci na uzel, který je horním následníkem aktuálního uzlu.
4. **Left** označuje referenci na uzel, který je spodním následníkem aktuálního uzlu.
5. **Upper** je reference na uzel, který se nachází přímo pod aktuálním uzlem a slouží k přechodům mezi jednotlivými úrovněmi.

6. **Bottom** je reference, která ukazuje na uzel, který se nachází přímo nad aktuálním uzlem, a je tedy opakem reference Upper.

6.3.2 Simulace

Samotný výpočet metody založené na simulacích průchodů využívá implementovanou datovou strukturu jako opěrný základ. Důležitou částí je způsob zajištění správného generování cest binomickým stromem, proto byl navržen jednoduchý způsob, jak generovat náhodná čísla, tak aby se dalo ovlivnit, zda budou spíše pozitivní nebo negativní. Způsob, jak toho bylo docíleno, je přiblížen v následující sekci 6.3.3.

Vstupní parametry byly stanoveny na základě existujících prací [6] [11] [5] Parametry vstupující do výpočtu:

- **Počáteční investice do on-premise infrastruktury** - Cena, kterou musíme zaplatit na začátku projektu za on-premise infrastrukturu
- **Nízká anuita** - Představuje poplatky za navýšení výkonu on-premise infrastruktury v podobě menších vylepšení (paměť, software, ...)
- **Střední anuita** - Představuje poplatky za navýšení výkonu on-premise infrastruktury v podobě dražších vylepšení (uložiště, procesor, ...)
- **Vysoká anuita** - Představuje poplatky za navýšení výkonu on-premise infrastruktury v podobě hodně velkého navýšení výkonu (servery, počítače, ...)
- **Počáteční investice do cloudu** - Cena, kterou musíme zaplatit za počáteční konfiguraci infrastruktury v cloudu
- **Cena poplatku za změnu cloudu** - Poplatek za navýšení výpočetního výkonu v cloudu
- **Bezriziková úroková míra**
- **Volatilita** - Úroveň proměnlivosti trhu, čím vyšší volatilita je, tím více a častěji se trh mění a je méně stabilní
- **Doba životnosti projektu**
- **Počet období**
- **Počet simulací**

V průběhu simulace si neseme list, do kterého ukládáme poplatky za on-premise, protože se musí odepisovat postupně a nejedná se tedy o jednorázovou akci jako v případě cloudu. To znamená, že v případě, že se posouváme do úrovně, kterou jsme ještě nenavštívili, tak do tohoto listu přidáme odpovídající hodnotu anuity. S tímto listem můžeme vypočítat vždy v aktuálním uzlu hodnotu nákladů na on-premise. Výpočet hodnoty nákladů na cloudové řešení

infrastruktury je jednoduchý, jedná se o jednorázovou akci, takže v aktuálním uzlu vždy snížíme nebo povýšíme cenu za cloud. S těmito dvěma hodnotami už můžeme jednoduše dopočítat rozdíl v nákladech jejich rozdílem.

Hodnoty za cloud není potřeba při samotných simulacích počítat, protože se nachází v datové struktuře, kde se při jejím vytváření automaticky dopočítávají a při simulaci už jenom čteme připravenou hodnotu. Výsledné diskontované vnitřní hodnoty se cestou sčítají, až nám ke konci vyjde jedno číslo, které určuje celkový rozdíl nákladů na cloud a on-premise IT infrastrukturu a které je uloženo jako výsledek první simulace. Tento proces se opakuje podle počtu na začátku zvolených simulací. Po skončení všech simulací proběhne výpočet aritmetického průměru nad množinou výsledků simulací a tím získáme informaci o tom, kolik průměrně ušetříme při použití cloudu na celkových nákladech.

6.3.3 Generování průchodů

Aby průchody binomickým stromem zohledňovaly reálnou situaci, byl navržen způsob jak generovat náhodné průchody stromem tak, aby průchody byly ovlivněny volatilitou. K tomu byly použity pravděpodobnosti růstu a poklesu, k jejichž výpočtu je potřeba právě zmiňované volatility. Samotný proces rozhodování, zda se z daného uzlu posuneme směrem nahoru nebo dolů, se pak skládá ze dvou kroků, prvním je vygenerování náhodného čísla mezi 0 - 1, krokem druhým je rozhodnutí, zda vygenerované číslo je větší nebo rovno rozdílu 1 a pravděpodobnosti růstu. Pokud tato rovnost platí, tak se z daného uzlu posuneme směrem nahoru, v opačném případě se posuneme směrem dolů. Tím máme zajištěno, že změna volatility bude ovlivňovat, zda budou cesty více pozitivní či negativní.

6.3.4 Grafické rozhraní

Hlavní a prakticky jedinou část grafického uživatelského rozhraní (GUI) tvoří dva formuláře. Jeden umožňující zadávat vstupní parametry pro hlavní metodu využívající simulace a druhý pro vedlejší metodu reálných opcí. Celé rozhraní je velmi jednoduché a intuitivní. GUI není stěžejním bodem hodnocení, jelikož šlo v první řadě o funkčnost a použitelnost nástroje, grafické uživatelské rozhraní bylo implementováno hlavně z důvodu jednoduchého vkládání vstupních parametrů pro výpočet.

Vzhled uživatelského rozhraní lze vidět na obrázcích 6.2 a 6.3.

6.3.5 Generování reportů

Pro získání více informací bylo rozhodnuto, že by nástroj měl generovat jednoduché reporty, ve kterých by byla zobrazena průchodnost jednotlivými uzly. Na tom je velmi dobře vidět porovnání mezi pozitivními a negativními cestami stromem. Reporty se generují ve formátu xlsx pro Microsoft Excel. Vybrán byl pro jednoduchost reprezentace stromové struktury. U metody zaměřené na simulace se generují reporty pozorující průchodnost uzlů z

Simulace	Real option
Initial on-premise fee	<input type="text"/>
Regular annuity 0-9	<input type="text"/>
Larger annuity 10-19	<input type="text"/>
Large annuity 20-29	<input type="text"/>
Initial cloud fee	<input type="text"/>
Regular cloud fee	<input type="text"/>
Volatilita	<input type="text"/>
Interest rate	<input type="text"/>
Life time of project	<input type="text"/>
Time steps	<input type="text"/>
Number of simulations	<input type="text"/>
<input type="button" value="Calculate"/>	
Result	<input type="text"/>

Obrázek 6.2: GUI pro metodu založenou na simulacích [autor]

různých úhlů (UpperParent, BottomParent, Right, Left), jedná se buď o to kudy do uzlu cesta přišla, nebo odešla. Zatímco u metody reálných opcí se generuje report zobrazující strom, v jehož začátku se nachází výsledná hodnota flexibility neboli výsledek této metody.

Jak takový report vypadá, můžete vidět na obrázku 6.4

6.4 Použitá technologie

Pro vývoj a implementaci nástroje byl zvolen objektově orientovaný programovací jazyk Java. Důvod proč jsem si zvolil právě Javu byl ten, že se jedná o jeden z nejpoužívanějších a nejpoblárnějších jazyků a zároveň je to uživatelsky přívětivý programovací jazyk. Předností Javy je její přenositelnost, která umožňuje použití programů na různých operačních systémech.

Kapitola 7

Implementace

Po kapitole, která se věnovala obecnému návrhu podpůrného nástroje, se přesouváme ke kapitole, ve které je postupně popsán postup implementace tak, jak probíhala krok po kroku až k výslednému produktu. Při této cestě jsou přiblíženy i problémy, které bylo nutné během implementace řešit a jakým způsobem ovlivnily další postup v implementaci.

7.1 Průběh implementace

V této sekci se podíváme krok po kroku na to, jak postupně probíhala implementace. Jako základ, na kterém byl následně implementován celý nástroj, byl použit kód skládající se z [12] a [13] z GitHubu, kde se nachází pod “GNU General Public License v3.0”, které dohromady tvoří jeden program, který počítá hodnotu evropské/americké opce. Tento kód byl použit jako základ práce s tím, že část zabývající se výpočtem hodnot opcí byla celá přepsána a část zabývající se generováním stromů byla pozměněna podle potřeb. V návaznosti na to byl implementován algoritmus pro vytvoření binomického rekombinačního modelu, výpočet vnitřních hodnot uzlů a první pokusy o simulace. Ukázalo se, že je potřeba k simulacím přistoupit trochu jinak a že vytvářet binomický strom v podstatě není potřeba. Aby se využil již naimplementovaný algoritmus, který vytvářel binomický model a počítal vnitřní hodnoty, tak bylo rozhodnuto dodělat poslední část pro výpočet metodou reálných opcí, a to zpětný průchod stromem, díky čemuž podpůrný nástroj kromě metody založené na simulacích nabízí také klasický výpočet metodou reálných opcí. Pro výpočet založený na simulacích byl navržen nový algoritmus, který pouze simuluje průchod stromem. Ten počítá všechny potřebné hodnoty v reálném čase a tak nutně nepotřebuje celou stromovou strukturu, do které by ukládal mezivýpočty. V rámci práce byla tato struktura již dříve naimplementována a tak byla aspoň částečně využita jak na ukládání některých mezivýpočtů, tak na ukládání informací pro generování reportů. Na závěr bylo vytvořeno jednoduché GUI pro snadné zadávání vstupních dat.

7.1.1 Sestavení stromu

Pro sestavení stromu byl využit algoritmus z kódu [12], který vytvářel nerekombinační binomický model pro oceňování opcí. Algoritmus byl upraven tak, aby výsledný binomický model byl rekombinační a obsahoval všechny potřebné reference. Nad takto vytvořenou strukturou už bylo možné začít provádět operace pro nastavení výchozích parametrů a následný výpočet vnitřních hodnot jednotlivých uzlů.

7.1.2 Výpočet hodnoty IT investice

K výpočtu hodnoty IT investice se nejprve přistoupilo stejným způsobem jako v metodě reálných opcí, k tomu byla potřeba implementovaná struktura binomického modelu, která umožňovala uchovávat informace o všech možných cestách do každého uzlu pro stanovení hodnoty nákladů on-premise, veškeré anuity se ukládají do "ArrayListu", který obsahuje další "ArrayListy", jejichž počet odpovídá počtu všech vedoucích cest do daného uzlu. To odpovídá rozvoji "Pascalova trojúhelníku", což znamená, že pro každou další úroveň se zvyšuje celkový počet cest do daného uzlu. To zapříčiní, že od určité úrovně je výpočet paměťově neudržitelný a je tedy z tohoto hlediska omezený.

Tento způsob se ukázal jako nevhodný, protože nechceme simulovat průchody nad stromem, kde jsou spočítány vnitřní hodnoty zohledňující všechny možné cesty pro daný uzel. Proto byl výpočet vnitřní hodnoty přesunut dovnitř simulací, neboť ji počítáme tak, aby zohledňovala pouze cestu, kterou procházíme, to znamená, že nepotřebujeme seznam všech anuit v každém uzlu, neboť nám stačí pouze jedna a to ta, která odpovídá cestě, kterou simulujeme.

Původní přístup byl zanechán a využit při výpočtu klasické metody reálných opcí.

7.1.3 Výpočet hodnoty opce

Výpočet hodnoty opce byl doimplementován jako vedlejší část práce, aby program za prvé nabízel i klasickou metodu reálných opcí pro porovnání a za druhé, aby se využila již naimplementovaná struktura binomického modelu. Jedná se o zpětný průchod stromem, při kterém je vypočítána výsledná hodnota opce. Podrobnější popis výpočtu lze najít v pracích [5], [6], [11].

7.1.4 Simulace

Klíčovou částí pro metodu založenou na simulacích bylo určit, jakým způsobem se budou průchody generovat. Při prvním pokusu bylo ke generování přistoupeno tak, že se náhodně generovala čísla mezi 0 a počtem všech existujících cest binomickým modelem, který odpovídá hodnotě

$$2^n \quad (7.1)$$

, kde n se rovná počtu období, na která je binomický model rozdělen. Takto získané číslo se následně převedlo na pole bitů a podle bitů se rozhodovalo, jakým směrem se z daného uzlu má jít dál, přesněji pro 0 směrem dolů, pro 1 směrem nahoru. Takto generované průchody se zdály zprvu jako ideální řešení, jenže se ukázalo, že to vlastně není úplně to, co jsme chtěli, protože bylo potřeba, aby se po každém kroku znovu rozhodovalo, kudy se půjde, a u takto generovaných cest bylo takovéto rozhodnutí jen jedno a vytvořila se naráz celá cesta. Dále se ukázalo vhodným mít možnost ovlivnit, zda se budou generovat spíše pozitivní nebo negativní cesty.

V novém přístupu jsme se rozhodli přistoupit k samotné generaci trochu jinak, už se negenerovaly celé cesty najednou, ale postupně se generovaly krok po kroku. To znamená, že se při přesunu vygenerovalo desetinné číslo mezi 0 - 1, které se zaokrouhlilo, a na základě toho se rozhodlo, zda se posunout do horního nebo spodního následníka. Tento proces se opakoval u každého posunu. Abychom byli schopní ovlivnit, zda se budou generovat spíše pozitivní nebo negativní cesty, tak se do procesu generování zapojila volatila v podobě pravděpodobnosti růstu, která je schopná posouvat hranice pro zaokrouhlování. Vzorec lze vidět níže.

$$n \geq 1 - \text{tree.getP}(); \quad (7.2)$$

Na levé straně proměnná “ n ” představuje vygenerované desetinné číslo mezi 0 - 1 a na straně pravé “ $1 - \text{tree.getP}()$ ” představuje novou hranici, od které se zaokrouhlí vygenerované číslo na 1. Vzniká tak, že se od jedničky odečte pravděpodobnost růstu, čímž získáme hodnotu, od které se vygenerované číslo zaokrouhlí na 1 a půjde tedy o posun směrem nahoru.

7.1.5 GUI

Pro intuitivní zadávání vstupních parametrů bylo vytvořeno úplně jednoduché uživatelské rozhraní, které lze vidět v sekci 6.3.4. Jediným smyslel rozhraní je předat uživateli parametry programu pro výpočet.

7.1.6 Generování reportů

Reporty byly přidány, aby bylo možné získat informace, které by se mohly v některých případech hodit. Všechny reporty jsou generovány ve formátu **xlsx** pro “Microsoft Excel”. V případě metody založené na simulacích se generuje sešit, ve kterém jsou čtyři strany, na kterých se nacházejí stromové struktury reprezentující binomický model a jeho jednotlivé uzly, čísla uvedená na místech uzlů pak říkají, jaká byla průchodnost uzlem v daném směru. Průchodnost se počítá ze čtyř směrů, a to jako množství cest příchozích ze spodního nebo horního předchůdce a množství cest odchozích do spodního nebo horního následníka.

Pro ukládání informací ohledně průchodů jednotlivými uzly během jednotlivých simulací se příhodně využila již naimplementovaná struktura binomického stromu, která je v pozadí a lze do ní tedy ukládat informace tohoto

typu, jelikož se po skončení jedné simulace neztrácí a zůstává v pozadí po celou dobu výpočtu. Po poslední provedené simulaci se pouze provede jeden průchod stromem, při kterém se uložené informace o průchodnosti zapíše do “Excelovského” souboru.

V případě vedlejší metody reálných opcí se generuje “Excelovský” soubor, který obsahuje vykreslený binomický model po provedení zpětného průchodu a tedy s výslednou hodnotou opce na začátku stromu.

7.2 Zhodnocení

Implementace z velké části splnila očekávání a kromě implementace trinomického modelu se stihlo vše, co jsme si zadefinovali. V případě trinomického stromu vyvstal problém v podobě otázky, jak generovat náhodné číslo mezi 0 - 1 - 2 pro rozhodování, jakým směrem se vydat z aktuálního uzlu, když výsledek chceme ovlivnit podle míry volatility. Hlavním problémem byl ovšem čas, který už nám neumožňoval se více zaměřit na trinomický model a jeho využití. Ve zbylém čase jsme se zaměřili hlavně na experimenty a testování samotného nástroje.

Následující kapitola se věnuje testování implementovaného nástroje.

Kapitola 8

Testování

Jelikož navrhnout testovací data pro testování nástroje není lehké, tak bylo po dohodě s vedoucím práce určeno, že použijeme již existující scénáře z jiných prací. Pro testování byly použity stejné testovací scénáře, které byly navrženy v rámci práce [11], která se zabývá stejnou problematikou. Kapitola je rozdělena na několik částí, a to na testování správnosti výpočtu IT investice a následné experimenty s různými vstupními daty, jako je volatilita a délka a počet období. U každého testu je vysvětleno, proč vychází zrovna tak, jak vychází, a proč se třeba liší od výsledků metody reálných opcí.

Data vychází z odborné literatury a na základě již existujících prací. Ve vstupních datech se nachází například různé navyšování investic do on-premise infrastruktury, které znázorňuje možné nadhodnocení a zakoupení více výkonu, než bylo potřeba, což se projeví větší neefektivitou on-premise, protože budeme více prodělávat. Navyšování investic je promítnuto pomocí anuit, které představují způsob, jak lze porovnávat smysluplně cloud a on-premise, tak aby bylo porovnáváno porovnatelné. Hodnoty těchto anuit získáme z odepisování nakoupeného hardwaru. Po každých 10 úrovních se hodnota anuit liší, to je způsobeno tím, že předpokládáme, že ne vždy budeme kupovat stejně drahý hardware. Častěji budeme kupovat třeba jenom paměti a jednou za čas dokoupíme nový server nebo provedeme diametrálně větší navýšení výkonu.

8.1 Experimentální testování

Cílem těchto scénářů je zjistit, jak se bude chovat výpočet, pokud budeme měnit určité vstupní parametry, jako jsou délka projektu a počet období, navyšování investic do on-premise a cloudu.

U hodnot vycházejících z metody založené na simulacích je patrné, že oproti hodnotám z metody reálných opcí je patrné, že se pohybují řádově v desetitisících, zatímco u metody reálných opcí se pohybují řádově ve stovkách. To je způsobeno tím, že při průchodu generované cesty při simulacích se ve výpočtu vnitřních hodnot uzlů nepoužívá pravděpodobnost navštívení daného uzlu, která hlavně ve vysokých úrovních způsobuje enormní zmenšení výsledné vnitřní hodnoty. V našem případě tato hodnota symbolizuje celkové průměrné náklady.

8.1.1 Různá volatilita

V prvním experimentu jsme se zaměřili na reakci způsobenou různou mírou volatility, jinak by se také dalo říci proměnlivostí či nejistotou trhu. Volatilita blížící se nule znamená prakticky nulové riziko a představuje stabilní ustálený trh. Volatilita přesahující jedničku na druhou stranu představuje rychle se měnící nestabilní trh.

Vstupní data

Scénáře v tomto experimentu se zabývají tím, jak volatilita ovlivňuje výpočet hodnoty IT investice. Vstupní hodnoty do obecné části výpočtu zobrazuje tabulka 8.1. Bylo provedeno dohromady šest testů. Testy byly prováděny pro dvouleté projekty rozdělené na 24 období. Bezriziková úroková míra byla zvolena 1 % na základě výnosnosti státních dluhopisů. Cílem experimentu bylo zjistit reakci výpočtu na různé hodnoty volatility, proto byly zvoleny různé hodnoty pro každý test. Hodnota volatility byla postupně zvyšována, což představuje zvyšující se riziko.

Test:	1	2	3	4	5	6
T	2					
n	24					
r	0,008					
σ^2	nízká 0,1	střední 0,3	vyšší 0,5 0,7		vysoká 0,9 1,1	
Anuita (počáteční)	10 083					
Anuita (běžná)	483					
Anuita (vyšší)	903					
Anuita (nejvyšší)	1 323					
Cloud (počáteční)	10 000					
Cloud změna	400					
Počet simulací	100 000					
Rozdíl v nákladech	23 449	25 192	26 567	27 969	29 407	30 915
	23 532	25 175	26 585	27 911	29 419	30 977
	23 410	25 154	26 609	28 033	29 449	30 866
	23 515	25 188	26 541	27 828	29 385	30 942
	23 483	25 194	26 528	28 003	29 414	30 919
	23 500	25 167	26 527	27 896	29 420	31 019
	23 505	25 186	26 581	28 037	29 365	30 970
	23 480	25 202	26 563	27 958	29 457	30 870
	23 550	25 137	26 505	27 916	29 437	31 019
	23 489	25 168	26 562	27 948	29 415	30 922

Tabulka 8.1: Experiment 1 s různou volatilitou

■ Vyhodnocení

Na výsledcích je vidět, že se hodnoty mírně liší, a to hlavně v řádech, takže zatímco v metodě reálných opcí vychází hodnoty v řádu stovek, tak u simulací se hodnoty pohybují o dva řády výše, jinak jsou výsledné hodnoty podobné, což dává smysl. Rozdíl o dva řády je způsoben tím, že metoda reálných opcí hodnoty násobí pravděpodobností příchodu do uzlu, kterou při simulacích vynecháváme, protože o tom, do jakého uzlu se bude pokračovat se rozhoduje pomocí náhodně generovaného čísla ovlivňovaného volatilitou. Každá konfigurace při testech byla spuštěna 10x, aby bylo ukázáno, že simulace jsou náhodné a tedy i výsledek je pokaždé trochu jiný. To, jak se výsledek mění, je vidět na těchto deseti bězích.

Když se podíváme na výsledky, tak můžeme vidět, že výsledné hodnoty pomalu rostou stejně jako u metody reálných opcí. To potvrzuje, že u simulací se chování výstupních hodnot s ohledem na změnu vstupních parametrů chová stejně jako u metody reálných opcí, to znamená, že je u obou metod stejné chování při změně vstupních parametrů.

■ 8.2 Různá délka životnosti a počet období

Druhý experiment se zaměřuje na vliv délky projektu a počtu období na hodnotu IT investice. Provedli jsme pět scénářů s rozdílnou délkou životnosti projektu a rozdílným počtem období. Jediná délka období zůstává stejná, a to jeden měsíc. Volatilita je nastavena na vyšší hodnotu, která by měla odpovídat tomuto odvětví [8]. Souhrn všech pěti běhů lze vidět v tabulce 8.2.

■ Vstupní data

Vstupní data jsou téměř totožná s předchozím experimentem až na to, že zde se liší délka životnosti projektu a počtu období.

■ Vyhodnocení

Je vidět, že s přibývajícím počtem období se rozdíl v celkových nákladech postupně zvětšuje, jak se rozvírají nůžky rozdílu u nákladů na on-premise a cloud IT infrastrukturu. Čím je počet období větší, tak tím jsou také rozdíly větší a bez násobení pravděpodobností se číslo nezmenšuje jako u metody reálných opcí. Z tohoto důvodu můžeme vidět, že s přibývajícím počtem období se ušetřené náklady více a více kumulují.

I zde je vidět, že je zachován trend stejný jako u metody reálných opcí, to lze vidět na tom, jak se výsledné hodnoty postupně zvětšují stejně jako u metody reálných opcí. Je ovšem vidět, že výsledné hodnoty se rapidně zvětšují s přibývajícím počtem období a delší životností projektu. To je způsobené tím, že se při výpočtu nepočítá s pravděpodobnostmi, které by jinak vysoké vnitřní hodnoty vznikající ke konci stromu snižovaly.

Test:	1	2	3	4	5
T	0,25	0,5	1	1,5	2
n	3	6	12	18	24
r	0,008				
σ^2	0,7				
Anuita (počáteční)	10 083				
Anuita (běžná)	483				
Anuita (vyšší)	903				
Anuita (nejvyšší)	1 323				
Cloud (počáteční)	10 000				
Cloud změna	400				
Počet simulací	100 000				
Rozdíl v nákladech	892	2 989	9 353	17 773	27 992
	890	2 990	9 378	17 852	28 012
	889	2 986	9 362	17 832	27 955
	889	2 987	9 340	17 824	27 918
	887	2 982	9 357	17 867	27 950
	890	2 983	9 371	17 874	27 989
	891	2 989	9 346	17 847	27 916
	891	2 981	9 357	17 815	27 987
	889	2 989	9 348	17 834	27 968
	891	2 988	9 329	17 834	27 965

Tabulka 8.2: Experiment 2 s různou délkou a počtem období

8.3 Různé investice do HW

Následující scénáře zkoumají situace, kde majitel záměrně nadhodnocuje investice do on-premise vybavení. K této situaci by pravděpodobně mohlo dojít ve skutečnosti, protože ve skutečnosti nemůžeme navyšovat výkon on-premise vždy, když se zvýší na úroveň, na které jsme ještě nebyli.

Vstupní data

Další experiment byl zaměřený na ověření reakcí týkajících se různých investic do HW a jejich vlivu na výslednou hodnotu. Období opět představuje jeden měsíc. Ve scénářích uvažujeme dvě strategie. Strategie A začíná se standardní počáteční investicí do on-premise vybavení. Při navyšování výkonu kupujeme nadhodnocené vybavení, takže ho ve výsledku nevyužijeme efektivně celé. Strategie B je velmi podobná s tím, že nadhodnotíme i počáteční investici do on-premise vybavení.

Vstupní data zůstávají stejná jako u předchozím experimentů.

Test:	1	2	3	4	5
T	2				
n	24				
r	0,008				
σ^2	0,7				
Anuita (počáteční) A	10 083				
Anuita (počáteční) B	10 083	11 050	11 533	12 016	12 499
Anuita (běžná)	483	966	1 449	1 933	2 416
Anuita (vyšší)	903	1 386	1 870	2 353	2 836
Cloud (počáteční)	10 000				
Cloud změna	400				
Počet simulací	100 000				
Rozdíl v nákladech A	27 945	44 425	60 939	77 702	94 045
Rozdíl v nákladech B	27 990	65 754	92 962	122 010	147 040

Tabulka 8.3: Experiment 3 s různými investicemi

■ Vyhodnocení

Je vidět, že podle toho, jak se mění počáteční investice a anuitní poplatky za nakoupený hardware, tak pokud se mění pouze anuitní poplatky a počáteční investice zůstává stejná, tak výsledné hodnoty rostou pomaleji oproti hodnotám, kde se mění i počáteční investice do hardwaru. V pátém scénáři je vidět, že se liší už o zhruba 50 tisíc.

I zde zůstává stejný trend jako u metody reálných opcí. Můžeme vidět, že výsledné hodnoty postupně rostou. Díky tomu, že nepočítáme s pravděpodobnostmi, tak se hodnoty šplhají k vysokým číslům s tím, jak zvyšujeme počáteční investici a anuity za nakoupené vybavení.

■ 8.4 Různé investice do HW a různé poplatky do Cloudu

V posledním experimentu jsme se zaměřili na různé investice do on-premise současně s různými poplatky za cloud a zjišťovali, jaký to bude mít vliv na výslednou hodnotu IT investice. Vstupní data a výsledky můžeme vidět v tabulce 8.4

■ Vstupní data

Vstupní data zůstávají stejná jako u předchozích experimentů s tím, že jsme pouze změnili počáteční investici do on-premise a do cloudu na 15 000 peněžních jednotek. U prvních čtyř scénářů je hodnota anuit z klasické hodnoty snižována po 10 procentech. Naopak poplatky za cloud od pátého scénáře o 10 procent navyšujeme.

Test:	1	2	3	4	5	6	7
T	2						
n	24						
r	0,008						
σ^2	0,7						
Anuita (počáteční)	15 041						
Anuita (běžná)	338	387	435	483			
Anuita (vyšší)	632	723	813	903			
Cloud (počáteční)	15 000						
Cloud změna	400				440	480	520
Počet simulací	100 000						
Rozdíl v nákladech	22 058	23 772	25 377	27 060	27 917	28 964	29 939

Tabulka 8.4: Experiment 4 s různými poplatky

■ Vyhodnocení

Z výsledných hodnot můžeme vidět, že i v tomto případě byl potvrzen trend stejný jako u metody reálných opcí. Výsledné hodnoty odpovídají hodnotám z metody reálných opcí s tím, že jak nepočítáme u simulací s pravděpodobnostmi, tak jsou o dva řády vyšší.

Kapitola 9

Aktuální stav a možnosti dalšího vývoje

9.1 Aktuální stav

Cílem práce bylo seznámit se s teorií reálných opcí a získané znalosti využít k návrhu podpůrného nástroje, který by kromě klasické metody výpočtu reálných opcí nabízel alternativní způsob výpočtu. Nástroj bylo potřeba navrhnout, implementovat a také otestovat.

V první řadě byla práce zaměřena na binomický model a na výpočty s ním spojené. Nejdůležitější částí bylo navrhnout, jak by metoda založená na simulacích měla vlastně fungovat, jak by měla k výpočtu přistupovat. Což obnášelo vymyslet nebo najít vhodnou datovou strukturu a k ní vymyslet algoritmy a průchody touto strukturou.

Prvním velkým problémem bylo zjištění, že jsme se na celou problematiku dívali špatně, protože jsme mysleli, že musíme počítat s pravděpodobnostmi, že do daného uzlu dorazíme, což se později ukázalo jako mylný předpoklad. Ve skutečnosti nás tato pravděpodobnost nezajímá, protože o tom, do kterého uzlu se vydáme, rozhodujeme pomocí náhodného generátoru. To zapříčinilo, že v první části byla implementována datová struktura, která se později ukázala u výpočtu metodou založenou na simulacích prakticky zbytečná. Aby to nebyl úplně ztracený čas, tak jsme se s vedoucím práce domluvili, že to tam necháme a doděláme celou metodu reálných opcí, kterou to bude nabízet jako alternativu, aby se obě metody mohly porovnávat. Toto a další menší technické problémy způsobily zdržení, kvůli kterému nebyla nakonec implementována a zjištěna použitelnost trinomického modelu namísto binomického.

Dalším problémem se ukázalo získání vstupních dat, neboť jak lze zjistit i v ostatních pracích, tak k této problematice nejsou žádné reálné příklady a tak bylo rozhodnuto, že použijeme vstupní data, která byla navržena ve spolupráci s vedoucím práce u předchozích prací.

Jedná se o univerzální nástroj, což znamená, že do budoucna se dá libovolně rozšiřovat, ať už o trinomický model, který se nestihl, nebo o další alternativní metody výpočtu. Také by se dalo věnovat tomu, jak vylepšit rychlost výpočtu v případě větších instancí.

Aktuálně je tedy kromě tohoto textu výstupem funkční aplikace implementovaná v programovacím jazyce Java, která je vhodná pro provádění experimentů a porovnávání s klasickou metodou reálných opcí. Své využití

najde pravděpodobně především v akademické sféře s tím, že se jedná o aplikaci, kterou lze jednoduše rozšiřovat o další funkcionalitu.

Kapitola 10

Závěr

Cílem práce bylo se seznámit s problematikou hodnocení investic, metodou reálných opcí, binomickým a trinomickým modelem oceňování opcí. Pomocí získaných znalostí navrhnout na základě již existujících prací alternativní způsob výpočtu, který by rozšířil ty stávající. Nový alternativní způsob výpočtu je postavený na základě simulací. Prvním krokem bylo tedy zjistit, jak vhodně využít simulace při výpočtu. Všechny stanovené cíle mimo trinomického modelu, na který nám nezbylo dost času, se podařilo úspěšně splnit a ověřit pomocí testů.

V průběhu práce jsme stanovili, že simulace budou probíhat nad binomickým stromem s tím, že místo využití pravděpodobnosti navštívení jsme se po každém kroku rozhodovali náhodně, kam půjdeme, zda nahoru nebo dolů.

Po návrhu metody založené na simulacích jsme se pustili do samotné implementace. Při implementaci jsme řešili několik situací, při kterých jsme zjistili, že původní návrh měl drobné nedostatky, takže byl postupně ještě upravován. Ve výsledku vznikl nástroj, který umožňuje jak výpočet pomocí metody založené na simulacích, tak zároveň výpočet pomocí metody reálných opcí. Každá metoda vrací lehce rozdílné výsledky z důvodu použití/nepoužití pravděpodobností při výpočtu.

Nástroj se dá dále rozšiřovat přidáváním další funkcionality v podobě dalších alternativních metod výpočtu.

Největším přínosem pro mě byla možnost se seznámit s úplně novou a relativně mému zaměření odlišnou problematikou reálných opcí a hodnocení IT investic obecně. Získané zkušenosti se jednoho dne budou určitě hodit v profesním životě, takže věřím, že se jednalo o přínosnou práci. Obecně se zamýšlet nad věcmi z ekonomického hlediska byla zajímavá zkušenost, kde si člověk uvědomí, že ve skutečnosti jde o mnohem více věcí.



Literatura

- [1] ENSI-MARIA, “Iaas, paas, saas – what do they mean?.” <http://cloudonmove.com/iaas-paas-saas-what-do-they-mean/>, August 2017. Accessed on 2019-04-05.
- [2] R. Chellapa, *Intermediaries in Cloud-Computing: A New Computing Paradigm*. INFORMS meeting in Dallas, 1997.
- [3] D. C. Plummer, T. J. Bittman, T. Austin, D. W. Cearley, and D. M. Smith, *Cloud Computing: Defining and Describing an Emerging Phenomenon*. Gartner Analysis, 2008.
- [4] D. C. Plummer, D. M. Smith, T. J. Bittman, D. W. Cearley, D. J. Cappuccio, D. Scott, R. Kumar, and B. Robertson, *Five Refining Attributes of Public and Private Cloud Computing*. Gartner Research, 2009.
- [5] P. Náplava, *Reálné opce a jejich využití v prostředí IT technologií*. České vysoké učení technické v Praze, 2017.
- [6] V. Trnka, *Nástroj pro podporu výpočtu hodnoty reálných opcí*. České vysoké učení technické v Praze, 2017.
- [7] H. Scholleová, *Hodnota flexibility. Reálné opce*. C. H. Beck, první vydání, 2007.
- [8] V. Nosková, *Možnosti využití cloudů a reálných opcí pro začínající technologické firmy*. ČVUT Praha, 2016.
- [9] M. Abrahámová, *Binomický a trinomický model oceňování opcí*. Zápa-dočeská univerzita v Plzni, 2015.
- [10] L. Ambrož, *Oceňování opcí*. C. H. Beck, první vydání, 2002.
- [11] M. Mastný, *Real Option Analysis and cloud computing investments*. Czech Technical University in Prague, Faculty of Electrical Engineering, 2017.
- [12] D. Dinter, “java-coxrossrubinstein.” <https://github.com/danieldinter/java-coxrossrubinstein>, Apr 2016. (accessed: 04.05.2019).

- [13] D. Dinter, “java-treemodel-dfs.” <https://github.com/danieldinter/java-treemodel-dfs>, Apr 2016. (accessed: 04.05.2019).



Příloha A

Seznam použitých zkratk

GUI Graphical User Interface (Grafické uživatelské rozhraní)

HW Hardware

SW Software

IaaS Infrastructure as a Service (Infrastruktura jako služba)

SaaS Software as a Service (Software jako služba)

PaaS Platform as a Service (Platforma jako služba)

IT Informační technologie

MS Microsoft

Příloha B

Obsah přiloženého CD

popis.txt.....	stručný popis obsahu CD
app.....	implementovaná aplikace
├─ solution.....	zdrojové soubory a projekt pro JAVA IDE
├─ option simulation tool.jar.....	
src.....	zdrojová forma textu práce ve formátu L ^A T _E X
reporty.....	ukázka reportů
├─ Real_Option_Method_Report.xlsx.....	
├─ Simulation_Method_Report.xlsx.....	
text.....	text práce ve formátu PDF
├─ BP_Brzobohatý_Miroslav_2019.pdf.....	