

## Ефективне планування задач проекту у розподіленому середовищі

**Олександр Боголій**

аспірант кафедри управлінських технологій,  
ВНЗ «Університет економіки та права «КРОК», м. Київ, Україна,  
e-mail: boholiom@krok.edu.ua,  
ORCID: 0000-0003-0253-667X

Розподілена розробка програмного забезпечення стала дуже популярною в останні роки, допомагаючи компаніям досягти високої продуктивності за рахунок використання ресурсів у всьому світі, заощаджуючи витрати на виробництво та скорочуючи час виходу на ринок. Разом з тим, така організація роботи, разом із перевагами, створює і певні труднощі для менеджменту. Особливо складним у віддаленому розподіленому середовищі є процес розподілу завдань [1]. На додачу до ряду факторів, які зазвичай враховуються при плануванні розподілу задач між членами команди за умов спільного розміщення, менеджерам тепер доводиться мати справу з різними графіками роботи та різними часовими поясами співробітників.

Якщо обсяг завдань не повністю відомий на початку проекту, а завдання надходять динамічно, найкращими є гнучкі алгоритми планування проекту, що здатні адаптуватися до змін. Ці алгоритми повинні бути в змозі обробляти зміни в розкладі проекту та розподіляти ресурси для нових завдань, коли вони надходять.

Прикладами алгоритмів динамічного планування є:

- ***Earliest Start Time (EST)***: Алгоритм EST в першу чергу планує завдання з найранішим часом початку. Це може бути хорошим вибором для проектів, де важливо завершити проект якомога раніше.

- ***Earliest Finish Time (EFT)***: Алгоритм EFT в першу чергу планує завдання з найранішим часом завершення. EFT враховує час виконання самого завдання плюс EST цього завдання. Це дозволяє фокусуватися на тих завданнях, які найбільше впливають на терміни завершення проекту.

- ***Critical Path Method (CPM)***: Алгоритм CPM визначає критичний шлях проекту, який є найдовшою послідовністю завдань, які необхідно виконати, щоб проект завершився вчасно. Алгоритм CPM можна використовувати для планування завдань таким чином, щоб мінімізувати ризик затримки проекту.

- ***Program Evaluation and Review Technique (PERT)***: Алгоритм PERT подібний до алгоритму CPM, але враховує невизначеність у тривалості завдань. Це може бути хорошим вибором для проектів, у яких тривалість завдання невідома.

В умовах, коли проект складається з багатьох взаємопов'язаних задач, особливо важливо адаптувати підходи до планування, щоб враховувати різний робочий час працівників. Якщо не брати до уваги цього фактору, може скластися ситуація, коли члени команди змушені чекати значно довше, поки залежні задачі виконують їхні колеги в інших часових поясах.

Щоб ефективно спланувати завдання проекту в таких умовах, можна використати існуючі програмні рішення, з невеликими модифікаціями. Особливо корисними можуть бути програмні засоби, що використовуються в комп'ютерній інженерії для планування завантаження графу задачі на багатоядерні та багатопроцесорні системи. Візьмемо за приклад, команду розробників, що очікує на специфікацію від інших колег, після чого може виконувати задачі паралельно. Так команда може моделюватися як багатоядерний процесор, а задачі проекту як задачі, що завантажуються на процесор. Планувальник задач тепер буде шукати яким чином оптимально розподілити задачі між розробниками, щоб закінчити проект в найкоротший термін.

Нижче приведено приклади кількох таких рішень, що описані в дослідницьких статтях з комп'ютерної інженерії:

- **Cloudsim toolkit [2]** - це відкритий програмний пакет для моделювання та симуляції хмарних обчислень. Cloudsim toolkit широко використовується в академічному та промисловому середовищах. Він був використаний для публікації численних наукових статей та для розробки хмарних систем. Використовує мову програмування Java.

- **SimSo [3]** - це симулятор планування задач для багатопроцесорних систем реального часу. Дозволяє проводити швидкі симуляції та швидке прототипування політик планування за допомогою Python.

- **YARTISS [4]** - це симулятор планування для багатопроцесорних систем реального часу. Це вільне програмне забезпечення, написане мовою програмування Java, в якому користувач може симулювати виконання великої кількості завдань на багатопроцесорних системах.

Використання програмних симуляторів для планування розподілу задач проекту між працівниками, що знаходяться в різних часових поясах, може позитивно вплинути на терміни виконання проекту та зменшити періоди очікування.

**Ключові слова:** розподілена розробка ПО; планування задач; часові пояси.

### Список використаних джерел

1. Боголій, О. (2023). У Держава, регіони, підприємництво: інформаційні, суспільно-правові, соціально-економічні аспекти розвитку. Отримано 6 квітня 2023 року з <https://conf.krok.edu.ua/SRE/SRE-2022/paper/view/1339>.
2. Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, Buyya R. CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience* 2011; 41(1):23–50.
3. Maxime Chéramy, Anne-Marie Déplanche, Pierre-Emmanuel Hladik. *Simulation of Real-Time Multiprocessor Scheduling with Overheads. International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2013), Jul 2013, Reykjavik, Iceland. pp. 5-14. (hal-00815502).*
4. Younès Chandarli, Frédéric Fauberteau, Damien Masson, Serge Midonnet, Manar Qamhieh. YARTISS: A Tool to Visualize, Test, Compare and Evaluate Real-Time Scheduling Algorithms. *WATERS 2012, Jul 2012, Italy. pp.21--26. (hal-00691985v2).*