

Облікова картка дисертації

I. Загальні відомості

Державний обліковий номер: 0821U102140

Особливі позначки: відкрита

Дата реєстрації: 15-07-2021

Статус: Захищена

Реквізити наказу МОН / наказу закладу:



II. Відомості про здобувача

Власне Прізвище Ім'я По-батькові:

1. Ларін Владислав Олегович

2. Larin Vladyslav O.

Кваліфікація:

Ідентифікатор ORCID ID: Не застосовується

Вид дисертації: доктор філософії

Аспірантура/Докторантура: так

Шифр наукової спеціальності: 113

Назва наукової спеціальності: Математика та статистика. Прикладна математика

Галузь / галузі знань:

Освітньо-наукова програма зі спеціальності: Не застосовується

Дата захисту: 07-07-2021

Спеціальність за освітою: Програмне забезпечення систем

Місце роботи здобувача: Ларін Владислав Олегович

Код за ЄДРПОУ: 3404613370

Місцезнаходження: вул. В. Симоменка, 5А, кв. 159, м. Київ, 79071, Україна

Форма власності:

Сфера управління:

Ідентифікатор ROR: Не застосовується

III. Відомості про організацію, де відбувся захист

Шифр спеціалізованої вченої ради (разової спеціалізованої вченої ради): ДФ 26.001.164

Повне найменування юридичної особи: Київський національний університет імені Тараса Шевченка

Код за ЄДРПОУ: 02070944

Місцезнаходження: вул. Володимирська, буд. 60, м. Київ, 01033, Україна

Форма власності:

Сфера управління: Міністерство освіти і науки України

Ідентифікатор ROR: Не застосовується

IV. Відомості про підприємство, установу, організацію, в якій було виконано дисертацію

Повне найменування юридичної особи: Київський національний університет імені Тараса Шевченка

Код за ЄДРПОУ: 02070944

Місцезнаходження: вул. Володимирська, буд. 60, м. Київ, 01033, Україна

Форма власності:

Сфера управління: Міністерство освіти і науки України

Ідентифікатор ROR: Не застосовується

V. Відомості про дисертацію

Мова дисертації:

Коди тематичних рубрик: 28.25

Тема дисертації:

1. Предметно орієнтований підхід до розробки акторних систем.
2. Domain-specific approach to the actor systems.

Реферат:

1. У дисертації розглянуті види паралельних та розподілених систем та їх характеристики, зокрема детально розглянуті моделі акторів та Map-Reduce, як універсальні моделі паралельних обчислень, з широким практичним застосуванням. В той же час показано, що модель акторів хоча і є гнучкою і безпечною при роботі з даними, в ній відсутні ефективні способи миттєвої взаємодії – паралельного запису чи читання спільної пам'яті, що є дуже важливою особливістю не тільки для практичного застосування, а як показано і для окремого класу задач – стохастичної глобальної оптимізації, на прикладі задачі рою часток. Для введення примітивів спільної пам'яті у акторну модель запропоновано використати предметно-орієнтований підхід – визначено синтаксис і граматику предметно-орієнтованої мови Strumok. Ця мова є розвитком ідей візуального (Visual Akka) та предметно-орієнтовного (Flow) описів акторної моделі на фреймворці Akka. Ці моделі розширили базову акторну модель визначенням інтерфейсу взаємодії актора. Цей інтерфейс

визначає список повідомлень які може обробити актор – так званий список методів актора; строго типізований формат аргументів цих методів; формат повідомлень (зворотних значень) які можуть бути створені у результаті обробки цього повідомлення. Це надає змогу ще на етапі трансляції/компіляції візуальної чи предметно-орієнтованої мови виконати статичну перевірку типів у всіх взаємодіях акторів. Крім того, виклики методів та відправка результатів у цій моделі є абстрагованими – таким чином окремих актор стає ізольований від свого зовнішнього контексту виконання, делегуючи конкретне зв'язування методів актора актору вищого рівня. Такий механізм робить можливою композицію багатьох акторів; дозволяє перевикористовувати вже розроблені актори у інших програмних системах. Гібридна модель Strumok, у свою чергу, розширює предметно-орієнтовану акторну модель Flow додаванням примітивів спільної пам'яті та асинхронного виконання методів. Для кожної комірки спільної пам'яті у актора визначається один з можливих модифікаторів способу її спільного використання (призначення): тільки для читання, тільки для запису, читання і запису. Для змінних які доступні тільки для міжакторного спільного читання чи запису можливо виділити окрему область, яка в рамках вузла дозволить проводити паралельне читання чи запис без необхідності міжакторної синхронізації за допомогою повідомлень, при цьому на інших вузлах ця область буде реплікована у реальному часі (тобто на інших вузлах буде існувати затримка запису, або відставання часу останніх доступних даних для читання, яка складатиме часу мережевої затримки між двома вузлами у мережі). Для спільних змінних зі змішаним модифікатором використання, необхідно ввести додатковий рівень абстракції для забезпечення безпечного доступу до спільних даних.

2. The thesis considers the types of parallel and distributed systems and their characteristics, in particular the models of actors and Map-Reduce, as universal models of parallel computations, with wide practical application. At the same time, it is shown that the actor model, although flexible and secure when working with data, it lacks effective ways of instant interaction - parallel writing or reading shared memory, which is an important feature not only for practical application, but as is also shown for a separate class of problems - stochastic global optimization, on the example of the problem of a swarm of particles. To introduce primitives of shared memory into the actor model, it is proposed to use a domain-specific approach – thus the syntax and grammar of the domain-specific language Strumok are defined. This language is a continuation of the ideas of visual (Visual Akka) and domain-specific (Flow) definitions of the actor model on the Akka framework. These models extended the basic actor model by defining an actor interaction interface. This interface defines a list of messages that can be processed by the actor - the so-called list of methods of the actor; strictly typed format of arguments of these methods; message format (return values) that can be created as a result of processing this message. This allows to perform a static type check in all interactions of actors at the stage of translation / compilation of visual or domain-specific language. In addition, the method calls and results in this model are abstracted - thus the individual actor becomes isolated from his external execution context, delegating the specific linking of the actor's methods to the higher-level actor. This mechanism makes it possible to compose many actors; allows to reuse already developed actors in other software systems. The Strumok hybrid model, in turn, extends the domain-specific actor model Flow by adding primitives of shared memory and asynchronous execution of methods. For each shared memory cell, the actor determines one of the possible modifiers of the way it is shared (usage intent): read-only, write-only, read-write. For variables that are only available for inter-actor sharing or writing, it is possible to define a separate area that will allow parallel read or write within the node without the need for inter-actor synchronization via messages, while on other nodes this area will be replicated in real time (replicated nodes will have a write delay, or lag time of the last available data to read, which will be the network delay time between two nodes in the network). For shared variables with a mixed usage modifier, additional level of abstraction introduced to ensure secure access to the shared data. Based on all read and write interaction of the mixed cell of shared data, a finite automaton is generated, which describes the states of the sub-actor in which the interaction with the cell of shared data is isolated. In this way, each mixed cell of shared data creates a finite set of secure areas for parallel writing or reading. But unlike the unidirectional case, in a mixed cell, each area can only execute a specific subroutine (set of operators). That is, those parts of the actor's program logic that fall into the area of a finite automaton are executed on a separate sub-actor with an isolated cell of shared memory. In the case of

simultaneous overlapping areas, the sub-actor message queue is used to ensure the synchronization of the shared data cell. At the level of the syntax of the domain-specific Strumok language, each variable of the shared data is described by adding the modifier keyword. Depending on the selected modifier, the translator selects the desired model for representing this shared variable. It is important to note that the generated finite automaton for the representation of a common variable, allows in an asynchronous (non-blocking) style to control the execution of methods of all actors who are in the area of a mixed common data cell. That is, if multiple actors need to be synchronized to process a cell, their program threads will not be occupied, and the context of the current execution will be saved (closure is created) to continue processing after receiving the results of the computations on the shared variable. The thesis shows that such asynchrony can also be introduced for the interaction of different actors with each other, and thus allow in the actor method to freely call the methods of other actors with asynchronous awaiting of the results of their execution.

Державний реєстраційний номер ДіР:

Пріоритетний напрям розвитку науки і техніки:

Стратегічний пріоритетний напрям інноваційної діяльності:

Підсумки дослідження:

Публікації:

Наукова (науково-технічна) продукція:

Соціально-економічна спрямованість:

Охоронні документи на ОПВ:

Впровадження результатів дисертації:

Зв'язок з науковими темами:

VI. Відомості про наукового керівника/керівників (консультанта)

Власне Прізвище Ім'я По-батькові:

1. Іванов Євген Олександрович

2. Ivanov Yevhen O.

Кваліфікація: к.ф.-м.н., 01.01.09

Ідентифікатор ORCID ID: Не застосовується

Додаткова інформація:

Повне найменування юридичної особи:

Код за ЄДРПОУ:

Місцезнаходження:

Форма власності:

Сфера управління:

Ідентифікатор ROR: Не застосовується

VII. Відомості про офіційних опонентів та рецензентів

Офіційні опоненти

Власне Прізвище Ім'я По-батькові:

1. Мелашенко Андрій Олегович
2. Melashenko Andriy O.

Кваліфікація: к.ф.-м.н., 01.05.03

Ідентифікатор ORCID ID: Не застосовується

Додаткова інформація:

Повне найменування юридичної особи:

Код за ЄДРПОУ:

Місцезнаходження:

Форма власності:

Сфера управління:

Ідентифікатор ROR: Не застосовується

Власне Прізвище Ім'я По-батькові:

1. Ігнатенко Олексій Петрович
2. Ignatenko Oleksii P.

Кваліфікація: д.ф.-м.н., 01.05.03

Ідентифікатор ORCID ID: Не застосовується

Додаткова інформація:

Повне найменування юридичної особи:

Код за ЄДРПОУ:

Місцезнаходження:

Форма власності:

Сфера управління:

Ідентифікатор ROR: Не застосовується

Рецензенти

Власне Прізвище Ім'я По-батькові:

1. Жереб Костянтин Анатолійович
2. Zhereb Kostiantyn Anatoliiovych

Кваліфікація: к. ф.-м. н., 01.05.03

Ідентифікатор ORCID ID: Не застосовується

Додаткова інформація:

Повне найменування юридичної особи:

Код за ЄДРПОУ:

Місцезнаходження:

Форма власності:

Сфера управління:

Ідентифікатор ROR: Не застосовується

Власне Прізвище Ім'я По-батькові:

1. Кривий Сергій Лук'янович

2. Kryvyj Sergii L.

Кваліфікація: д.ф.-м.н., 01.05.01

Ідентифікатор ORCID ID: Не застосовується

Додаткова інформація:

Повне найменування юридичної особи:

Код за ЄДРПОУ:

Місцезнаходження:

Форма власності:

Сфера управління:

Ідентифікатор ROR: Не застосовується

VIII. Заключні відомості

**Власне Прізвище Ім'я По-батькові
голови ради**

Нікітченко Микола Степанович

**Власне Прізвище Ім'я По-батькові
головуючого на засіданні**

Нікітченко Микола Степанович

**Відповідальний за підготовку
облікових документів**

Реєстратор

**Керівник відділу УкрІНТЕІ, що є
відповідальним за реєстрацію наукової
діяльності**

Юрченко Т.А.

