

ZÁKLADNÍ METODIKA SIMULAČNÍ STUDIE PŘI VYUŽITÍ PARALELNÍ DISKRÉTNÍ SIMULACE

Ing. Zdeněk Ulrych, Ph.D.

Ing. Pavel Raška

Ing. Petr Hořejší

Kateřina Candrová

ZČU v Plzni – Fakulta strojní - Katedra průmyslového inženýrství a managementu

1 Úvod

Nástup informačního věku v posledních dvou desetiletích 20. století překonal mnohá dosavadní tvrzení či dosud obecně uznávané teorie. V současné informační společnosti nelze konkurenceschopnost podniků spatřovat pouze ve schopnosti co nejrychleji zavést nové technologie či bezchybně spravovat a řídit finanční toky či majetek. Informační společnost vyžaduje k dosažení úspěchu pro výrobní i služby poskytující firmy nové schopnosti. Přesněji schopnosti zužitkovat, správně využívat a řídit neviditelné zdroje – tedy informace, které umožňují správně implementovat informační technologie či celé systémy.

Specifickou oblastí moderních informačních technologií je simulace, tedy napodobování či reprezentace chování celku jiným systémem, s nímž se stále více setkáváme jak v běžném životě tak také v oblastech výrobních, logistických či transformačních procesů.

Mezi nejrozšířenější způsob praktického využití interaktivních simulačních modelů (vymezené úseky zkoumané reality) patří analytické aplikace a programy určené pro podporu rozhodování, dále programy rozvíjející oblast výzkumu/vývoje prototypů, aplikace usnadňující prognózy počasí či nejrůznější tréninkové a zábavní simulátory. Většina simulačních modelů, které se v těchto oblastech využívají, je velice rozsáhlá a komplexní, a proto je kladen stále větší důraz na rychlost jejich výpočtu i flexibilitu vstupujících parametrů. A právě rychlost simulačních výpočtů je možné významně zvyšovat a to díky paralelní a distribuované simulaci, kdy je do řešení jednoho úkolu zapojeno více procesorů a každý „zodpovídá“ za část simulace. Tyto dvě výpočetní techniky využívají multiprocesorové a distribuované výpočetní platformy a lze je úspěšně uplatnit nejen při výpočtu velmi náročných řešení, ale také při tvorbě virtuálních prostředí či řízení rozsáhlých výrobních modelů.

2 Cíl projektu

Katedra průmyslového inženýrství a managementu se v poslední době zaměřuje na paralelní simulaci. Jedná se především o provádění experimentů v počítačových laboratořích. V tomto příspěvku je popsáno možné řešení projektu, který je zaměřen na tvorbu, vývoj, implementaci a ověření navrhované metodiky pro oblast paralelní simulace. Metodika navržená v rámci projektu byla ověřena na jednoduchém simulačním modelu.

V další navazující etapě se bude analyzovat možné využití některých optimalizačních metod za účelem zefektivnit nastavení hodnot vstupních parametrů modelu. Tyto metody budou případně modifikovány pro účel paralelní simulace. Je také zapotřebí vyvinout aplikaci umožňující automatické řízení experimentů na paralelních simulačních modelech, které budou spouštěny na jednotlivých počítačích v rámci sítě. V navržené SW aplikaci bude využita architektura *klient-server*, pomocí níž je možné spouštět a vyhodnocovat běh simulačních modelů z jednoho počítače, který řídí simulace na okolních počítačích.

3 Paralelní a distribuovaná simulace

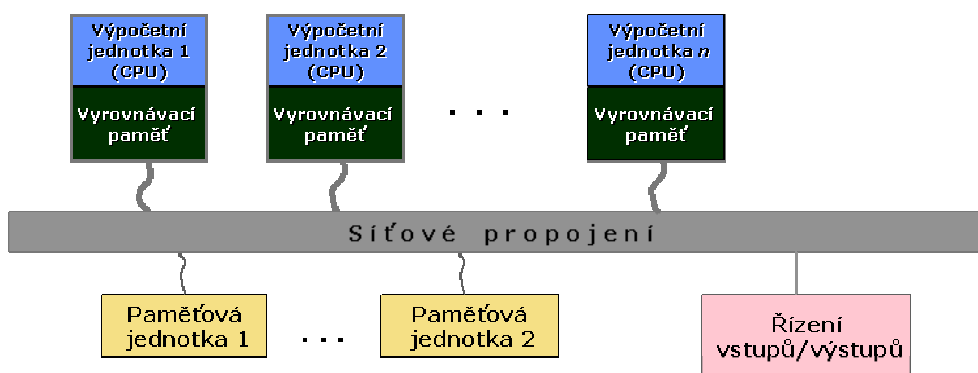
Počítačová simulace představuje výpočet, který modeluje chování reálného či abstraktního systému. Paralelní a distribuovaná simulace umožňuje spuštění tohoto simulačního programu na více počítačových procesorech, které jsou vzájemně síťově propojeny. Rozdíl mezi těmito dvěma přístupy je založen na formě výpočetního systému, na němž simulace běží. Základní charakteristiky obou technologií jsou uvedeny v následující tabulce:

	Paralelní simulace	Distribuovaná simulace
Fyzická poloha	Počítačová laboratoř	Budova, město, svět
Procesory	Homogenní	Většinou heterogenní
Komunikační síť	LAN	LAN/WAN
Komunikační latence	Několik (max. desítky) mikrosekund	Stovky mikrosekund, sekunda

Tabulka 1 Základní charakteristiky paralelní/distribuované simulace [1]

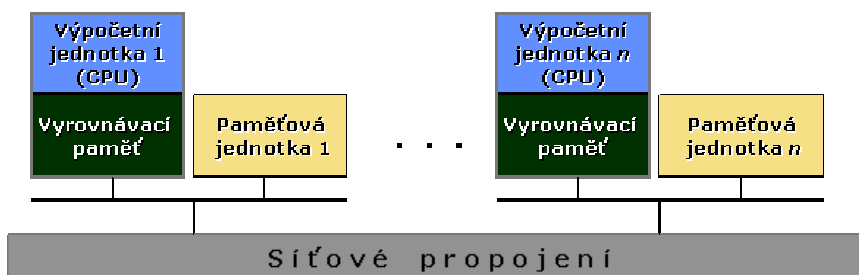
3.1 Podstata a význam paralelní/distribuované simulace

Principem paralelní diskrétní simulace je realizace výpočtu a běhu simulačních programů na bázi multiprocessorových výpočetních platform. Simulace probíhá na skupině počítačů jednoho výpočetního centra. Schematicky je podstata paralelní simulace znázorněna na následujícím Obrázek 1.



Obrázek 1 Schematické uspořádání komponent paralelní simulace

Oproti tomu podstata distribuované simulace představuje výpočet prováděný na geograficky vzdálených různě rozmístěných počítačích (výpočetní střediska mohou být rozmístěna po celém světě). Ty jsou vzájemně propojeny prostřednictvím lokální či světové sítě, viz Obrázek 2 Schematické uspořádání komponent distribuované simulace.



Obrázek 2 Schematické uspořádání komponent distribuované simulace

V obou případech je výpočet jednotlivých simulačních modelů (třeba tvořených z několika simulačních programů či podprogramů) distribuován na několik procesorů. Existuje mnoho důvodů proč využít distribuce simulačního výpočtu a jeho spuštění na více počítačových systémech a to např.:

- **Zkrácení výpočetních časů** – rozčleněním rozsáhlých simulačních modelů do množství subvýpočtů, které jsou řešeny souběžně, je možné zredukovat čas výpočtu úměrně počtu využitých procesorů.
- **Geografické rozmístění** – spuštění simulačních programů na množině zeměpisně zcela odlišně umístěných počítačů, umožňuje vytvářet jeden virtuální svět za participace mnoha účastníků (ti mohou být také fyzicky rozmístěny po celém světě). Tento způsob uspořádání výpočetního systému snižuje cestovní náklady, časové prodlevy, usnadňuje komunikaci a v konečném důsledku značně zvyšuje efektivitu.
- **Integrace modelů, jež jsou počítány na strojích různých výrobců** – lze uvést jednoduchý příklad praktického využití: letecké simulátory pro různé typy letadel (či pro jednotlivá letecká odvětví) byly vyvíjeny zcela odděleně různými výrobci. Mnohem jednodušší a nákladově efektivnější způsob než složitý „porting“ těchto programů na jeden počítač („násilné svázání jednotlivých simulátorů dohromady“), je tvorba nového virtuálního prostředí, ve kterém je každý výpočet prováděn na jiném počítači (ty jsou však virtuálně propojeny) s možností sjednotit komunikační architekturu.
- **Tolerance k chybám** – dalším významným přínosem multiprocesorových výpočetních technik je rostoucí odolnost vůči chybám. Pokud totiž v průběhu simulace jeden procesor „vypadne“, ostatní mohou dále pokračovat ve výpočtu, aniž by byl přerušen simulační běh. Výpadek jednoho stroje tedy neznamená výpadek celé simulace.[2] Je však nutné posoudit, zda tento výpadek nezpůsobí zkreslení výpočtu.

4 Volba softwaru, simulačního přístupu

Jak již bylo popsáno v úvodu, simulační výpočty rozsáhlých modelů jsou velice časově náročné. V našem případě záměrně nevytváříme vysoce složitý testovaný model, přesto je pro budoucí experimenty (zejména s ohledem na jejich množství a možnost dodefinování některých parametrů - *počet pracovišť, ekonomických parametrů*) žádoucí minimální hodnota výpočetních časů.

Před realizací projektu a při volbě simulačních nástrojů je nutné brát v úvahu softwarové a hardwarové zázemí místa realizace – tedy katedry Průmyslového inženýrství a managementu (která disponuje licencemi softwarového produktu *ARENA*), dále charakter modelu (jeho komplexnost, složitost, charakteristika vstupů a výstupů, možnost modifikace jednotlivých parametrů, výpočetní rychlost) a samozřejmě zkušenosti z oblasti řešené problematiky.

Po analýze předchozích faktorů byl jako nejvhodnější simulační přístup zvolen **paralelní simulační výpočet**. Tento přístup je založen na multiprocesorové výpočetní platformě, kdy jednotlivé procesory spolu vzájemně komunikují prostřednictvím místní sítě. Ve většině případů vzájemně homogenní procesory jsou umístěny v jedné počítačové učebně (tzn. nejedná se o distribuovaný, geograficky odlišně realizovaný, výpočet). Výpočetní časy se pohybují v mikrosekundách a úměrně se snižují s počtem využitých procesorů.

Pro tvorbu simulačního modelu byl zvolen velice komplexní softwarový nástroj *COTS* (*Commercial Off The Shelf*) produkt *ARENA 9.0* firmy Rockwell software, jehož licence je ve vlastnictví katedry Průmyslového inženýrství a managementu (ve 20 kopiích). *COTS* lze definovat jako oblast simulačních programů, které jsou převážně zaměřeny na entity, procesy

a toky. *ARENA* je nástroj, pomocí něhož je možné velmi rychle a elegantně provádět laboratorní simulace. Je svým konceptem přizpůsobena také potřebám průmyslového inženýrství a lze ji tedy využít pro modelování případů z této oblasti. Za cenu o trochu nižší univerzálnosti se k nám tedy dostává intuitivní, elegantní a rychlé řešení.

Software *ARENA* velice spolehlivě „komunikuje“ s nástrojem *Visual Basic*, který umožňuje realizaci velice elegantního řešení. Samotné zpracování výstupů (tedy výsledných přehledů, tabulek a statistik – dále jen výsledků) pak usnadňuje jejich export a zobrazení v databázovém programu *Access*. Uživatelé se tak nabízejí možnost rychlého přístupu a systematického přehledu usnadňující vyhodnocení výsledků experimentu.

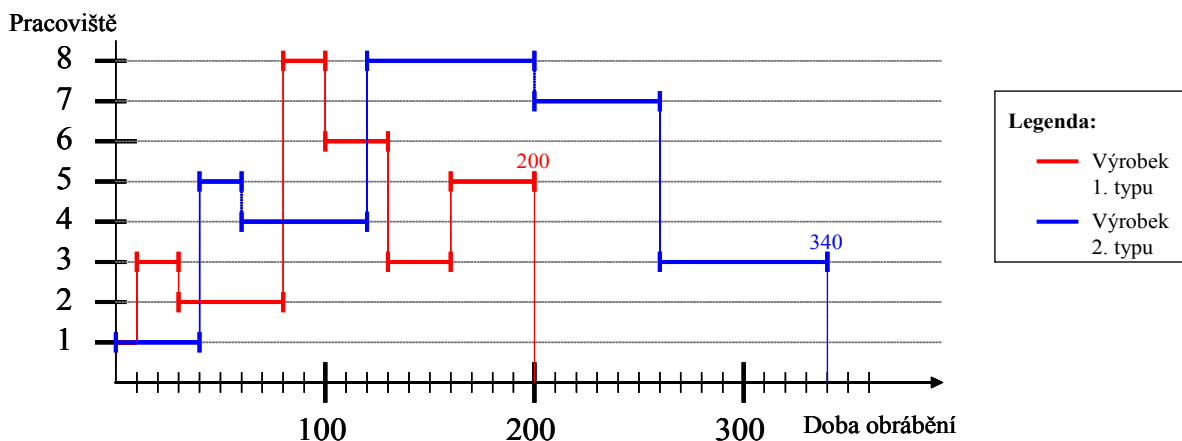
Popis tvorby modelu a jeho jednotlivých komponent je blíže uveden v kapitole Tvorba simulačního modelu.

Jelikož je cílem automatizace procesu paralelní simulace, bude vhodné parametry pro modely zadávat a modifikovat na počítači klient a vlastní simulaci provádět na serveru. Je tedy zapotřebí vytvořit aplikaci, která umožňuje provádět všechny předem vyjmenované podmínky.

4.1 Metodika pro tvorbu a řízení rozsáhlých simulačních modelů

Po vytýčení hlavního cíle (tvorba a ověření metodiky při vytváření a využití paralelní simulace) lze projekt rozvrhnout do několika dílčích cílů, které mají za úkol:

- a) prostřednictvím zvoleného softwarového nástroje *ARENA* vytvořit jednoduchý model výrobní dílny. Tato dílna je tvořena 8 pracovišti. Každé pracoviště disponuje různým počtem zdrojů, nejvíce však 10. Vyrábí se 2 odlišné komponenty - 100 kusů výrobku typu 1 a 200 kusů výrobku typu 2. Oba typy výrobků se kromě intervalu příchodu do výroby, technologického postupu a celkového počtu kusů mohou také lišit prioritami, které jim jsou přiděleny pro čekání ve frontách;



Graf 1 Sekvence a doby obrábění obou typů výrobků

- b) tvorba platformy, která umožňuje paralelizaci výpočtu na 15 počítačích;
- c) vytvoření vhodné logické vrstvy a uživatelského rozhraní pro modifikaci vstupních parametrů;
- d) vytvoření vhodné logické vrstvy a uživatelského rozhraní pro zpracování a prohlížení výsledků;
- e) provedení řady experimentů nad takovýmto modelem, jejich vyhodnocení a zobrazení kritériální funkce.

Realizace samotného projektu je tedy rozdělena do několika částí. Popis realizace projektu je rozebrán v následujících kapitolách. V následujícím textu je navržena metodika pro tvorbu a

řízení komplexních simulačních modelů, která sestává ze sekvence činností. Klíčovým parametrem metodiky, který významně ovlivňuje její celkovou podobu, charakter a způsob realizace jednotlivých kroků, je způsob a princip vzájemného propojení počítačů provádějících simulační výpočet. Metodika byla zpracována a prakticky je ověřena právě pro paralelní simulaci. Navržená metodika tvorby a řízení simulačních modelů představuje sled činností, z nichž každá je charakterizována řadou parametrů. Sled těchto činností v logické i faktické návaznosti je ve stručné a přehledné formě uveden v následující Tabulka 2.

Činnost		Požadavky na softwarový nástroj		Použitý SW nástroj
1.	Vytvoření modelu	1.	Zrychlení a zjednodušení procesu tvorby modelu	ARENA 9.0
		2.	Detailnost tvorby modelu, komplexnost	
		3.	Formát vstupních dat	
		4.	Úroveň animace	
		5.	Celkový výstup (reporting výsledků, export)	
		6.	Automatický běh	
2.	Paralelizace modelu	1.	Výpočetní rychlost	VISUAL BASIC 2005
		2.	Rozsah paralelizace výpočtu	
		3.	Pasport pro vstup dat	
		4.	Zabezpečení dat (vyžádání dat)	
		5.	Jednoduchá konfigurace souboru modelu	
		6.	Forma komunikace mezi moduly	
		7.	Rozsah funkcionalit	
3.	Uživatelské rozhraní - vstup	1.	Možnost konfigurace vstupních parametrů na úrovni GUI	MS ACCESS, VISUAL BASIC 2005
		2.	Možnost nalezení okolních počítačů	
4.	Uživatelské rozhraní - výstup	1.	Přehlednost, uživatelská příjemnost, možnost grafického zobrazení apod.	MS ACCESS, VISUAL BASIC 2005
5.	Nástroj vyhodnocení výsledků	1.	Rozsah funkcionalit	MS EXCEL, MATLAB
		2.	Požadavky na formu a formát výstupních dat	

Tabulka 2 Posloupnost činností navržené metodiky

Jednotlivé kroky metodiky jsou detailně popsány v [5]. Popišme si je v následujících podkapitolách jen ve stručnosti.

4.2 Tvorba simulačního modelu

Cílem praktické části projektu je tvorba modelu, nad kterým budou prováděny experimenty, dále příslušných rozhraní (vstupní parametry a výstupní přehledy) a také nástroje, který bude podporovat paralelní simulaci.

4.2.1 Vstupní parametry modelu – analýza

Data, která představují vstupní hodnoty simulačního výpočtu, jsou do modelu importována z databáze ve formátu *.mdb.

Tyto vstupní parametry lze rozdělit do dvou základních skupin:

1. Povinné podmínky:
 - generovány dva typy výrobků - deterministický interval;

- jednotlivé výrobky mají definovány vzájemně odlišné technologické postupy;
- 2. Ovlivnitelné (vnitřní) podmínky, kterými jsou:
 - Investice do zdrojů - zvětšování kapacity zdrojů až do maximální hodnoty;
 - Organizace práce – priority;
- 3. Neovlivnitelné (vnější) podmínky:
 - Dominance výrobku ve výrobní náplni – čistá doba výroby výrobku. Tu však v reálném systému nelze nikdy dosáhnout (v modelu nejsou uvažovány časy mezioperační manipulace, manipulace s výrobkem na pracovišti atd.), proto jsme stanovili minimální upravený čistý čas $T_{i\max} = T_i * k_i$, kde k_i konstanta zohledňující popsané prodlevy reálného systému;
 - Dodací podmínky – ty jsou pevně stanoveny odběratelem. Jejich charakter je vyjádřen průběhem kritériální funkce, která přesně stanoví výši penalizace při jejich nedodržení.

4.3 Vytvoření softwarové pomůcky pro podporu paralelní v prostředí ARENA

V následující kapitole je rámcově popsán navržený nástroj, který na úrovni paralelní simulace umí spolupracovat s produktem *ARENA* a analyzovat výsledky simulací.

Prostředí programu *ARENA* přímo podporuje aplikace psané v architektuře *.NET* a současně je jejím „mateřským“ programovacím jazykem *Visual Basic*. Pro vývoj aplikace tedy použijeme právě *Visual Basic*.

V případě tohoto programu je zřejmé, že se bude jednat o síťovou aplikaci. Podívejme se nejprve na teorii (kapitola *ARENA Interop* 4.3.1).

4.3.1 ARENA Interop

ARENA zahrnuje typovou knihovnu (*Type Library*), která obsahuje objekty, jejich vlastnosti, metody, události a konstanty, které můžeme externě využívat jako vstup do projektu v např. *Microsoft Visual Basic (VB)*. Pro snazší tvorbu ve *VB* využijeme produkt *Visual Studio 2005*, kde v rámci GUI přímo nastavíme odkaz na typovou knihovnu.

Podobně bychom tak mohli učinit i v případě jiných programovacích jazyků a vývojových prostředí (např. *Delphi 2005* a vyšší atp.). Z těchto vývojových prostředí je možné i vidět přímo do objektů, které využívá *ARENA*. Pro náš případ jsme zvolili prostředí *Visual Basic*, což je současně jazyk, který *ARENA* přímo podporuje. Po prvním spuštění *ARENY* je typová knihovna automaticky registrována, tudíž je možné ji přímo vidět ve výběru (podobně lze přistupovat i např. k celé sadě *MS Office*).

Stejně jako je tomu například analogicky u aplikace *MS Excel*, můžeme k *ARENĚ* externě přistupovat přes standardní objekt, který se jmenuje *Application*. Pokud bychom nespécifikovali jinak a vznikl konflikt mezi jednotlivými objekty *Application*, pak *VB* vybere knihovnu s vyšší prioritou. Priority lze v rámci vývojového prostředí měnit. Pokud bychom nechtěli měnit prioritu, můžeme deklarovat instanci *Application* přímo s celou cestou *Arena.Application*, tedy:

```
Dim arenaObj As Arena.Application
```

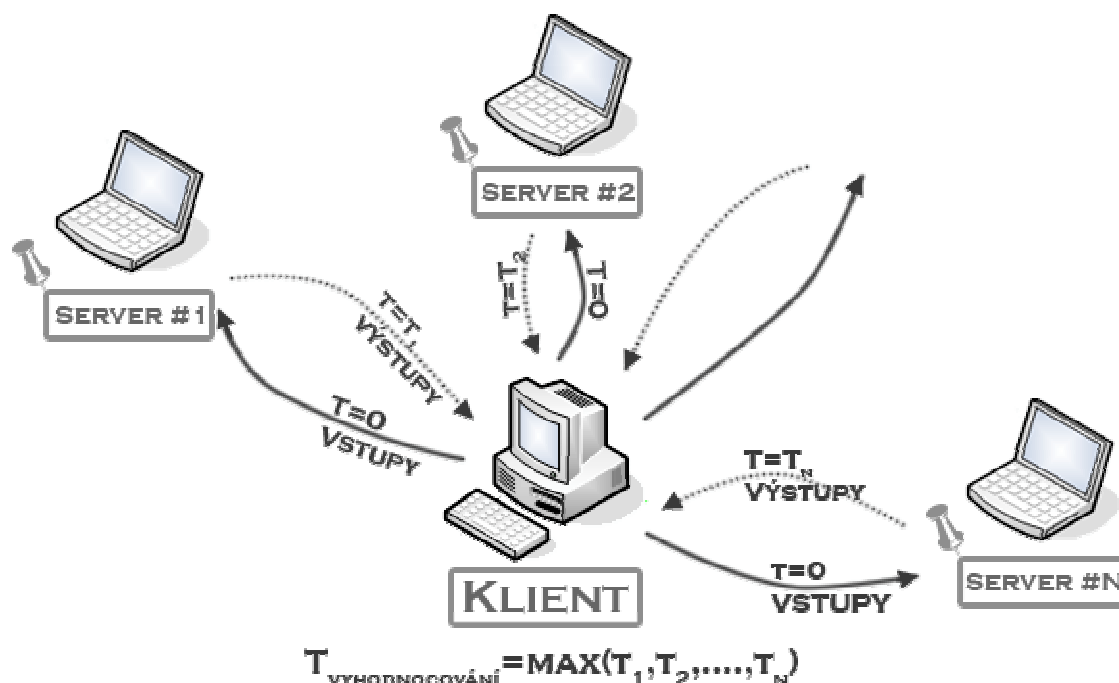
Pro přístup k samotnému modelu je zde možnost deklarovat proměnnou typu model:

```
Dim arenaModel As Arena.Model
```

Po překladu programu, který využívá takové objekty knihovny typů programu *ARENA*, se k přeloženému *.exe* přidá ještě knihovna *Interop.Arena.dll*. [4]

4.4 Architektura klient-server

Úkolem je vytvořit prostředí s architekturou klient-server, které bude schopné inteligentně kontrolovat přenosy simulačních modelů na vzdálené počítače, zajistí zpětnou vazbu s transferem výsledků proběhlé simulace na vzdálených počítačích a posléze vyhodnotí výsledky, dle nichž pak vytvoří nové parametry modelů, které budou opět zaslány na vzdálený počítač (viz Obrázek 3 Zadání programu pro kontrolu paralelní simulace pomocí programu ARENA).



Obrázek 3 Zadání programu pro kontrolu paralelní simulace pomocí programu ARENA

Zatím se zaměříme jen na první fázi, ve které se budeme snažit nakonfigurovat vstupní data tj. parametry modelů. Budeme hledat rozhraní pro komunikaci s programem *ARENA* na serveru a rozhraní pro komunikaci s uživatelem na straně klienta. Po úspěšném provedení simulačních experimentů na stranách všech serverů budou výsledná data stažena na správcovský počítač. Teď bude potřeba na základě těchto dat rozhodnout o vhodném způsobu optimalizace. Vlastní optimalizace bude pak předmětem fáze druhé.

Rozdělme si takové softwarové řešení na několik modulů.

- Tím prvním bude správa nastavení pro zasílání modelů. Bude se jednat o **vrstvu klienta** – ta bude zahrnovat možnost nastavit adresy vzdálených počítačů a konfigurace vstupních souborů a modelů;
- Dalším modulem by pak bylo softwarové řešení **serveru** – na každém počítači bude stejná implementace serveru – rozlišit bychom je mohli kupříkladu pomocí názvů počítačů či IP adres;
- Nyní se dostáváme k fázi, kdy předpokládáme, že jsme všechna data stáhli a máme tedy již k dispozici výstupní data. Nyní bude potřeba z těchto výsledků vytěžit ty, které potřebujeme, což bude úkolem třetího **vytěžovacího modulu**, který na základě parametrů, které budeme chtít sledovat, sestaví tabulku vstupních dat. Tabulka je sestavena na základě výsledků všech simulačních experimentů;

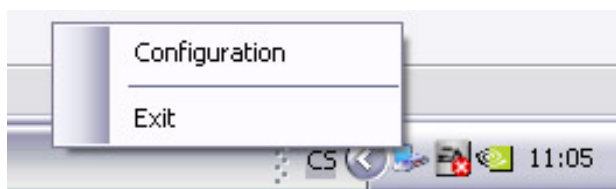
- Posledním modulem bude **vizualizace**, která na základě vstupních dat připraví grafy, na základě nichž se pak budeme moci rozhodnout jaký způsob optimalizace a kritériální funkce zvolit.

V další fázi, která však nebude předmětem následujících stránek, bude potřeba na základě průběhu grafů zvolit vhodný způsob optimalizace a pak zautomatizovat další vhodné nastavení vstupních parametrů.

V rámci produktu *ARENA* lze pracovat s *.mdb* soubory (DB *MS Access*), a to nejen pomocí skriptů, ale také je do tohoto formátu ukládána statistika simulace. [3]

4.4.1 GUI (Grafické uživatelské prostředí) Serveru

Při návrhu uživatelského prostředí bylo důležité vědět, že bychom velmi rádi, aby GUI serveru uživatele „obtěžoval“ co možná nejméně. Byla zvolena možnost vložit jej do *system-traye* jen jako ikonku. Když je ikonka viditelná, znamená to, že je server naživu. Na Obrázek 4 GUI Serveru vidíme ikonu a menu, které se objeví při kliknutí pravým tlačítkem. V konfiguraci lze nastavit port pro komunikaci.



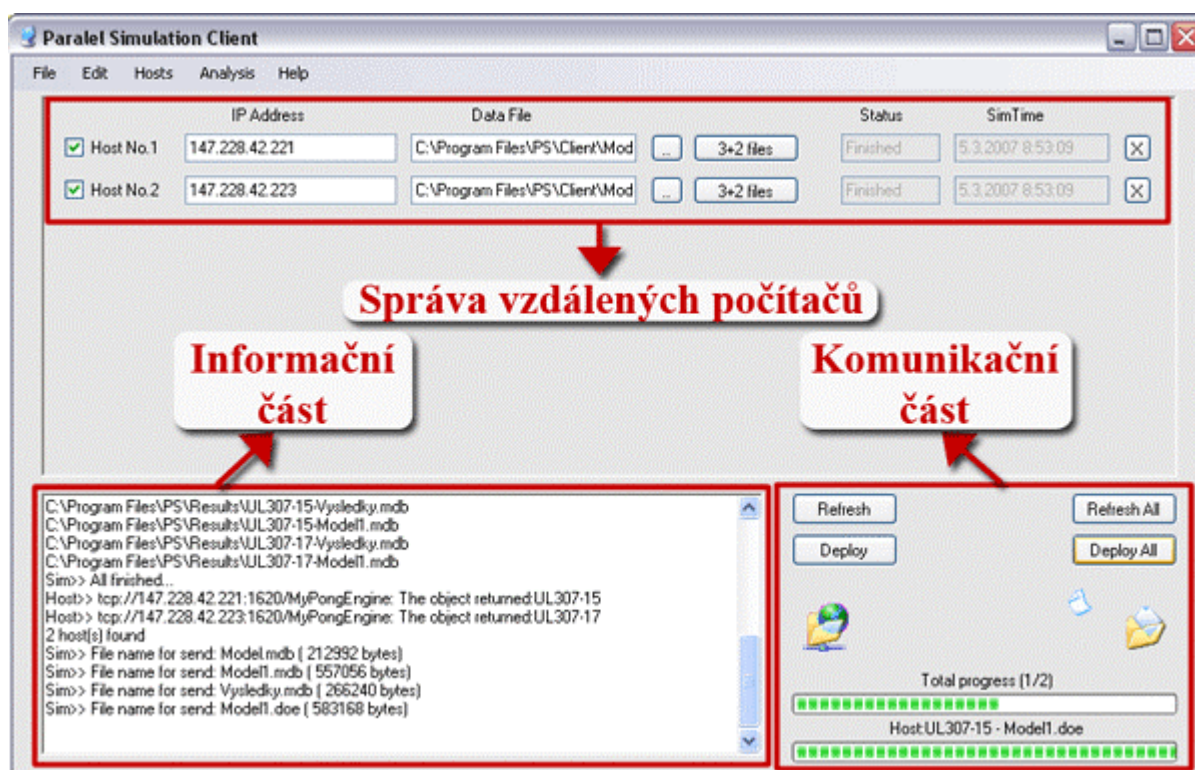
Obrázek 4 GUI Serveru

4.4.2 GUI Klienta

Na Obrázek 5 GUI Klienta vidíme obrazovku aplikace klienta. Pracovní plocha je rozdělena na tři části, tak jak je vidět na obrázku a řekněme si o každé z nich.

- Správa vzdálených počítačů - do této oblasti je možné přidat konfiguraci každého nového vzdáleného počítače včetně všech nastavení nutných pro úspěšný transfer a spuštění simulace. Můžeme také vidět v jakém je každý model stavu – běží, skončil, je pozastaven či nastala chyba;
- Informační část - v této části nalezneme všechny důležité informace o přenosech a komunikaci. Ty jsou vypisovány do listovacího boxu. Tato část obsahuje: informace o obnovení (tj. nalezení) vzdálených počítačů a služeb na nich, informace o přenosu a zpětném přenosu (stažení), informace o prováděných a ukončených simulacích, chybová hlášení;
- Komunikační část – můžeme zde nalézt tlačítko pro nalezení vzdálených služeb a také tlačítko pro zahájení přenosu „Deploy“ popř. „Deploy All“. Po kliknutí na toto tlačítko jsou opět nalezeny vzdálené počítače. Pokud na nich služby byly objeveny, můžeme začít s vlastním přenosem. Postupně budeme posílat na vzdálené počítače všechny relevantní soubory. Ihned máme možnost vidět, jak se nám daří soubory přenášet a kolik procent z celku je již přeneseno. Po dokončení práce vidíme krátkou zprávu o přenosech. Můžeme se dočíst, kde jsou uloženy výsledné soubory a jak dlouho trvaly simulace pro každý vzdálený počítač. Zprávu můžeme vytisknout, uložit či jen zavřít.
- Menu - v uživatelském menu můžeme najít několik položek:
 - **File** (Soubor) → Open XML, Add from XML, Save All, Save selected;
 - **Edit** (Editace) → Select All, Unselect All, Preference;
 - **Hosts** (Vzdálené počítače) → New Host, Remove Selected, Remove Selected Remove All;
 - **Analysis** (Analýza) → Data Mining, Visualisation, Queries Editor;

- **Help** (Nápověda) → HTML Help, About. [3]



Obrázek 5 GUI Klienta

5 Kriteriační funkce

Kriteriační funkce charakterizuje míru splnění či nesplnění podmínek dodání výrobků, v našem případě se zaměřuje na časové lhůty. Tato funkce je závislá na počtu zdrojů, prioritách přidělených jednotlivým výrobkům, intervalu vstupu výrobků do výroby, výrobních časech a množství požadovaných kusů.

V našem případě jsme průběh kriteriační funkce stanovili takto:

$$p = h_1 + (\tau - T_{\max}) \cdot k$$

...kde τ = skutečná doba výroby;

$T_{\max}$ = maximální požadovaná hodnota doby výroby;

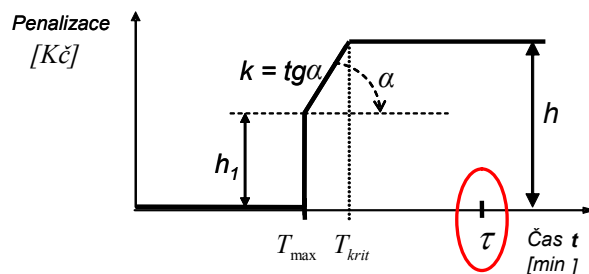
h_1 = skokový nárůst penalizace při překročení povolené maximální lhůty (tedy $T_{\max}$);

k = konstanta (charakterizující časový nárůst).

Je nutné poznamenat, že hodnota $T_{\max}$ neodpovídá čistému času výroby jednoho výrobku – tedy hodnotám 200 a 340 jednotek (viz graf 2.1). Tento čistý čas byl upraven koeficientem¹ $K=1,05$ tak, abychom se více přiblížili v praxi dosažitelným a ve skutečném výrobním systému realizovatelným hodnotám.

Průběh kriteriační funkce je zřejmý z následujícího obrázku:

¹ Tímto koeficientem se snažíme vyjádřit podmíněně nutné přestávky výrobního cyklu, tedy časy upínání a odepínání výrobku, apod.



Graf 2 Kriteriační funkce

V našem případě jsme jednotlivé parametry definovali takto:

- Hodnota úhlu α je rovna 45° , tedy nárůst penalizace má v úseku $\langle T_{\max}; T_{krit} \rangle$ lineární průběh charakteru přímé úměry (tzn. prodloužení maximální povolené doby výroby o 1 časovou jednotku = nárůst penalizace rovněž o 1 peněžní jednotku);
- Hodnota h představuje marži výrobce. Pokud výrobní čas dosáhne hodnoty T_{krit} a vyšší, pro výrobce přestává být daný výrobek ziskový (penalizace dosáhla výše veškeré marže);
- Hodnota h_1 svou výší odpovídá 50-60% marže (dle typu výrobku). [4]

6 Provedené experimenty, mapování kriteriální funkce

Nad vytvořeným modelem bylo provedeno prostřednictvím vytvořené aplikace middleware velké množství experimentů, které si kladly za cíl zjistit průběh a charakter kriteriální funkce charakterizující splnění dodavatelských podmínek. Samotná kriteriální funkce, její vlastnosti a explicitní vyjádření je popsáno v následující podkapitole.

6.1 Provedené experimenty

Po nastavení povinných parametrů modelu byly přiděleny priority oběma výrobkům tak, aby byly zcela vyrovnané, tzn. ve všech frontách před pracovišti se vyřízení požadavků řídí metodou FIFO. Poté bylo provedeno několik desítek experimentů tak, aby bylo určeno optimální nastavení systému – tedy takové nastavení, kdy se před jednotlivými pracovišti netvoří výrazné fronty (maximálně v počtu několika jednotek kusů) a čekací časy výrobků nepřekračují 10 minut. Zároveň je nutné výrobní zdroje vytížit tak, abychom jejich disponibilní kapacitu využili minimálně na 60% (nelze však uvažovat 98% a vyšší využití – v reálném systému je tato celková disponibilní kapacita snížena o časy oprav, podmíněčně nutných přestávek² či případných poruch). Takto nastavený systém, tedy systém, kdy se netvoří velké fronty před jednotlivými pracovišti a i zdroje jsou adekvátně vytíženy, byly testovány na zatížení při různých hodnotách priorit vstupujících výrobků.

Dále byla provedena velká množina experimentů s cílem zjistit průběh kriteriální funkce. Logický postup byl následující:

1. volba pracoviště, které bude označeno za kritické;
2. nastavení počtu zdrojů na kritickém pracovišti na hodnotu určenou jako vyvážený stav systému;
3. úprava vstupních parametrů: zadány priority vstupujících výrobků do výroby tak, aby prvních 10% výrobků typu 1 bylo obraceáno se stejnou prioritou jako výrobek 2, zbylých 90% výrobků typu 1 již však s vyšší prioritou;
4. snížení počtu vstupujících výrobků typu 1 s vyšší prioritou o 10% (tj. prvních 20% výrobků typu 1 bude obraceáno se stejnou prioritou jako výrobek 2, zbylých 80% výrobků typu 1 má přidělenou vyšší prioritou);

² Podmínečně nutnou přestávkou jsou manipulační časy při upínání a odepínání výrobku do stroje apod.

5. opakování předchozích dvou kroků až po dosažení konečné hodnoty 100% výrobku jedna vstupujících do systému se stejnou prioritou jako typ 2;
6. vyhodnocení výsledků, výpočet bodů grafu – tedy hodnot bodů, které určují průběh kritériální funkce.

Uvedený postup jsme opakovali pro počet zdrojů na kritickém pracovišti 3, 4 a 6. Volba těchto hodnot simuluje následující stavy systému:

- v případě 3 zdrojů na pracovišti předpokládáme poruchu dvou strojů kritického místa;
- v případě hodnoty 4 předpokládáme poruchu 1 stroje kritického místa;
- v případě hodnoty 6 předpokládáme přidání jednoho zdroje na kritické místo a tím odstranění jeho rizikovosti (viz teorie TOC).

Provedené experimenty byly vyhodnoceny a na základě výstupů byl vyšetřen průběh kritériální funkce. [4]

6.2 Dosažené výsledky

V souladu s popsáním logickým postupem v úvodu jsme v první fázi řešení hledali (prostřednictvím řady experimentů) tzv. **vyvážený stav systému**. Tedy takový stav, kdy se před jednotlivými pracovišti tvoří přijatelné fronty (tj. řádově jednotky minut) a naopak – utilizace (tedy vytížení) zdrojů není menší než 60% (výjimku představuje pouze jedno pracoviště).

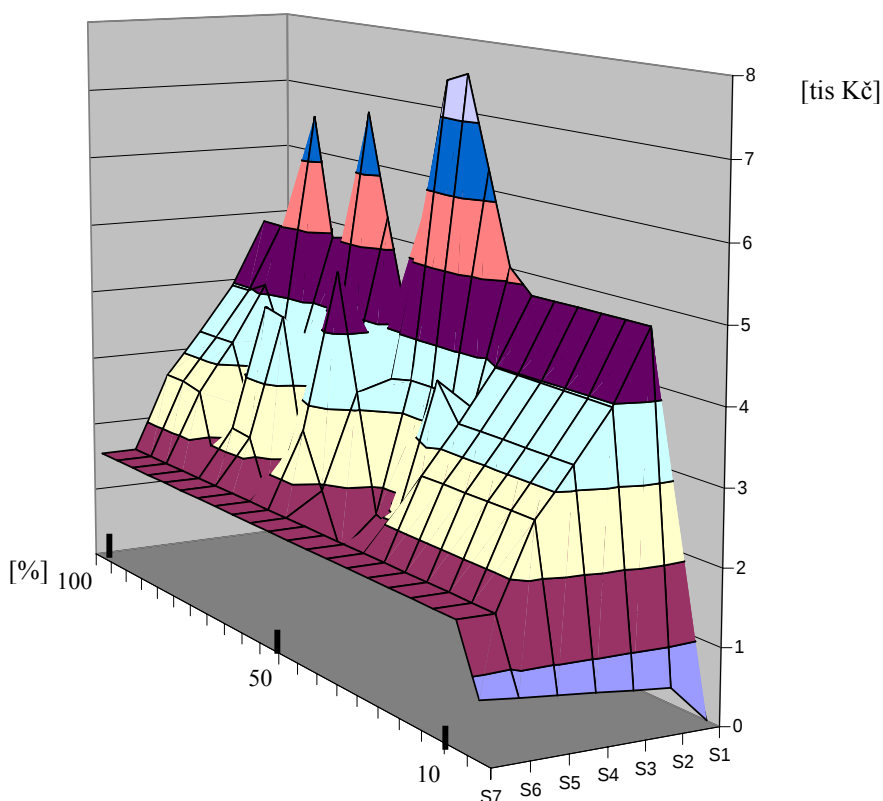
Hodnoty vytížení některých pracovišť neodpovídají počátečním předpokladům. U zdroje typu S6 je velmi nízká utilizace způsobena malým množstvím operací a krátkými časy obrábění vykonávanými na tomto pracovišti. Počet zdrojů je snížen na minimum, tedy jedna. Vyšší vytížení tohoto pracoviště (a tedy všech jeho zdrojů) není při současných vstupních parametrech možné.

Zdroje pracoviště 4 a 7, tedy S4 a S7, jsou vytíženy na velice vysokou míru a to přibližně z 95%. Tato hodnota byla zvolena tak vysoká z toho důvodu, že po přidání dalšího zdroje dochází ke skokovému snížení utilizace a to na hodnoty pohybující se v rozsahu 25 – 35%, což je pro reálný výrobní systém značně neefektivní využití kapacit. Proto byla ponechána tato vysoká míra utilizace (i za cenu tvorby velice malých front).

Kritériální funkce celého systému, která je závislá na počtu strojů kritického místa a poměru priorit, je pak znázorněn v 3D grafu a představuje průběh plnění dodacích podmínek včetně případných penalizací.

Tvar kritériální funkce odpovídá zjištěným hodnotám jednotlivých experimentů. Lze ji tedy interpretovat jako graf závislosti penalizace celého systému na počtu strojů kritického místa (osa x nabývá hodnoty S1 až S7, kde např. S2 značí počet zdrojů kritického pracoviště = 2) a procentu výrobků typu 1 se zvýšením priority v průběhu výroby (osa y; nabývá hodnoty 10 až 100%). Míru penalizace definuje osa z.

Pro potřeby tohoto experimentu je kritériální funkce vyjádřena pro pracoviště č. 3 (viz výše; časově nejvíce vytížené pracoviště) a jednotlivé priority výrobku typu jedna jsou měněny dle výše uvedeného postupu. Průběh vypočtené funkce je pak následující:



Graf 3 Průběh kritériální funkce

Na tomto místě je nutné poznamenat, že zvolený algoritmus pro hledání grafu není vhodný pro všechny průběhy funkce a cílem další práce je přidat možnosti generování dle dalších algoritmů (tj. křivek – parabol, kvadrik, atd.). Ze samotného Graf 3 Průběh kritériální funkce je patrné, že nejvyšší hodnoty penalizace je dosaženo pro systém s jedním výrobním strojem. Obecně můžeme tedy na základě grafu považovat výrobu s jedním, dvěma, třemi i čtyřmi zdroji na kritickém pracovišti za značně neefektivní. Od hodnoty 5 je již funkce penalizace konstantní a stává se rovinou plochou. Závěrem můžeme konstatovat, že nejvýhodnější je volba 5ti zdrojů kritického pracoviště 3 a s ohledem na míru penalizace i utilizace. V případě, že dojde k poruše jednoho zdroje a budeme nuceni vyrábět pouze se čtyřmi zdroji (tj. x-ová souřadnice S4), penalizace se zdvojnásobí. [4]

7 Závěr a další kroky

Realizace samotného projektu byla rozdělena do několika částí a to ve shodě s uvedenými cíly. Jednalo se především o ověření metodiky pro paralelní simulaci na jednoduchém simulačním modelu. Všechny dílčí cíle byly realizovány díky týmu a technickému zázemí Katedry průmyslového inženýrství a managementu. Pro potřeby experimentů (zejména pak s důrazem na jednoduchou modifikaci vstupních parametrů modelu) bylo navrženo uživatelsky příjemné rozhraní v programu *Visual Basic*. To umožňuje snadnou změnu disponibilních kapacit pracovišť, priorit výrobku, dobu výroby či množství vyrobených kusů a uživatel tak má možnost velice jednoduše a elegantně měnit vstupní parametry modelu. Jednoduché zpracování jednotlivých výstupů (tedy výsledků experimentu) umožňuje export dat do databáze *Access*. Uživatel tak získá přehledný výpis výrobních časů jednotlivých výrobků, údaje o délkách front před pracovišti, dobách čekání či utilizaci zdrojů. Odpovídající hodnoty jsou uživateli k dispozici v přehledných tabulkách, vždy v maximální, průměrné a

minimální hodnotě a lze je vizualizovat. Popsané výsledkové listy se generují a ukládají (na zvolené místo) zcela automaticky již v průběhu výpočtu modelu.

Jako další etapu předpokládáme navržení nadstavby nad řídicí aplikací, která bude automaticky nastavovat hodnoty vstupních parametrů modelu pomocí optimalizačních algoritmů. Nejprve je ale nutné stanovit vhodnost jednotlivých stochastických algoritmů včetně algoritmů genetických. Tyto algoritmy budou muset být nejprve analyzovány a poté implementovány, popřípadě modifikovány pro paralelní simulaci.

Použitá literatura

- [1] Tutorial CSCI 8220 Parallel and Distributed Simulation, Maria Hybinette, Barrow, 2003
- [2] Parallel and Distributed Simulation, Richard M. Fujimoto, Georgia Institute of Technology Atlanta, *Proceedings of the 1999 Winter Simulation Conference*, 1999
- [3] Rigorózní práce Diskrétní distribuované simulační systémy a aplikace High Level Architecture ve strojírenství, Petr Hořejší, Západočeská univerzita v Plzni, 2005
- [4] Interní grant Výzkum metod optimálního řízení distribuovaných simulačních modelů výrobních procesů a jejich integrace do konceptu digitálního podniku - Experimenty s modelem virtuální dílny, Petr Hořejší, Kateřina Candrová, Západočeská univerzita v Plzni/FST/KPV, Plzeň, 2006
- [5] Diplomová práce Tvorba a řízení rozsáhlých simulačních modelů výrobních systémů, Kateřina Candrová, Západočeská univerzita v Plzni, 2007

Kontaktní údaje

Ing. Zdeněk Ulrych, Ph.D.
Západočeská univerzita v Plzni, Fakulta strojní
Univerzitní 8, 306 14 Plzeň
Tel: 377638406
email: ulrychz@kpv.zcu.cz

Ing. Pavel Raška
Západočeská univerzita v Plzni, Fakulta strojní
Univerzitní 8, 306 14 Plzeň
Tel: 377638480
email: foxpraska@seznam.cz

Ing. Petr Hořejší
Západočeská univerzita v Plzni, Fakulta strojní
Univerzitní 8, 306 14 Plzeň
Tel: 377638480
email: tucnak@cbox.cz

Kateřina Candrová
Západočeská univerzita v Plzni, Fakulta strojní
Univerzitní 8, 306 14 Plzeň
email: candrova@post.cz