

Структура и изисквания към проекта по конкурентно програмиране

ВАСИЛ ГЕОРГИЕВ

 v.georgiev@fmi.uni-sofia.bg

Фази на софтуерния проект

Фаза Анализ

- Функционален анализ
 - избор на **образци** с аналогична или сходна функционалност
 - ДЪЛЪГ СПИСЪК** *Списък източници – атрибути на източниците:*
 - КЪС СПИСЪК**
 - [Номер в списъка] Автор, Заглавие на документа, Издател, Дата, (Интернет адрес)
 - примери:* [7] Система за обслужване на Теленор България, (<http://www.telenor.bg/bg>) ;
 - [8] Free Responsive Mobile Website Templates, Agile Information Technologies (<https://w3layouts.com/>);
 - [9] Stefano Meschiarì, Colliding N-body spheres: Particle Mayhem! github.io, (<http://...>)
 - опция** (при по сложни проекти): **интервюта** с [бъдещите] потребители на системата по роли – примерен разгърнат списък потребителски роли включва: системни администратори, администратори на данните, регистрирани потребители, нерегистрирани потребители (гости)
 - анализът на избраните образци завършва със **Диаграма на случаите на употреба** (UML) и/ли със сравнителна **таблица** от вида :

Образец	Черти:	Функция 1	Функция 2	...	Функция n	Коментар
[n] или име на образец						
[m] или име на образец						
...						
Име на проектираното приложение						

Фази на софтуерния проект

- ... ФАЗА АНАЛИЗ
 - Нефункционален анализ
 - **ТЕХНОЛОГИИ** — НАПРИМЕР ИЗПОЛЗВАНИ ЕЗИЦИ, ПЛАТФОРМИ, БИБЛИОТЕКИ И ДРУГИ СРЕДСТВА ПРИ ИЗБРАНИТЕ ОБРАЗЦИ И ОБОСНОВКА НА ИЗБОРА ЗА ПРОЕКТИРАНОТО ПРИЛОЖЕНИЕ
 - **ИНФРАСТРУКТУРНИ ИЗИСКВАНИЯ** — НАПРИМЕР
 - МОДЕЛ НА ОБСЛУЖВАНЕТО — КЛИЕНТ-СЪРВЕР ИЛИ P2P ВКЛЮЧИТЕЛНО
 - СОФТУЕРЕН МОДЕЛ — НАПРИМЕР MVC, MASTER-SLAVES — И ДР. ПРИЛОЖИМИ
 - ПО ВЪЗМОЖНОСТ **ОЦЕНКИ** НА :
 - ИЗИСКВАНИЯ КЪМ ПРОИЗВОДИТЕЛНОСТТА И СВЪРЗАНОСТТА НА СЪРВЕРИТЕ И/ЛИ КЛИЕНТСКИТЕ УСТРОЙСТВА
 - МОДЕЛ НА НАТОВАРВАНЕТО — ТЕМП И СЛОЖНОСТ НА ПОТРЕБИТЕЛСКИТЕ ЗАЯВКИ ЗА ОБСЛУЖВАНЕ, СИСТЕМЕН СВРЪХТОВАР
 - **КОМЕНТАРИ И ОБОСНОВКА** НА ИЗБРАНОТО РЕШЕНИЕ ЗА ПРОЕКТИРАНОТО ПРИЛОЖЕНИЕ

Фази на софтуерния проект

- **ФАЗА ПРОЕКТИРАНЕ:**

- **ФУНКЦИОНАЛНО ПРОЕКТИРАНЕ (“USER GUIDE”):**

- ОПИСАНИЕ НА ИМПЛЕМЕНТИРАНИТЕ ФУНКЦИИ И ИНТЕРФЕЙСИТЕ КЪМ ТЯХ

- КОМЕНТАР НА АЛГОРИТЪМА И ДРУГИ ХАРАКТЕРИСТИКИ ЗА ВСЯКА ФУНКЦИЯ

- ИЗПОЛЗВАТ СЕ БЛОКОВИ ДИАГРАМИ, ДИАГРАМИ НА МАШИНАТА НА СЪСТОЯНИЕТО, ДИАГРАМИ НА ДЕЙНОСТИТЕ, ПОСЛЕДОВАТЕЛНОСТНИ ДИАГРАМИ И ДРУГИ ПОДХОДЯЩИ UML-2 ДИАГРАМИ

- ПОДХОДЯЩИ ИНТЕРФЕЙСНИ ЕКРАНИ

- ПОДХОДЯЩИ ЕКРАНИ ОТ РЕАКЦИЯТА НА ПРИЛОЖЕНИЕТО

- **НЕФУНКЦИОНАЛНО ПРОЕКТИРАНЕ (“TECHNICAL REFERENCE”):**

- ОПИСАНИЕ НА РАЗГРЪЩАНЕТО НА ПРИЛОЖЕНИЕТО

- КЛИЕНТСКА И СЪРВЕРНА ЧАСТ

- ИЗИСКВАНИ УСЛУГИ НА СЪОТВЕТНИТЕ ПЛАТФОРМИ

- ИЗИСКВАНА ПРОИЗВОДИТЕЛНОСТ - ВКЛ. ПОДДРЪЖКА НА МНОГОНИШКОВА/ПАРАЛЕЛНА ОБРАБОТКА

- ОЧАКВАНИ НЕФУНКЦИОНАЛНИ (QoS) ЧЕРТИ , НАПРИМЕР:

Фази на софтуерния проект

- ... ОЧАКВАНИ НЕФУНКЦИОНАЛНИ (QoS) ЧЕРТИ , НАПРИМЕР:
 - СРЕДНО ВРЕМЕ НА ОБСЛУЖВАНЕ НА ЗАЯВКИТЕ
 - ТЕМП НА НАТОВАРВАНЕТО ПО БРОЙ И СЛОЖНОСТ НА ЗАЯВКИТЕ
 - ПОДДРЪЖКА НА ГЪВКАВО РЕАЛНО ВРЕМЕ — НАПР. ПРИ МУЛТИМЕДИЙНО СЪДЪРЖАНИЕ
 - ПОДДРЪЖКА НА ТВЪРДО РЕАЛНО ВРЕМЕ — ПРИ КОНТРОЛНИТЕ СИСТЕМИ
 - ЗАЩИТА НА ИНФОРМАЦИЯТА — ПРОТОКОЛИ И МОДЕЛ НА АТАКАТА И ЗАЩИТАТА
 - ПРЕНОСИМОСТ НА ПРИЛОЖЕНИЕТО И УСЛУГИТЕ ПО ОТНОШЕНИЕ НА
 - СЪРВЕРНА ЧАСТ — ВКЛ. ИЗИСКВАНИЯ ПО ОТНОШЕНИЕ НА СИСТЕМНИТЕ УСЛУГИ — ОС, ПРОТОКОЛИ, СВЪРЗАНОСТ
 - КЛИЕНТСКА ЧАСТ — НАПР. ПРОИЗВОДИТЕЛНОСТ, ОС, КЛИЕНТСКА ПЛАТФОРМА — БРАУЗЪРИ, МОДЕЛИ И Т.Н.

Част от посочените тук описания са неприложими при проекта поради неговата ограниченост

Фази на софтуерния проект

- ... ФАЗА ПРОЕКТИРАНЕ
 - ТЕСТВАНЕ, НАСТРОЙКА, АПРОБАЦИЯ:
 - В РАЗГЪРНАТИТЕ СОФТУЕРНИ ПРОЕКТИ СЕ ИЗГОТВЯТ ПЛАН НА ТЕСТОВЕТЕ, ПРОТОКОЛИ ЗА ПРОВЕДЕНИТЕ ТЕСТОВЕ И ДРУГИ ПРОЦЕДУРИ ПО ОСИГУРЯВАНЕ НА КАЧЕСТВОТО;
 - ЗА ПРОЕКТА ПО МИС Е ДОСТАТЪЧНО:
 - КОМЕНТАР И ДОКАЗАТЕЛСТВО **ЗА ПРЕНОСИМОСТТА** ПО ОТНОШЕНИЕ НА
 - РАЗЛИЧНИ ТИПОВЕ КЛИЕНТСКИ УСТРОЙСТВА – ПРОИЗВОДИТЕЛНОСТ, РАЗМЕР И РЕЗОЛЮЦИЯ НА ЕКРАНА И ДР. Т.
 - ПРИЛОЖИМОСТ ЗА РАЗЛИЧНИТЕ ОС И ВЕРСИИ
 - ПРИЛОЖИМОСТ ЗА РАЗЛИЧНИТЕ БРАУЗЕРИ И ВЕРСИИ
 - КОМЕНТАР И ДОКАЗАТЕЛСТВО **ЗА ПРОИЗВОДИТЕЛНОСТТА** (САМО ПРИ ИЗПОЛЗВАНЕ НА МНОГОНИШКОВ КОД - НАПРИМЕР АНИМИРАНИТЕ СИМУЛАЦИИ)

Описанието се придружава с подходящи по брой и съдържание екрани от интерфейсите на приложението

Описанието се придружава с времедиаграми и анализ на ускорението при многонишкова обработка

Фази на софтуерния проект – моделиране

Динамично – например централизирано: пул от нишки, добро само при

• ФАЗА МОДЕЛИРАНЕ - ЕФЕКТИВНОСТ:

• ДЕКОМПОЗИЦИЯ

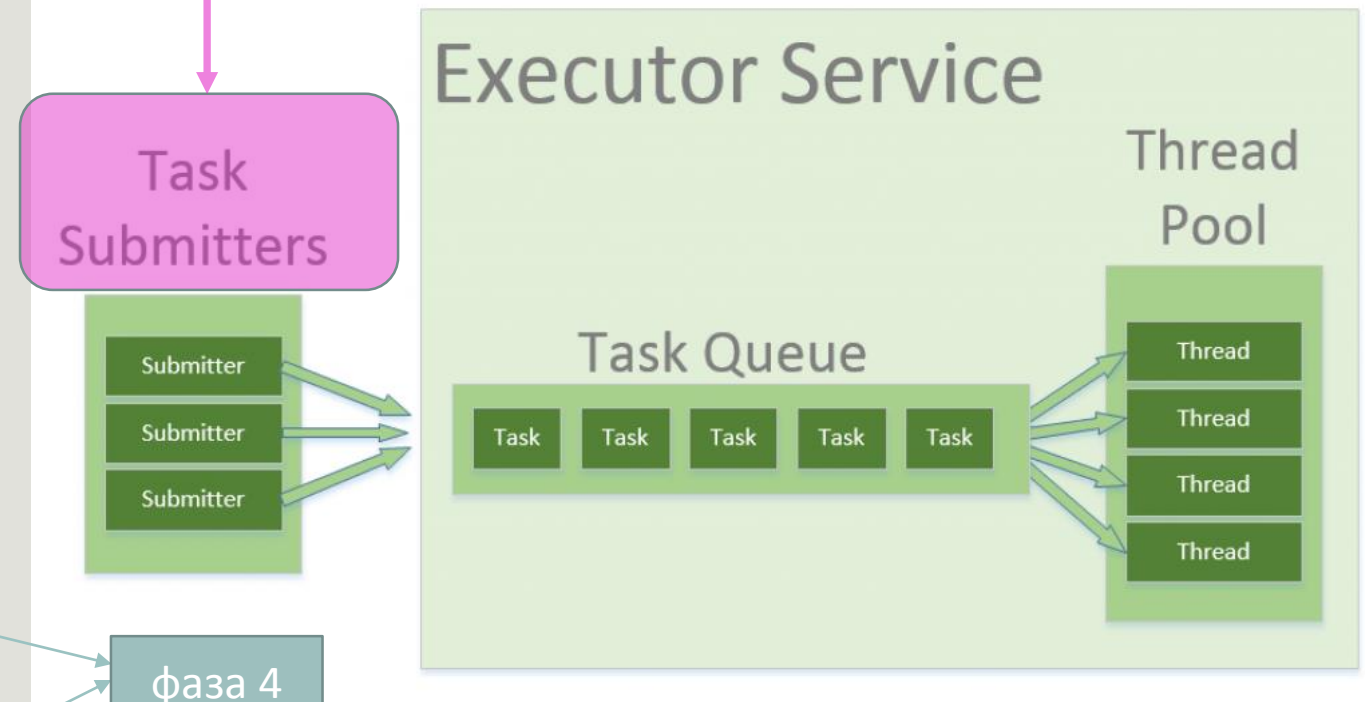
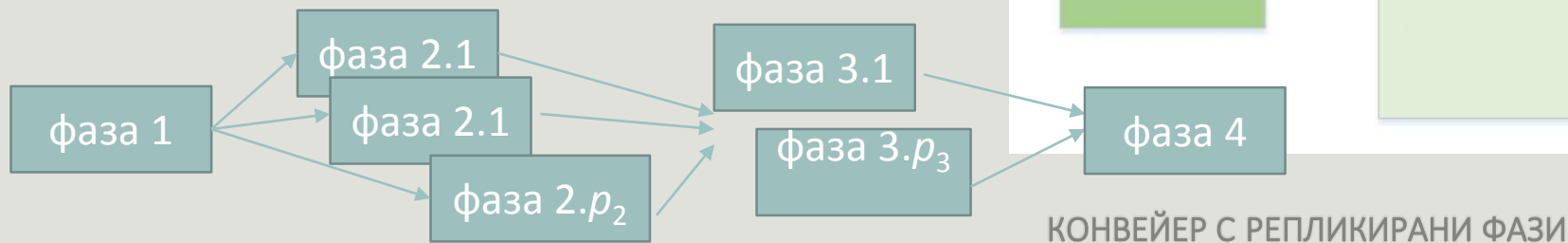
- ПО УПРАВЛЕНИЕ (MPMD = ПОТОВОВА АРХИТЕКТУРА) – КОНВЕЙЕР С РЕПЛИКИРАНИ ФАЗИ
- ПО ДАННИ (SPMD = MASTER/SLAVES)

• БАЛАНСИРАНЕ И РАЗПРЕДЕЛЯНЕ

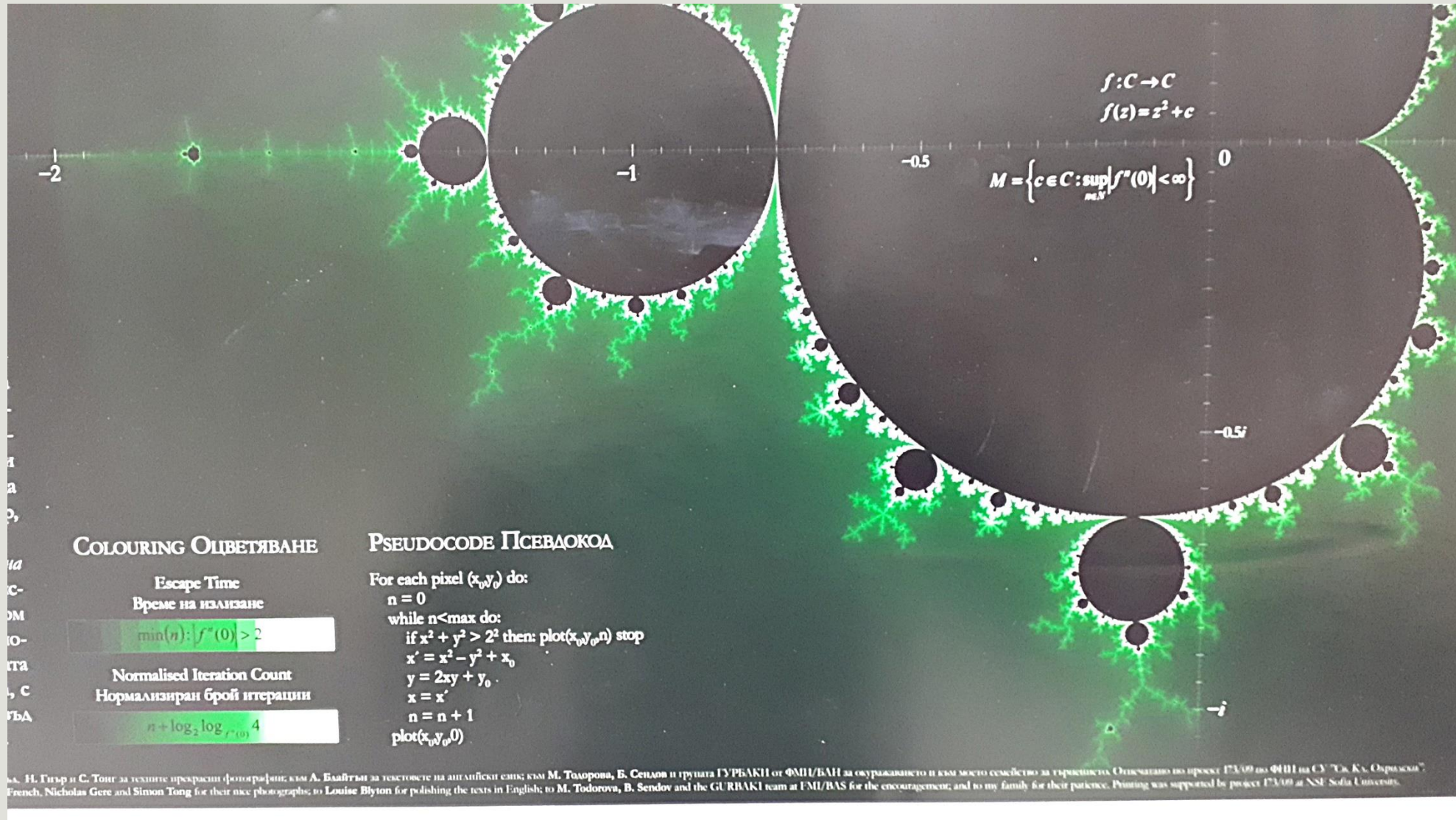
- **СТАТИЧНО** (ПРЕДВАРИТЕЛНО ФИКСИРАНЕ ПО РАВЕН БРОЙ ОТ N ПОДЗАДАНИЯ МЕЖДУ P НИШКИ)
- ДИНАМИЧНО **ПОДХОДЯЩО САМО ЗА ДИНАМИЧНО ПОСТЪПВАЩИ ЗАДАНИЯ:**

- 1) динамично постъпване на заданията &
- 2) малък паралелизъм – тук 4.

[*] Paraschiv E. Introduction to Thread Pools in Java, GuavaJava+ (<https://www.baeldung.com/thread-pool-java-and-guava>)



Избор на задача от теста на Манделброт



Тестов план и тестови резултати

1. ПОВТОРЯЕМОСТ ЗА ВСЕКИ ТЕСТОВ СЛУЧАЙ – **НАПРИМЕР 5 ПЪТИ** (СТАТИСТИЧЕСКИ ДОСТАТЪЧЕН БРОЙ ПЪТИ СПРЯМО ВЕРОЯТНОСТТА ЗА ФОНОВО НАТОВАРВАНЕ, ВЪЗМОЖНО В РАЗЛИЧНИ ЧАСОВЕ НА ДЕНЯ)

2. ПАРАМЕТРИЗАЦИЯ – ВАРИРАТЕ ПО СЪСТАВЕНИЯ ПЛАН

- ГРАНУЛАРНОСТ – НАПРИМЕР $N = g * p$, КЪДЕТО Е КОЕФИЦИЕНТ НА ГРАНУЛАРНОСТ И ЩЕ ВИ Е КОМАНДЕН ПАРАМЕТЪР – НАПРИМЕР $g = 1$ Е МАКСИМАЛНО ЕДРА, 4 Е ЕДРА, 16 СРЕДНА И 64 – КЪМ ФИНА;

- ВАРИАНТ – **ФИКСИРАНА ГРАНУЛАРНОСТ** ЗА АДАПТИВНОСТ

КЪМ РАЗМЕРИТЕ НА КЕША

- БАЛАНСИРАНО ПЛАНИРАНЕ –

- СТАТИЧНО С ОПИСАНИЕ НА РАЗПРЕДЕЛЯНЕТО – НАПРИМЕР ЦИКЛИЧНО,

- ЗА 6+ - ПУСКАТЕ И ДИНАМИЧНО ЦЕНТРАЛИЗИРАНО –

КАКЪВТО Е ПУЛЪТ ОТ НИШКИ И ПОКАЗВАТЕ КОЛКО ПО-ЛОШ Е РЕЗУЛТАТА,

И ЗА 6++ ДИНАМИЧНО РАЗПРЕДЕЛЕНО – НАПР. ВЕРИГАТА ОТ ИЗПЪЛНИТЕЛНИ НИШКИ В 3. Л-Я

- НАТОВАРВАНЕ – ПРОМЕНЯТЕ ТЕСТОВАТА ОБЛАСТ

- ЗА ПО-МАЛКИ И ПО-ГОЛЕМИ ЗАДАНИЯ – ПРОМЯНА РАЗМЕРА НА МАТРИЦАТА ИЛИ ПРОМЯНА НА ПОДОБЛАСТТА КЪМ ПО-ЛЕКА ИЛИ ПО-ТЕЖКА

- ЗА ПО-БАЛАНСИРАНИ И ПО-НЕБАЛАНСИРАНИ ЗАДАНИЯ (ПРИ МАНДЕЛБРОТ – ПО „КЪДРАВИ“ И НЕ-ДИСПЕРСНИ (С ОБОСОБЕНИ „ЧЕРНИ“ ИЛИ „БЕЛИ“ ОБЛАСТИ))

1	2	3	4	1	1	2	3	4	3
2	3	4	1	2	1	2	3	1	4
3	4	1	2	3	2	3	1	4	2
4	1	2	3	4	4	3	1	3	2

примерно разпределяне на подзадания по 4 нишки при статично циклично и при динамично балансиране

...Тестов план и тестови резултати

планират се 2 нишки на процесорно ядро (вместо 1), като едната заема инструкционния конвейер, а другата – операции с кеша или FPU

- 3. СТАТИСТИЧЕСКА ОБРАБОТКА НА ПОЛУЧЕНИТЕ РЕЗУЛТАТИ— В СЛУЧАЯ САМО НАЙ-ДОБРИЯ РЕЗУЛТАТ ОТ 5ТЕ ПОЛУЧЕНИ
- 4. ПАРАЛЕЛИЗЪМ $p = 1 \div 32+$: ПРИ МАШИНЕН ПАРАЛЕЛИЗЪМ 20 С ХИПЕРТРЕДИНГ УСКОРЕНИЕТО РАСТЕ ДО КЪМ 17-18 ЗА $p = 24-28$, ТЕСТВАЙТЕ ДО ДОСТИГАНЕ НА НЕМОНОТОННА АНОМАЛИЯ
- 4. МЕТРИКА — ВРЕМЕТО ОТ СТАРТА НА ПЪРВАТА НИШКА ДО ПРИКЛЮЧВАНЕ НА ПОСЛЕДНОТО ЗАДАНИЕ С ГЕНЕРИРАНЕ НА РЕЗУЛТАТА В ОП (ОСНОВНАТА ПАМЕТ)
 - БЕЗ ЗАПИСВАНЕ НА ФАЙЛ
 - БЕЗ ИЗВЕЖДАНЕ КЪМ ВИДЕОТО
- 5. ДОПЪЛНИТЕЛНИ ТЕСТОВЕ В НАСТРОЙВАЩ РЕЖИМ (DEBUG MODE) —
 - [ЗА ДИНАМИЧНОТО БАЛАНСИРАНЕ] СТАТИСТИКА КОЯ НИШКА КОИ И/ЛИ КОЛКО ЗАДАНИЯ Й СЕ Е ПАДНАЛО ДА ОБСЛУЖИ (НА ДВЕТЕ ТАБЛИЦИ — СТАТИЧНО-ЦИКЛИЧНО И ДИНАМИЧНО БАЛАНСИРАНЕ), ИМА ЛИ ИЗЧАКВАНЕ ЗА ДОСТЪП ДО ПУЛА
 - КОЛКО ВРЕМЕ ПРЕДИ КРАЯ НА ОБЩАТА ОБРАБОТКА Е ПРИКЛЮЧИЛА
 - ИМА ЛИ ИЗЧАКВАНЕ ЗА ДОСТЪП ДО МАТРИЦАТА С РЕЗУЛТАТИ — АКО ИМА — ПРАВИТЕ ВАРИАНТ С НЯКОЛКОКРАТНО ЗАПИСВАНЕ В МАТРИЦАТА ЗА КРАЙНИ РЕЗУЛТАТИ ТАКА ЧЕ ВСЯКА НИШКА ДА ПРАВИ ДОСТЪП В РАЗЛИЧЕН МОМЕНТ

Примерен тестов план с резултати

#	p	g	$T_1^{(1)}$	$T_1^{(2)}$	$T_1^{(3)}$	$T_1 = \min()$	$T_p^{(1)}$	$T_p^{(2)}$	$T_p^{(3)}$	$T_p^{(4)}$	$T_p^{(5)}$	$T_p = \min()$	$S_p = T_1/T_p$	$E_p = S_p/p$
1	2	1	$T_1^{(1)}$	$T_1^{(2)}$	$T_1^{(3)}$	0	$T_2^{(1)}$	$T_2^{(2)}$	$T_2^{(3)}$			0	#DIV/0!	#DIV/0!
2	4						$T_4^{(1)}$	$T_4^{(2)}$	$T_4^{(3)}$			0	#DIV/0!	#DIV/0!
3	8						$T_8^{(1)}$	$T_8^{(2)}$	$T_8^{(3)}$			0	#DIV/0!	#DIV/0!
4	12						$T_{12}^{(1)}$	$T_{12}^{(2)}$	$T_{12}^{(3)}$			0	#DIV/0!	#DIV/0!
5	16						$T_{16}^{(1)}$	$T_{16}^{(2)}$	$T_{16}^{(3)}$			0	#DIV/0!	#DIV/0!
6	20						$T_{20}^{(1)}$	$T_{20}^{(2)}$	$T_{20}^{(3)}$			0	#DIV/0!	#DIV/0!
7	24						$T_{24}^{(1)}$	$T_{24}^{(2)}$	$T_{24}^{(3)}$			0	#DIV/0!	#DIV/0!
8	28						$T_{28}^{(1)}$	$T_{28}^{(2)}$	$T_{28}^{(3)}$			0	#DIV/0!	#DIV/0!
9	2	5	$T_1^{(1)}$	$T_1^{(2)}$	$T_1^{(3)}$	0	$T_2^{(1)}$	$T_2^{(2)}$	$T_2^{(3)}$			0	#DIV/0!	#DIV/0!
10	4						$T_4^{(1)}$	$T_4^{(2)}$	$T_4^{(3)}$			0	#DIV/0!	#DIV/0!
11	8						$T_8^{(1)}$	$T_8^{(2)}$	$T_8^{(3)}$			0	#DIV/0!	#DIV/0!
12	12						$T_{12}^{(1)}$	$T_{12}^{(2)}$	$T_{12}^{(3)}$			0	#DIV/0!	#DIV/0!
13	16						$T_{16}^{(1)}$	$T_{16}^{(2)}$	$T_{16}^{(3)}$			0	#DIV/0!	#DIV/0!
14	20						$T_{20}^{(1)}$	$T_{20}^{(2)}$	$T_{20}^{(3)}$			0	#DIV/0!	#DIV/0!
15	24						$T_{24}^{(1)}$	$T_{24}^{(2)}$	$T_{24}^{(3)}$			0	#DIV/0!	#DIV/0!
16	28						$T_{28}^{(1)}$	$T_{28}^{(2)}$	$T_{28}^{(3)}$			0	#DIV/0!	#DIV/0!
17	2	10	$T_1^{(1)}$	$T_1^{(2)}$	$T_1^{(3)}$	0	$T_2^{(1)}$	$T_2^{(2)}$	$T_2^{(3)}$			0	#DIV/0!	#DIV/0!
18	4						$T_4^{(1)}$	$T_4^{(2)}$	$T_4^{(3)}$			0	#DIV/0!	#DIV/0!
19	8						$T_8^{(1)}$	$T_8^{(2)}$	$T_8^{(3)}$			0	#DIV/0!	#DIV/0!
20	12						$T_{12}^{(1)}$	$T_{12}^{(2)}$	$T_{12}^{(3)}$			0	#DIV/0!	#DIV/0!

В този план варираме паралелизма p и грануларността g :

1 – макс. едра

4 – едра

8 – средни

и т. н.

Докладваме $i=3$ до 5 измервания $T_p^{(i)}$, статистически ги обработваме (в случая минимума) и по формулата изчисляваме S_p и E_p , които представяме в две фамилии XY-диаграми $S_p(p)$ и $E_p(p)$ за различните g , вж. диаграмите във 2. л-я

Съдържание на документа (записката)

- ДОКУМЕНТЪТ СЛЕДВА ФАЗИТЕ НА АНАЛИЗ И ПРОЕКТИРАНЕ И СЪДЪРЖА СЛЕДНИТЕ **ЗАДЪЛЖИТЕЛНИ** ЧАСТИ
- 1. Увод – ЦЕЛ И ПРЕДНАЗНАЧЕНИЕ НА ПРОЕКТИРАНОТО ПРИЛОЖЕНИЕ, ВКЛ. ИМЕ НА ПРОЕКТА, КАКТО И СЛЕДНИТЕ ТОЧКИ
 - 1.1. „Функционален анализ“. **N.B.!** Заглавията и подзаглавията в документа се конкретизират според съдържанието – т.е. т. 1.1. не се озаглавява Функционален анализ, а например Обзор на функциите в сайтовете за търговия с музикални инструменти
 - 1.2. „Нефункционален анализ“ – приложимост на графични библиотеки и т.н.
- 2. ПРОЕКТИРАНЕ
 - 2.1. „User Guide“ – със съответното съдържание, диаграми, най-характерни примери от кода
 - 2.2. „Technical Reference“ – аналогично
- 3. ТЕСТВАНЕ, ПРЕНОСИМОСТ, НАСТРОЙКА – съгл. слайд 6.
 - По конкретния проект се изследват различни параметри на паралелизма – брой паралелни процеси, варианти на планирането на задания, варианти на декомпозицията на данни – например грануларност. В подходяща форма – основно семейство криви на ускорението – се представят различните резултати и се коментират оптималните параметри на предложения паралелен алгоритъм
- 4. Източници – номериран списък съгласно слайд 2. в азбучен ред или по реда на първо цитиране в т. 1.
 - Всеки източник от списъка трябва **да бъде цитиран** чрез своя номер в текста по тт. 1.1. и 1.2. (списъкът не може да съдържа източници, които не са цитирани в тези точки)

Специфични изисквания към проекта по РСА/СПО/РИТАрх

Документът следва фазите и структурата от слайд 7. със следните особености, произтичащи от ограничената функционалност на курсовия проект

1. По фаза/секция Анализ. Функционалният анализ не изисква проектиране на потребителски роли, техните случаи на употреба и съответно отпада Диаграмата на случаите на употреба, както и интервюто на потребителите с различни роли. По тази точка се излага накратко реализираната функция/алгоритъм и се съставя обзор на избраните източници за решаване на този алгоритъм със сравнителна таблица или друга подходяща диаграма. Същите и/ли други източници се анализират и в Технологичния (нефункционалният) анализ, където трябва да се обоснове приложимостта на избрания междинен слой - например протокол за общи променливи, протокол за обмен на съобщения и т.н.

2. По фаза Проектиране

2.1. Функционално проектиране – описанието на проектирания алгоритъм трябва да се предшества от инженерно представяне на общата архитектура на приложението с подходящи UML-диаграми. По конкретно трябва да се даде ясен отговор на въпросите:

- ❑ Модел на паралелизма – най-често SPMD, вкл. **МОДЕЛ НА СОФТУЕРНАТА АРХИТЕКТУРА** – например MASTER-SLAVES, с UML-диаграми
- ❑ Модел на **ДЕКОМПОЗИЦИЯ НА ДАННИТЕ** – например при P обработващи процеса (SLAVES) на колко части се разделят данните (грануларност на декомпозицията) и каква е схемата на тяхното разпределяне между процесите
 - ❑ **СТАТИЧНО БАЛАНСИРАНЕ**, заложено в програмата разпределяне – например циклично (базирано на процесния номер), случайно или друго по-“хитро” статично разпределяне
 - ❑ **ДИНАМИЧНО БАЛАНСИРАНЕ** – разпределяне по време на обработката – например всеки процес обслужва поредното под-задание от началото на опашка на заданията

... Специфични изисквания към проекта по РСА/СПО/РИТАрх

2. ...По фаза ПРОЕКТИРАНЕ

2.2. НЕФУНКЦИОНАЛНО ПРОЕКТИРАНЕ — ТРЯБВА ДА СЕ ДАДЕ ЯСЕН ОТГОВОР НА ВЪПРОСИТЕ:

- ❑ Модел на **КОМУНИКАЦИЯТА/СИНХРОНИЗАЦИЯТА** МЕЖДУ ПРОЦЕСИТЕ — Т.Е. **[GM|DM]-[SV|MP]:**
 - ❑ [локално | глобално]-синхронен или асинхронен алгоритъм
 - ❑ ИЗБОР, РОЛЯ И ИЗПОЛЗВАНИ ФУНКЦИИ НА МЕЖДИННИТЕ ПРОТОКОЛИ
 - ❑ ВЪЗМОЖНИ ПРОБЛЕМИ И МЕРКИ ЗА ОТСТРАНЯВАНЕ НА „СТРАНИЧНИТЕ ЕФЕКТИ“ (ЗАКЪСНЕНИЯ, БЛОКИРОВКИ, ВЗАИМНИ БЛОКИРОВКИ ИЛИ ПРИОРИТЕТНА ИНВЕРСИЯ)

3. ПО Т. ТЕСТВАНЕ И НАСТРОЙКА

- ❑ ДИАГРАМИТЕ ДА СА ВЪВ ВИДА НА ПРИМЕРНИТЕ ДИАГРАМИ ЗА УСКОРЕНИЕ И ЕФЕКТИВНОСТ КЪМ ВТОРА ЛЕКЦИЯ
- ❑ ДА СЕ КОМЕНТИРА ПОЛУЧЕНАТА КРИВА НА УСКОРЕНИЕТО И ВЪЗМОЖНИТЕ МЕРКИ ЗА ПО-ДОБРА ЛИНЕЙНОСТ (СКАЛИРУЕМОСТ)
 - ❑ ПРЕПОРЪЧИТЕЛЕН АНАЛИЗ ПО ТАЗИ ТОЧКА Е ДА СЕ ПРИВЕДАТ СЕМЕЙСТВО КРИВИ ЗА УСКОРЕНИЕТО ПРИ РАЗЛИЧНИ ПАРАМЕТРИ НА ПАРАЛЕЛИЗМА — НАПР. РАЗЛИЧНА ГРАНУЛАРНОСТ, РАЗЛИЧНА СХЕМА НА БАЛАНСИРАЩО РАЗПРЕДЕЛЯНЕ НА ПОДЗАДАНИЯТА И Т.Н. — С КОЕТО ДА СЕ ОПТИМИЗИРА ПРОЕКТИРАНОТО ПРИЛОЖЕНИЕ

Обобщение: Задължителни атрибути на проекта

- За т. 1.1. Функционален анализ: **Текстови преглед (описание по 1 параграф)** на 2.-3.-4. източника, описващи паралелното решение на аналогичен алгоритъм и при сходен паралелизъм и
 - съпоставянето на описаните в текста параметри в **сравнителна таблица**, подобна на таблицата от 3.33 **с параметрите на паралелизма**: максимален паралелизъм на изследването, синхронност, типа декомпозиция т.е. модела, типа балансиране и грануларност, но с добавени контекстни параметри
 - **с допълнителни контекстни параметри** – например при Манделброт и Wator размер на изследваната област (**за Манделброт 4К-матрица и дълбочина на итерациите 256**), при nBody, и сортировките – брой тела/елементи и т.н.
- За т. 1.2 Технологичен анализ – за същите и евентуално други образци на каква платформа са изпълнени – мултипроцесор плюс система за общи променливи или обмен на съобщения (според обявените данни в съответния източник)
- За т. 2.1 Функционално проектиране: описание на модела вкл. **последователностна диаграма** (UML – вж. 5. и 6. л-я)
- За т. 2.2. Технологично проектиране:
 - точно описание на характеристиките на машината – **модел процесор, памети – L1, L2, L3 и основна памет**
 - използвани примитиви за **стартирането на p нишки и за обмена между тях** (съответните няколко реда от кода)
- За т. 3. Тестови резултати
 - **Тестовия план съгласно таблицата от слайд 11.** и получените диаграми за ускорението на всички тестови случаи
 - Коментар на резултатите – **каква е скалируемостта и предложение за адаптивност** на паралелната програма към машини с различен паралелизъм и памет – **взимате машинните параметри чрез LS CPU.**
- За Източници – качествени източници с **пълно цитиране – Автори, Заглавие, Издател, Сайт.** (лекции, учебници, википедия не са препоръчителни и адекватни източници в общия случай, тъй като издават началното ниво на автора на съответния проект)