



**Нижегородский государственный университет
им. Н.И.Лобачевского**

Факультет Вычислительной математики и кибернетики

Образовательный комплекс

Технологии разработки параллельных программ

**Intel Thread Profiler
Лабораторная работа №1
Распределение нагрузки**

Корняков К.В., Шишков А.В.
Кафедра математического
обеспечения ЭВМ

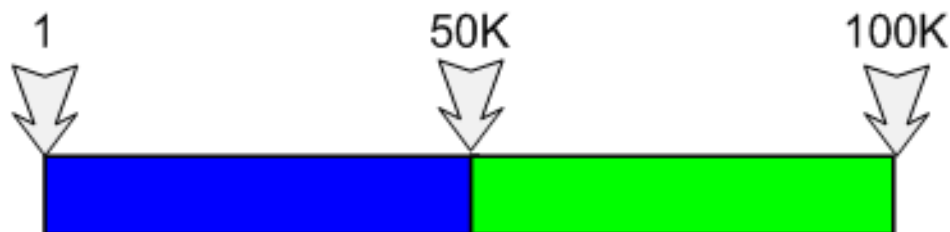
Цель лабораторной работы

- Изучить основные вопросы, возникающие при распределении нагрузки между потоками.
- Рассмотреть различные подходы к балансировке, приобрести навыки их анализа и сравнения между собой при помощи Intel® Thread Profiler (ITP).



Задание 1

- ❑ Откройте проект **Factorization** в **Microsoft Visual Studio 2005**.
- ❑ Подготовьте приложение к профилированию.
- ❑ Изучите функцию `factorization1`.

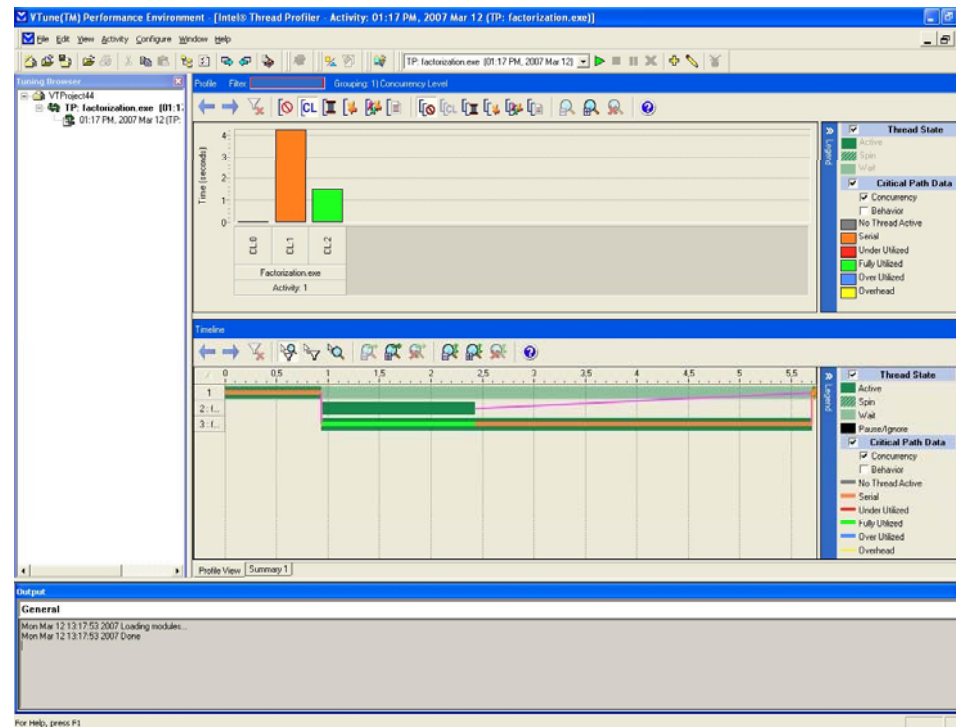


- ❑ Создайте проект ITP.
- ❑ Запустите процесс профилирования.
- ❑ Проанализируйте полученные результаты.



Подход 1

Результаты

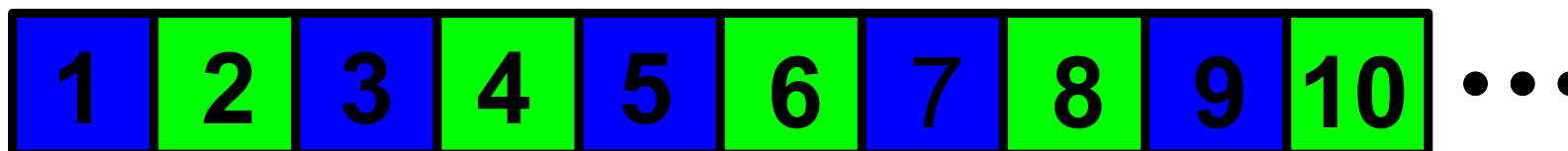


□ Зафиксируйте время работы



Задание 2

- ❑ Вернитесь **Microsoft Visual Studio**.
- ❑ Изучите функцию `factorization2`.

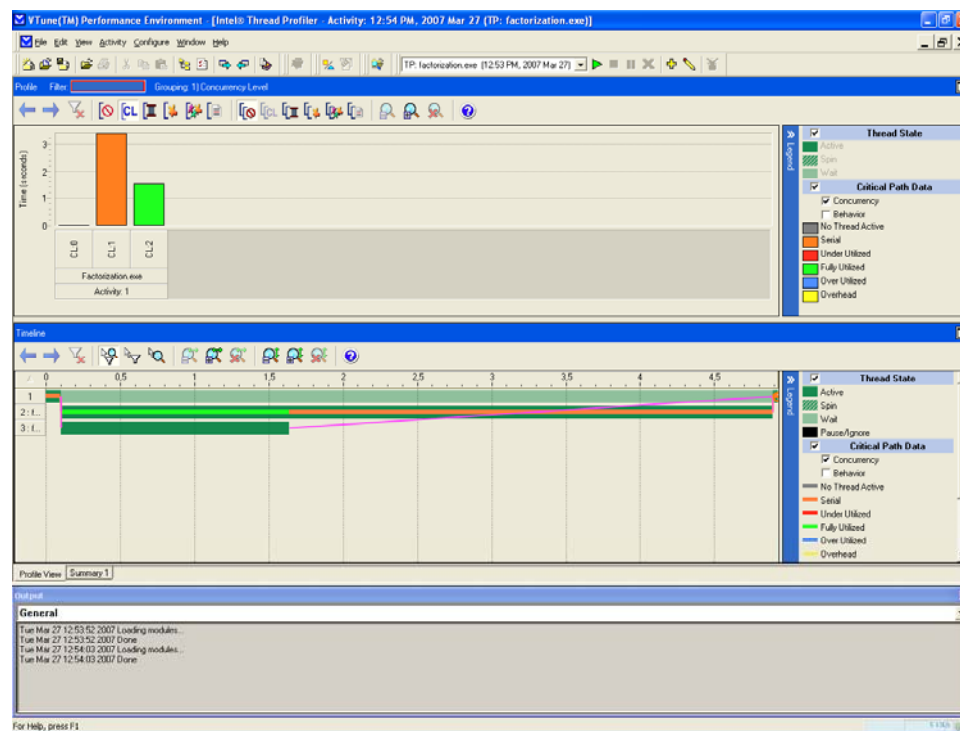


- ❑ Замените аргумент функции `CreateThread` на `factorization2`.
- ❑ Вернитесь в ИТР и запустите процесс профилирования.
- ❑ Проанализируйте полученные результаты.



Подход 2

Результаты



□ Зафиксируйте время работы



Задание 3

- ❑ Вернитесь **Microsoft Visual Studio**.
- ❑ Изучите функцию `factorization3`.

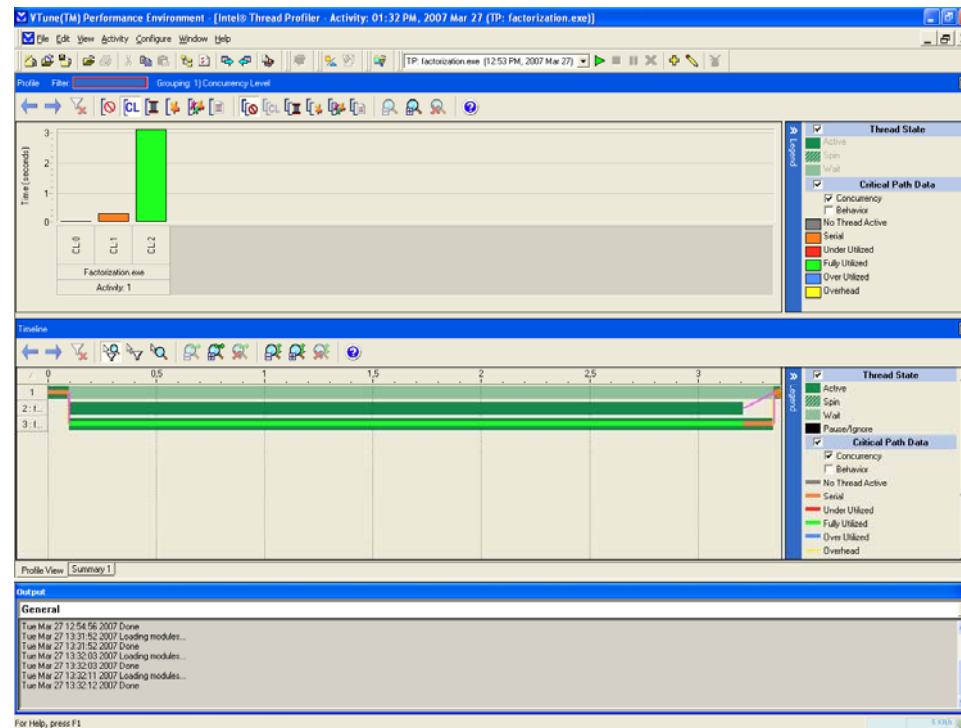


- ❑ Замените аргумент функции `CreateThread` на `factorization3`.
- ❑ Вернитесь в ИТР и запустите процесс профилирования.
- ❑ Проанализируйте полученные результаты.



Подход 3

Результаты



- Сравните время работы приложения, использующего этот подход с рассмотренными ранее.



Самостоятельная работа

- Определите оптимальную степень гранулярности для третьего подхода
- Ответьте на контрольные вопросы
 - В задачах какого класса должен хорошо показать себя первый из описанных подходов распределения нагрузки?
 - Как вы думаете, если использовать второй подход с четырьмя потоками, равномерно ли распределится нагрузка между ними? Почему? Проведите эксперимент и проверьте свои предположения.
 - Предложите свой метод распределения нагрузки между потоками. Если имеется время, запрограммируйте его и сравните с подходами, приведенными в настоящем документе.
 - Как вы думаете, какой должен быть следующий шаг, если мы хотим продолжить повышение производительности приложения **Factorization**?



Резюме

- ❑ Нередко причина низкой производительности – большая доля последовательных вычислений в приложении.
- ❑ ITP позволяет определить участки последовательного исполнения и время работы каждого из потоков.
- ❑ Далее разработчик либо распараллеливает последовательный участок, либо перераспределяет нагрузку.



Источники информации

1. «Developing Multithreaded Applications: A Platform Consistent Approach», Intel Corporation, March 2003.
2. «Threading Methodology: Principles and Practices», Intel Corporation, March 2003.
3. «Multi-Core Programming for Academia», Student Workbook, by Intel.
4. «Multi-Core Programming», book by Sh. Akhter and J. Roberts, Intel Press 2006.
5. «Intel® Thread Profiler. Getting Started Guide».
6. «Intel® Thread Profiler. Guide to Sample Code».
7. «Intel® Thread Profiler Help».
8. «Intel® Thread Profiler – краткое описание», материалы по образовательному комплексу «Технологии разработки параллельных программ».

