VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Modelování a simulace – Projekt
SHO v logistike

Matej Cimmerman
Lukáš Kristek

5.12.2022

1 Úvod

1.1 Téma

Projekt rieši modelovanie [1] (s. 8) a simuláciu [1] (s. 8) tzv. systému hromadnej obsluhy [1] (s. 136) (ďalej len ”SHO”) v oblasti logisitky. Za riešený SHO bol vybraný logistický systém prijimania a vybavovania objednávok v reštaurácii Pizza23 v meste Nové Zámky (SR).

Systém rieši prijímanie objednávok, prípravu jedla, balenie objednávok a následnú dopravu zákazníkovi.

1.2 Zdroje informácií

Hlavnými zdrojmi pre obsah SHO boli informácie od zamestnancov danej pobočky podniku Pizza23. Zdrojmi informácií pre implementačné riešenie boli výukové materiály predmetu Modelování a simulace (VUT FIT v Brně) a dokumentácia knižnice SIMLIB.

1.3 Validita informácií

Informácie boli získané od zamestnancov reštaurácie, avšak vzhľadom na dĺžku praxe opýtaných zamestnancov a stálosť časových údajov systému sa tieto informácie dajú považovať za dôveryhodné a použiteľné pre model systému.

1.4 Validita modelu

Údaje získané pomocou experimentov vykonaných nad vytvoreným modelom vo veľkej miere odpovedajú aktuálnemu stavu skúmaného systému, to znamená, že vytvorený model môžme považovať za validný - spĺňa validitu modelu [1] (s. 37).

2 Koncepcia modelu

2.1 Parametre

Medzi sledované parametre systému, ktoré budeme sledovať v experimentoch patria : počet kuchárov, počet rozvozárov, trvanie prípravy jedla a trvanie dopravy online objednávky.

2.2 Petriho sieť modelu

Petriho sieť modelu popisuje proces objednávky. pričom reštaurácia má systém pre osobné a systém pre online objednávky. Obidva druhy objednávok sú generované jedným generátorom a rozdeľujú sa na základe percentuálneho podielu - osobné objednávky tvoria cca. 60% a online objednávky 40%;

Procesy sa generujú v určitých intervaloch. Jeden vygenerovaný proces označuje 1 objednávku, avšak 1 objednávka sa nerovná 1 jedlu, ale celému zoznamu objednaného jedla z danej objednávky. (Kuchár, ktorý si danú objednávku berie, ju väčšinou pripravuje naraz.)

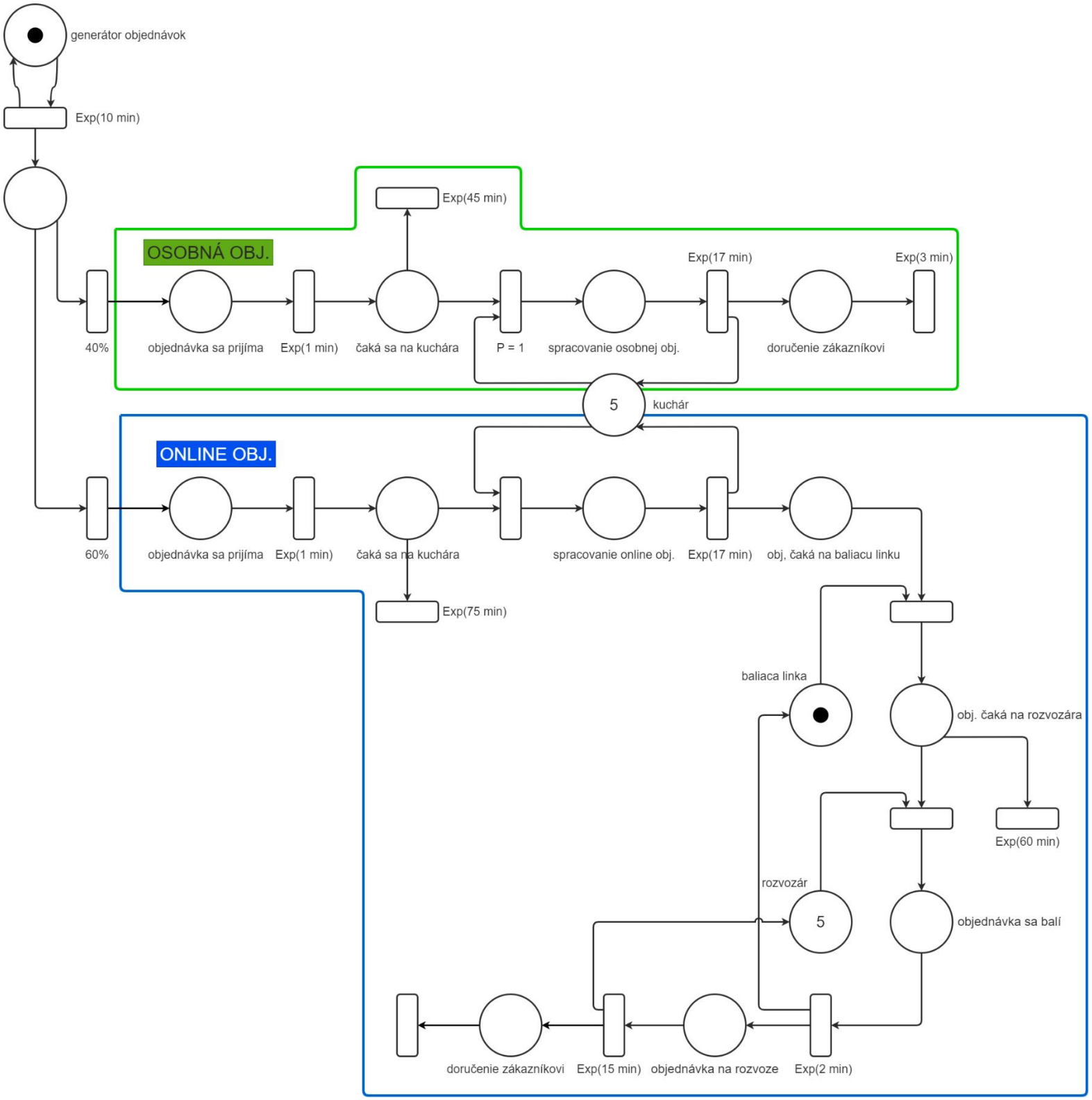
Časť *osobná obj,* popisuje proces objednávky objednanej priamo v reštaurácii. Táto objednávka zahŕňa prijatie čašníkom, prípravu jedla a následné servírovanie zákazníkovi.

Časť *online obj.* popisuje objednávku prijatú cez webovú stránku reštaurácie. Od osobnej sa odlišuje nutnosťou zabalenia jedla, kvôli jeho udržaniu počas prepravy a dopravou objednávky na určené miesto.

Samotní čašníci vybavujúci osobné objednávky nie sú v modeli systému, pretože nie sú pre účely experimentov akokoľvek potrební.

Obidva druhy objednávok zdieľajú spoločnú časť modelu, ktorou sú samotní kuchári, keďže jedlo je pripravované na jednom mieste (v kuchyni).

Zároveň obidva procesy sú modelované s prihliadnutím na netrpezlivosť zákazníkov a po určitej dobe po vybavení objednávky opúšťajú systém, ak nie sú prevzaté jedným z kuchárov, alebo v prípade ak nie sú prevzaté rozvozárom. Online objednávky majú o niečo vyššiu trpezlivosť, než osobné.


generátor objednávok
Exp(10 min)
OSOBNÁ OBJ.
Exp(45 min)
Exp(17 min)
Exp(3 min)
40%
objednávka sa prijíma
Exp(1 min)
čaká sa na kuchára
P = 1
spracovanie osobnej obj.
doručenie zákazníkovi
5
kuchár
ONLINE OBJ.
60%
objednávka sa prijíma
Exp(1 min)
čaká sa na kuchára
spracovanie online obj.
Exp(17 min)
obj, čaká na baliacu linku
Exp(75 min)
baliaca linka
obj. čaká na rozvozára
Exp(60 min)
rozvozár
5
objednávka sa balí
doručenie zákazníkovi
Exp(15 min)
objednávka na rozvoze
Exp(2 min)

3 Simulačný model

3.1 Prevod na simulačný model

Prevod na simulačný model [1] (s. 44) bol vykonaný pomocou najnovšej verzie SIMLIB-u [2] v jazyku C++. Simulačný model je obsiahnutý v súbore ims.cpp.

Program sa dá spustiť :

- v základnom režime (so zistenými hodnotami) ako : `make run`
- s nastavením počtu kuchárov : napr. `./ims -k 8`
- s nastavením počtu rozvozárov : napr. `./ims -r 9`
- s nastavením doby prípravy jedla : napr. `./ims -j 15`
- s nastavením doby dopravy jedla : napr. `./ims -d 20`
- s nastavením časového rozostupu objednávok : napr. `./ims -a 10`

Parametre sa dajú ľubovoľne kombinovať.

3.2 Popis implementovaného modelu

Model obsahuje dva procesy: `OsobnaObjednavka` a `OnlineObjednavka`. Kuchári a rozvozári (čiže pracovníci) sú navrhnutí cez tzv. `Store()`, pričom kapacita každého Store() predstavuje počet pracovníkov a je možné ju meniť cez parametre programu. Obslužná baliaca linka je implementovaná ako `Facility BaliacaLinka`.

Generovanie prebieha v jednom generátori a pravdepodobnostné rozdelenie je implementované pomocou funkcie `Random()`, ktorá generuje pseudonáhodné číslo v intervale 0-1 a na základe hodnoty sa vygeneruje proces osobnej alebo proces online objednávky.

Metóda `Behavior()` v obidvoch procesoch implementuje ich chovanie, teda činnosti, ktoré vykonávajú. Využívajú sa funkcie `Enter()` a `Leave()`, ktoré popisujú zaberanie a uvoľňovanie kuchárov alebo rozvozárov. Pre obslužnú linku sa za tým istým účelom využívajú metódy `Seize()` a `Release()`.

Trieda `Timeout()` implementuje časovač čakania procesov v troch miestach systému - pri čakaní na kuchára pre obidva druhy procesov a pri čakaní na rozvozára pre proces online obj. Pri dosiahnutí vopred definovaných časov sa proces odstráni, t.j. opustí systém.

Všetky časové údaje sú obsiahnuté v definovaných premenných a sú implementované s exponenciálnym [1] (s. 91) rozdelením. Tieto časové údaje sú využívané v metóde `Wait()`, ktorá je použitá na označenie doby vykonávania určitej činnosti, alebo pri časovom rozostupe generovaných procesov v modeli.

4 Simulácie

4.1 Popis experimentov

Pred návrhom experimentov sme simulovali chod systému so skutočnými získanými údajmi a sledovali sme jednotlivé prvky systému a ich chovanie. Medzi sledované údaje patrí hlavne vyťaženie pracovníkov, t.j. ako stíhajú spracovávať objednávky v jednotlivých miestach systému, vyťaženie baliacej linky a počet nespokojných zákazníkov.

4.2 Motivácia experimentov

V realite je skúmaný systém funkčný, ale vzhľadom na momentálnu ekonomickú situáciu sa reštaurácia snaží ušetriť finančné prostriedky vo všetkých bodoch systému. Vlastnosti ako cena jedál, alebo výplata pracovníkov v tomto modeli skúmať nebudeme, ale je možné pracovať so zmenou počtu zamestnancov.

Vzhľadom na to, že znižovanie počtu zamestnancov na úplné prijateľné minimum je bežná praktika vo viacerých firmách, tak môžme zistiť, či je v skúmanom systéme možné vykonávať takéto zmeny (a v akej miere).

4.3 Postupy experimentov

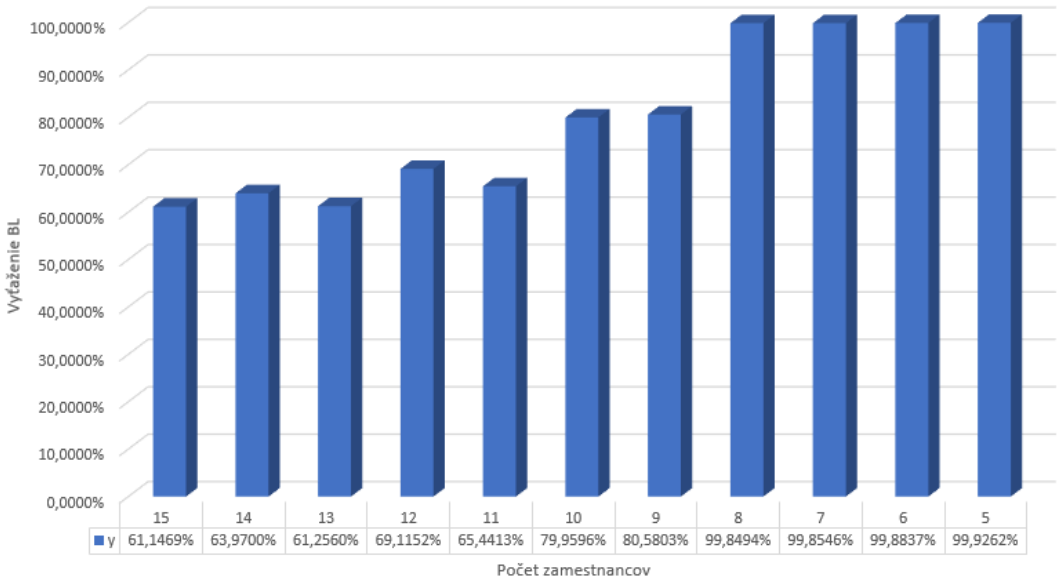
4.3.1 Experiment 1

Príkaz: `./ims -k K` , pričom $K \in < 3, 5 >, K \in Z$

V prvom experimente upravujeme hodnotu počtu kuchárov a to postupným znižovaním z reálnej hodnoty. Sledujeme hlavne čakanie objednávok na prevzatie ľubovoľným kuchárom.

Zisťujeme, že.

- .
- .



Obr. 1: Experiment 1

Vzhľadom na získané údaje sme zistili, že znižovaním počtu kuchárov sa predlžuje čakanie na objednávku, ale pokiaľ je reštaurácia ochotná zvýšiť čakanie, aj napriek možnej nespokojnosti zákazníkov, tak je možné znížiť počet na menšiu hodnotu, pričom hodnoty čakania sú uvedené v grafe 1

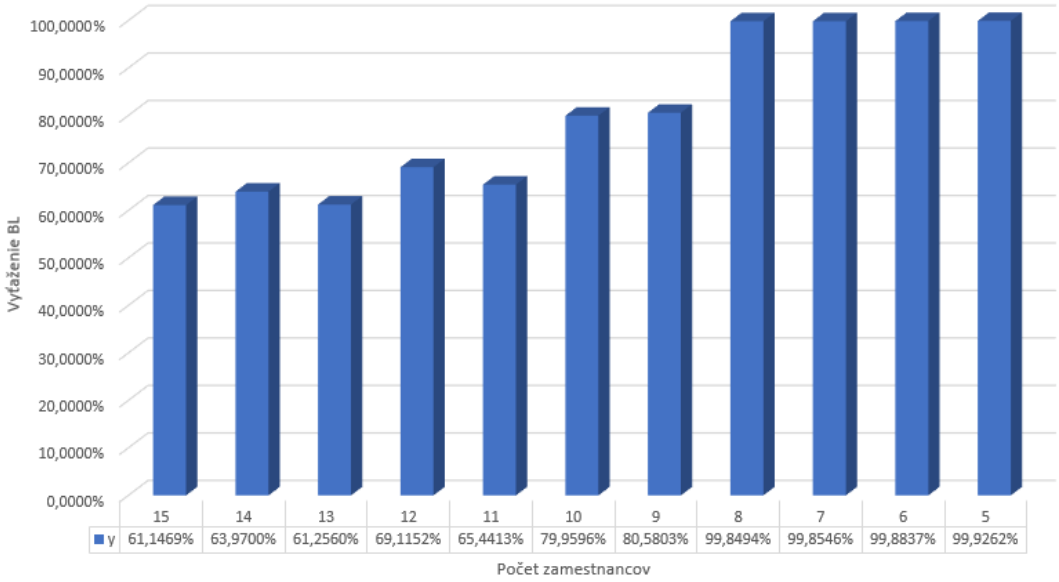
4.3.2 Experiment 2

Príkaz: `./ims -r R` , pričom $R \in < 3, 5 >, R \in Z$

V druhom experimente upravujeme hodnotu počtu rozvozárov a to postupným znižovaním z reálnej hodnoty. Sledujeme čakanie pripravených objednávok na rozvoz a následnú dopravu k zákazníkovi.

Zisťujeme, že.

- .
- .



Obr. 2: Experiment 2

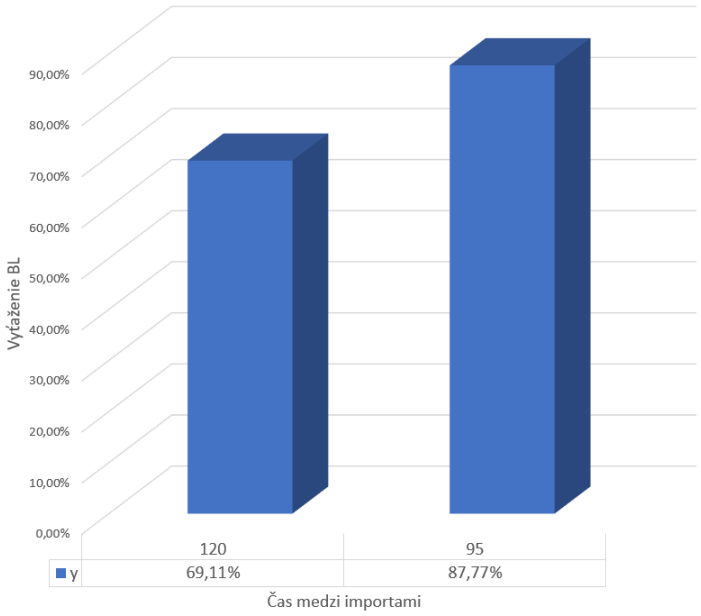
Vzhľadom na získané údaje sme zistili, že aj znižovaním počtu rozvozárov sa predlžuje čakanie na objednávku. Avšak znovu je možné tento počet znížiť na nižšiu hodnotu, čo však znamená znížený štandard služieb pre zákazníka.

4.3.3 Experiment 3

Príkaz `./ims -f F` , pričom $F = 8$

V treťom experimente testujeme chovanie systému počas kritických období, napr. začiatok zimného obdobia, kedy je záťaž systému o niečo vyššia.

Zvýšenie záťaže sa dá jednoducho napodobniť zvýšením frekvencie generovaných objednávok. Pre experiment znížime časový rozostup generovaných objednávok na 8 min.



Obr. 3: Experiment 3

todo

5 Záver

Literatúra

[1] Peringer P., Hrubý M., *Modelování a simulace*, 2021
http://www.fit.vutbr.cz/study/courses/IMS/public/prednasky/IMS.pdf

[2] Peringer P., *SIMulation LIBrary for C++*, 1991
http://www.fit.vutbr.cz/~peringer/SIMLIB/.cs