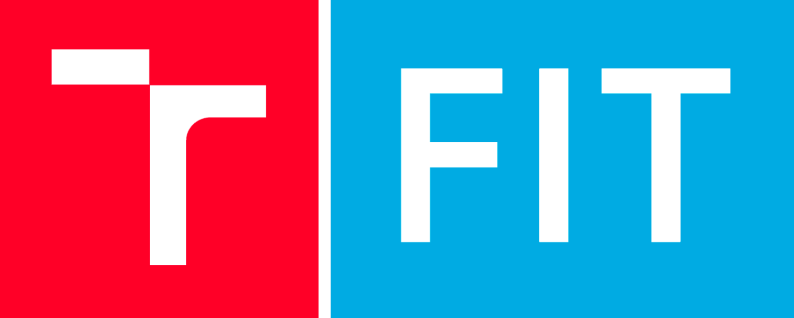
FAKULTA INFORMAČNÍCH TECHNOLOGÍ

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



***Modelování a simulace (IMS)***

***Zákaznická podpora***

***Téma: Služby, infrastruktura a energetika***

**Autoři:** Milan Hruban, xhruba08

David Hél, xhelda00

4. prosince 2017

**Obsah**

[**1.** **Úvod** 3](#_Toc500092259)

[**1.1** **AUTOŘI** 3](#_Toc500092260)

[**1.2** **PROSTŘEDÍ A PODMÍNKY OVĚŘOVÁNÍ VALIDITY MODELU** 3](#_Toc500092261)

[**2.** **Rozbor tématu a použitých metod/technologií** 4](#_Toc500092262)

[**2.1** **LIVECHAT** 4](#_Toc500092263)

[**2.3** **TECHNIK** 5](#_Toc500092264)

[**2.4** **POUŽITÉ POSTUPY** 5](#_Toc500092265)

[**2.5** **PŮVOD POUŽITÝCH METOD** 5](#_Toc500092266)

[**3.** **Koncepce** 5](#_Toc500092267)

[ **Modelářská témata** 6](#_Toc500092268)

[**3.1** **Z** 6](#_Toc500092269)

[**3.2** **FORMY KONCEPTUÁLNÍHO MODELU** 6](#_Toc500092270)

[ **Implementační témata** 6](#_Toc500092271)

[**4.** **Architektura simulačního modelu/simulátoru** 6](#_Toc500092272)

[**4.1** **MAPOVÁNÍ ABSTRAKTNÍHO MODELU DO SIMULAČNÍHO** 6](#_Toc500092273)

[**5.** **Podstata simulačních experimentů a jejich průběh** 6](#_Toc500092274)

[**5.1** **POSTUP EXPERIMENTOVÁNÍ** 6](#_Toc500092275)

[**5.2** **DOKUMENTACE EXPERIMENTŮ** 6](#_Toc500092276)

[**5.3** **ZÁVĚRY EXPERIMENTŮ** 6](#_Toc500092277)

[**6.** **Shrnutí simulačních experimentů a závěr** 7](#_Toc500092278)

# **Úvod**

Tato práce se zabývá problematikou zákaznické podpory ve firmě poskytující hosting herních a virtuálních serverů. V následujících kapitolách si rozebereme vybrané téma, architekturu simulačního modelu, simulační experimenty a základní informace o tvorbě simulátoru pomocí dostupné knihovny SIMLIB k programovacímu jazyku C++ .

Zadáním projektu bylo zaměřit se na problematiku v některé z oblasti služeb, infrastruktury či energetiky a pomocí experimentů zjistit možná rizika, možnosti rozvoje, ekonomické aspekty a podobně. Pro konkrétní případ jsme zvolili zákaznickou podporu společnosti poskytující herní servery. Cílem našeho projektu bylo modelovat současný způsob zpracovávání zákaznických požadavků a pokusit se navrhnout nový způsob, který bude efektivnější z ekonomického hlediska nebo z pohledu komfortu zákazníků. V dalších kapitolách jsou popsána námi navržená vylepšení a výsledky experimentů.

Díky simulaci lze snadněji pochopit zpracování jednotlivých požadavků přicházejících na zákaznickou podporu a jejich následnou obsluhu. Experimentováním jsme schopni ukázat…

Asdasdasdasdasd TODO

## **AUTOŘI**

Na vypracování projektu do předmětu Modelování a simulace spolupracovala dvojice studentů bakalářského studia. Jednotlivé úkoly si oba členové týmu rozdělili následovně:

* **Milan Hruban (vedoucí týmu)** – návrh modelu, implementace
* **David Hél** – tvorba dokumentace, zjišťování informací

Konzultace a odborná fakta související se zákaznickou podporou nám poskytl vedoucí podpory ve společnosti Hicoria, jež se zabývá poskytováním herních serverů. K projektu bylo navíc využito následující odborné a pomocné literatury:

[1]

[2]

[3] PERINGER, Petr. *Popis simulační knihovny SIMLIB* [online]. Vysoké učení technické v Brně, 1997 [cit. 2017-11-27]. Dostupné z: http://www.fit.vutbr.cz/~peringer/SIMLIB/doc/html-cz/

## **PROSTŘEDÍ A PODMÍNKY OVĚŘOVÁNÍ VALIDITY MODELU**

Ověřování validity modelu probíhalo následujícím způsobem. Z námi vytvořeného modelu byla pomocí simulačních experimentů získána sada dat, která byla následně konzultována s vedoucím podpory dané firmy. Data byla porovnána se záznamy v databázi požadavků na podporu, kterou si firma vede, a shledána jako odpovídající.

# **Rozbor tématu a použitých metod/technologií**

Veškerá fakta a informace v této kapitole byla zjištěna na základě konzultace s jedním z vedoucích pracovníků (pokud není uvedeno jinak) a jedná se tedy o přesná data z interních záznamů firmy, které si vede pro vlastní potřeby.

Podpora obdrží za den v průměru 160 požadavků. Vzhledem k tomu, že požadavky v rámci dne nepřichází rovnoměrně, je třeba rozdělit den na několik intervalů. V tabulce (*Tabulka 1 – Příchody požadavků v danou hodinu*) jsou zobrazeny jednotlivé intervaly s průměrným počtem zákaznických požadavků. Asi jednou za měsíc dojde k hromadné poruše, která ovlivní větší množství serverů. Porucha je opravena průměrně za 30 minut a během této doby požadavky vznikají 10x rychleji. Tento údaj byl zjištěn výpočtem ze situace, kdy porucha trvala 25 minut, a přišlo 50 požadavků. Jedná se tedy pouze o nepřesný údaj, více dat jsme však nebyli schopni zajistit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabulka příchodu požadavků v danou hodinu | | | | |
| Čas příchodu | **0-12** | **12-18** | **18-22** | **22-24** |
| Počet požadavků | **21** | **77** | **47** | **15** |

Tabulka 1 - příchody požadavků v danou hodinu

## **LIVECHAT**

Zákazník může požádat o podporu buď livechat a nebo může svůj požadavek napsat ve formě zprávy, takzvaného ticketu, který je následně obdržen a zodpovězen pracovníkem podpory. Livechat je k dispozici pouze v pracovní dny od 12:00 do 18:00 a může zároveň vést až 5 konverzací. Pokud je livechat plně obsazený, zákazník může vyčkat na jeho uvolnění nebo napsat ticket. Jestliže je zrovna k dispozici livechat, až 85% zákazníku jej upřednostňuje před možností napsat ticket.

Průměrný počet požadavků na live chat/Průměrný počet obdržených požadavků od 12 do 18 \* 100%

Konverzace na livechatu trvá v případě krátké konverzace přibližně 2 minuty a v případě dlouhé konverzace přibližně 15 minut. Toto dělení vychází z dvou skupin problémů, které se vyskytují: opakující se problémy, na které má zaměstnanec již připravenou odpověď, a neobvyklé problémy, které vyžadují delší popis problému ze strany zákazníka. K přesnému rozdělení zákaznických požadavků do zmíněných skupin neexistují data, ale podle odhadu obsluhy livechatu se vyskytuje mírná převaha opakujících se problémů. Existuje 10% šance, že zákazníkův požadavek nelze vyřešit přes livechat a je odkázán na ticket podporu.

* 1. **TICKETY**

Tickety řeší zaměstnanec podpory, který nemá žádnou pevně stanovenou pracovní dobu. Jeho úkolem je minimalizovat dobu čekání zákazníka. Jeho pracovní režim vypadá následovně: 3x denně zkontroluje, zda jsou ve frontě nějaké nevyřešené tickety (většinou v 11, 16 a 20 hodin) a pokud ano, na tickety odpovídá, dokud všechny nevyřeší. Zhruba 5% ticketů vyžaduje zásah technika – pošle tedy zprávu technikovi s požadavkem. Průměrná doba řešení jednoho ticketu je 2 minuty. Tato výrazně nižší doba než při řešení požadavku na livechatu je způsobena tím, že pomocí ticketů jsou řešeny operace, které na livechatu vyřešit nelze, ale trvají málo času (převod kreditů, zrušení serveru…) a také proto, že jeden ticket je chápán jako dvojice dotaz a odpověď, zatímco požadavek na livechatu může být konverzace, tedy série těchto dvojic.

## **TECHNIK**

Chování technika není pro výsledný systém příliš významné, protože množství požadavků na něj je minimální. Získaná data jsou pouze odhadem zaměstnance podpory. Pokud je technik během dne (8 – 22 hodin) vyzván k řešení ticketu, zprávu zaregistruje průměrně během jedné hodiny. Řešení ticketu mu pak trvá zhruba 2 minuty v případě jednoduchého zásahu, nebo přibližně 30 minut v případě složitějšího problému.

## **POUŽITÉ POSTUPY**

Jediná použitelná metoda k získání informací o modelovaném systému byla konzultace se zaměstnancem firmy, který má přístup ke všem interním datům.

K vytvoření abstraktního modelu jsme zvolili modelování pomocí petriho sítě. Zvolený postup je pro daný problém ideální, protože se jedná diskrétní systém, definovaný stavy a vzájemnou komunikací jednotlivých prvků, která za jistých podmínek způsobí změnu stavu.

Tvorba simulačního modelu proběhla za pomocí jazyka C++ s použitím knihovny SIMLIB.

## **PŮVOD POUŽITÝCH METOD**

* Petriho síť TODO ODKAZY
* C++
* SIMLIB

# **Koncepce**

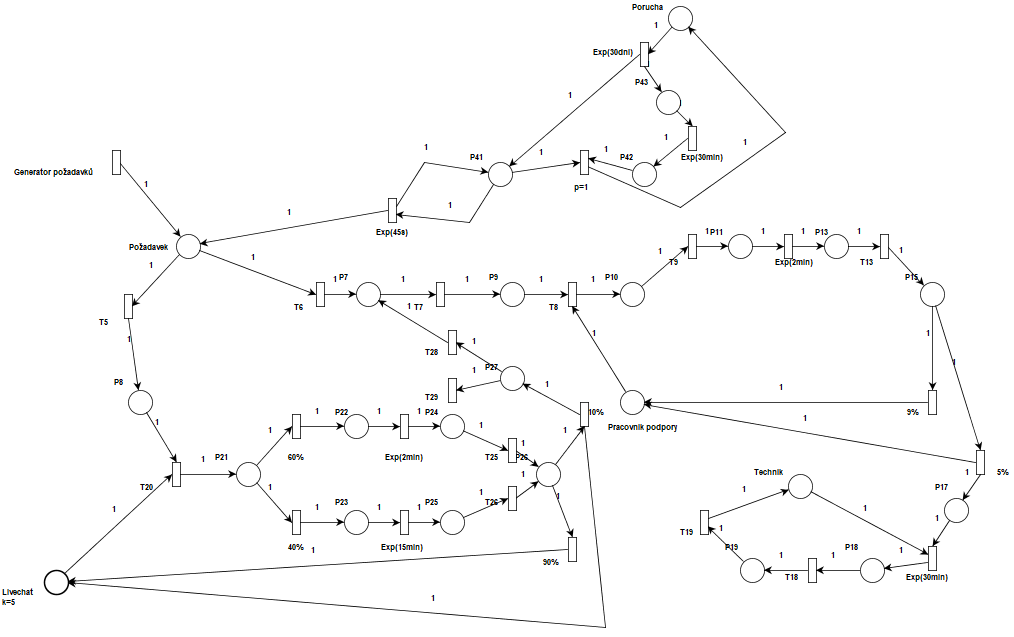
Veškerá fakta zmíněná v předchozí kapitole jsou relevantní pro sestavení abstraktního modelu. TODO opravdu sem mam opisovat znova to stejne\_. wutface

## **Popis konceptuálního modelu**

Pro tvorbu abstraktního modelu jsme zvolili petriho síť (z důvodů popsaných v kapitole 2.4). Na obrázku Obrázek 1 TODO je znázorněn zjednodušený abstraktní model. Zjednodušení za účelem přehlednosti spočívá v nezahrnutí částí, které modelují čas v rámci dne a týdne. Přechod Generátor požadavků tedy generuje požadavky podle pravidel popsaných v kapitole 2. Stejně tak zaměstnanci podpory (stavy Livechat, Pracovník podpory a Technik) jsou v abstraktním modelu znázorněni jako přítomni neustále, ve skutečnosti (a v simulačním modelu) se řídí pravidly popsanými v kapitole 2. Rychlost požadavků vznikajících při poruše se také odvíjí od aktuálního času, pro účely abstraktního modelu byla modelována jako Exp(45s). Jedná se o hrubý odhad průměrné doby v rámci dne.

Přechody T29, znázorňující zákazníka odkázaného zaměstnancem livechatu k napsání ticketu, který ticket nenapíše, a T28, znázorňující zákazníka, který ticket naopak napíše, postrádají pravděpodobnostní ohodnocení. Důvodem je, že nebylo možno získat relevantní informace. V simulačním modelu jsou tyto požadavky opouštějící systém skrz přechod T29 zcela zanedbány, avšak v abstraktním modelu byl přechod ponechán, protože odpovídá modelovanému systému. Validita modelu tímto zjednodušením zůstane nezměněna, protože počet požadavků na livechat a počet ticketů je sledován nezávisle, tedy zákazníka, který byl odkázán k napsání ticketu, ale neučinil tak, chápeme jako splněný požadavek. Pokud ticket napsal, vnímáme jej jako nový požadavek, nezávislý na předchozí komunikaci.

## **FORMY KONCEPTUÁLNÍHO MODELU**



Obrázek 1 - Petriho síť

# **Architektura simulačního modelu/simulátoru**

Simulační model odpovídá abstraktnímu modelu v předchozí kapitole, je však v nějakých ohledech zjednodušen (popsáno v předchozí kapitole) a doplněn o modelování času v rámci dne a dnů v rámci týdne. V následující kapitole je popsáno mapování abstraktního modelu do simulačního. Zahrnuje všechny použité objekty vystupující v simulaci a popisuje jejich chování a roli, kterou zastávají v rámci systému.

## **MAPOVÁNÍ ABSTRAKTNÍHO MODELU DO SIMULAČNÍHO**

Třída Generator

Generátor požadavků zákazníků. Požadavky vytváří v časových intervalech daných aktuální denní dobou.

Třída BreakdownGenerator

Generuje poruchu ovlivňující větší množství serverů a zároveň ovládá i logiku jejího opravení.

Třída CustomerRequirement

Reprezentuje potřebu zákazníka na službu podpory. S ohledem na aktuální denní dobu si požadavek může zabrat *liveChat* a nebo vytvoří instanci třídy Ticket.

Třída BackendWorker

Reprezentuje technika, logiku jeho chování – pokud dostane zprávu o novém ticketu (reprezentován pomocí fronty waitTicketsBackend) ticket vyřeší.

Třída SupportWorker

Představuje zaměstnance, který zpracovává tickety. Bere si tedy tickety z fronty waitTickets. Jeho přítomnost je modelována pomocí atributu operatingHours[], který představuje hodiny, kdy zaměstnanec obvykle kontroluje nové tickety.

Třída Ticket

Proces znázorňující životní cyklus jednoho ticketu. Je zařazen do fronty waitTicket, a čeká, až jej někdo aktivuje (tedy vyřeší).

Dále program využívá již dříve zmíněné Store liveChat, představující pracovníka livechatu a fronty waitTickets a waitTicketsBackend, reprezentující tickety čekající na pracovníka podpory a technika.

# **Podstata simulačních experimentů a jejich průběh**

## **POSTUP EXPERIMENTOVÁNÍ**

fghghgfh

## **DOKUMENTACE EXPERIMENTŮ**

fghghgfh

## **ZÁVĚRY EXPERIMENTŮ**

fghghgfh

# **Shrnutí simulačních experimentů a závěr**

fghghgfh