

вирішення проблем інтернет-конвергенції комунікаційних послуг.

Перелік джерел посилання

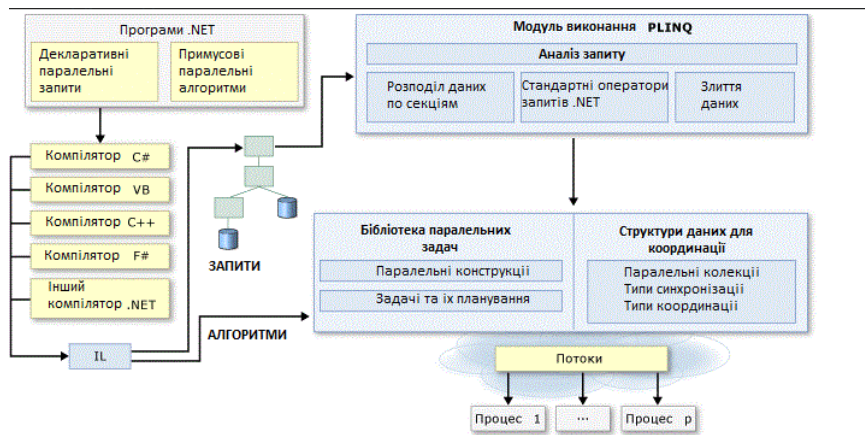
1. Вікіпедія [Електронний ресурс] –<http://en.wikipedia.org/wiki/WebRTC>.
2. Основи WebRTC (інструкція) [Електронний ресурс] – <http://www.html5rocks.com/en/tutorials/webrtc/basics/>.
3. WebRTC - Огляд [Електронний ресурс] – <http://www.pkeconsulting.com/pkewebrtc.pdf>.
4. WebRTC, питання та відповіді [Електронний ресурс] – <http://www.webtorials.com/content/2013/05/eleven-answers-webrtc-explained.html>.
5. WebRTC - новий метод веб комунікації [Електронний ресурс] – <http://webrtcbook.com/presentations/WebRTCIEEE04-02-13.pdf>.
6. Офіційний веб-сайт WebRTC [Електронний ресурс] – <http://www.webrtc.org/>.

*Кривонос Михайло Олександрович,
студент 3 курсу освітньої програми Середня
освіта (Інформатика)
Житомирський державний університет імені
Івана Франка, Житомир*

ПАРАЛЕЛЬНЕ ПРОГРАМУВАННЯ В .NET

Персональні комп'ютери та робочі станції оснащені декількома ядрами центрального процесора, що надають можливість одночасно виконувати декілька потоків. Для того щоб використати дану перевагу обладнання, можна використати паралельне програмування для розподілення роботи між декількома процесорами.

Донедавна паралельне програмування потребувало керування потоками та взаємоблокування на низькому рівні. Visual Studio і .NET надають розширену підтримку паралельному програмуванню через середовище виконання, типів бібліотек класів та середовище діагностики. Вперше ці можливості з'явилися в .NET Framework 4 і вони значно спростили паралельне програмування. Це надає змогу створювати ефективний, деталізований та масштабований код за допомогою природніх засобів без необхідності роботи з потоками на пряму. Наведемо загальну архітектуру паралельного програмування в .NET.



Бібліотека паралельних задач TPL. Бібліотека паралельних задач TPL являє собою набір відкритих типів і API-інтерфейсів у просторі імен System.Threading та System.Threading.Tasks .

Завдання TPL – підвищення продуктивності розробників за рахунок спрощення процесу додавання паралельного програмування в розробку додатків. TPL динамічно масштабує ступінь паралельності для найбільш ефективного використання використання всіх доступних процесів. Крім того, в бібліотеці паралельних задач виконуються секційні процеси, планування потоків в пулі ThreadPool, підтримка відміни, керування станом та інші низькорівневі задачі. Завдяки використанню бібліотеки паралельних задач можна збільшити ККД коду, зосередившись на роботі , для якої призначена програма.

Паралелізм даних. Поняття паралелізму даних відноситься до сценаріїв, в яких одна й тає операція виконується одночасно (тобто паралельно) для елементів в масиві або початковій колекції. В паралельних операціях з даними початкова колекція будується у такий спосіб, щоб декілька потоків могли одночасно виконуватись в різних сегментах.

Бібліотека паралельних задач TPL підтримує паралелізм даних за допомогою класу System.Threading.Tasks.Parallel. Діний клас надає паралельну реалізацію на основі методів циклів for і foreach (for і foreach в Visual Basic). Ви описуєте логіку циклу для Parallel.For або Parallel.Foreach в значній степені так само, як би ви писали поступовий цикл. Немає необхідності створювати потоки або черги елементів. В базових циклах немає необхідності застосовувати

блокування. Бібліотека паралельних задач опрацьовує всі низькорівневі запити.

Приклад простого циклу foreach та його паралельний еквівалент.

//Sequential version	//Parallel equivalent
foreach(var item in sourceCollection) { Proces(item);}	Parallel.ForEach(sourceCollection, item => Proces(item));

Привиконанні паралельного циклу бібліотека паралельних задач розділяє початкові данні, для того щоб цикл мав можливість одночасно працювати в декількох частинах. Задача розділяється враховуючи симтемні ресурси та робоче навантаження. По можливості розпріділяє роботу по декількох потоках і процесам, якщо робоче навантаження стає незбалансованим.

Обидва методи Parallel.For і Parallel.ForEach мають декілька переважанень, які надають можливість призупиняти виконання циклу, відстежувати стан циклу в інших потоках, обслуговувати локальний стан потоку, завершувати локальні по відношенню до потоку об'єкти, керувати ступенем паралельності тощо. Допоміжні типи, забезпечують данну функціональну можливість та містять ParallelLoopState, ParallelOptions, ParallelLoopResult, CancellationToken, CancellationTokenSource.

*Воробкало Тетяна Василівна, к.т.н., доцент
Євтушенко Тетяни Сергіївни,
Воробкало Олексій Костянтинович,
Черкаський державний технологічний
університет, Черкаси*

ДОСЛІДЖЕННЯ ТА МОДЕЛЮВАННЯ АНТЕН ДЛЯ RFID СИСТЕМ

Останнім часом все більшої популярності набувають різноманітні цифрові технології. Однією з таких є технологія радіочастотної ідентифікації (RFID). Це спосіб автоматичної ідентифікації об'єктів, в якому за допомогою радіосигналів зчитуються або записуються дані в так званих "RFID-мітках". RFID-мітки зазвичай складаються з мікрочіпа та антени.