

УДК 004.75

І.А. Чесніший, М.В. Галт

Національний технічний університет України «Київський політехнічний інститут»,
Україна

РОЗРОБКА ПЛАНУВАЛЬНИКА СИСТЕМИ РЕАЛЬНОГО ЧАСУ

I.A. Chesnishii, M.V. Halt

IMPLEMENTATION OF SCHEDULER FOR REAL-TIME SYSTEMS

Для систем реального часу характерно наступне: гарантований час реакції на зовнішні події; жорстка підсистема планування процесів (високопріоритетні завдання не повинні витіснятися низькопріоритетні, за деякими винятками); підвищені вимоги до часу реакції на зовнішні події або реактивності (затримка виклику обробника переривання не більше десятків мікросекунд, затримка при перемиканні задач не більше сотень мікросекунд) [1].

Планувальник. Основою будь-якої системи реального часу є планувальник [2]. Планувальник займається прийомом та розподіленням задач між агентами. Одним з головних компонентів планувальника є балансувальник, який використовує алгоритм динамічного балансування DAGman [3]. Планувальник забезпечує паралельне виконання декількох задач, розподіляючи ресурси між різними завданнями різних користувачів. Важливою метою планувальника є ефективний розподіл відрізків системного часу за умови забезпечення користувачу часу очікування на прийнятному рівні. Крім цього, перед планувальником можуть стояти суперечливі одна одній цілі, такі, як мінімізація часу очікування при виконанні критично важливих завдань реального часу і максимальне використання ресурсів.

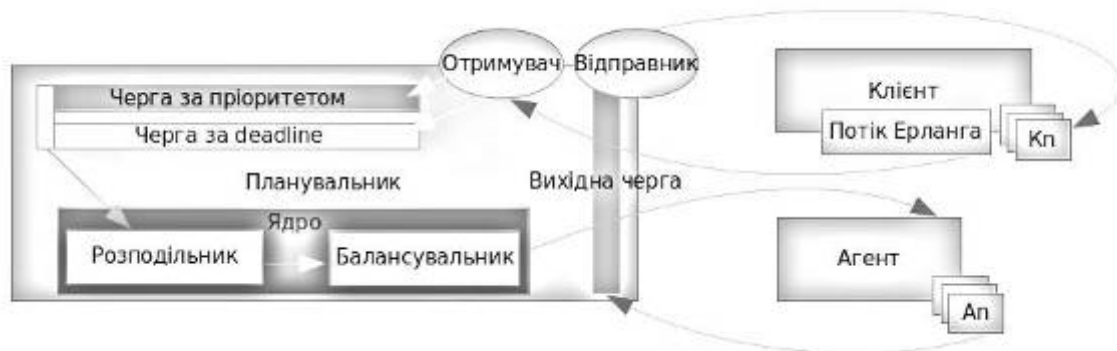


Рис. 1. Модель розробленого планувальника

На Рис.1 зображена модель планувальника, який складається з наступного:

- *клієнт* – програмний модуль, який генерує завдання за допомогою потоку Ерланга та відправляє їх планувальнику.
- *отримувач* – компонент планувальника, який отримує завдання від клієнтів та розподіляє їх по чергах.
- *черга за пріоритетом* – черга, сформована за пріоритетами виконання завдань.
- *черга за deadline* – черга, сформована за крайнім терміном виконання завдань.
- *розподільник* – компонент ядра планувальника, що виконує розподіл виконання високопріоритетних завдань та завдань, в яких спливає крайній термін виконання.
- *агент* – програмний модуль, що безпосередньо виконує завдання.

- *вихідна черга* – черга виконаних завдань, що чекають відправки клієнту.
- *відправник* – компонент планувальника, який розсилає готові результати виконання завдань клієнтам.

Система планувальника написана на мові програмування Java та працює на основі технології RMI (Remote Method Invocation), ця технологія дозволяє викликати методи віддалених об'єктів, які знаходяться на інших хостах. RMI виконує серіалізацію об'єктів і розпаковування параметрів без обрізання типів, підтримуючи справжній об'єктно-орієнтований поліморфізм. RMI здійснює менеджмент безпеки та доступу до об'єктів, що дозволило побудувати розподілений планувальник реального часу.

Клієнти. Клієнт представляє собою програмний модуль, який генерує випадковий сигнал. Генератор стаціонарного випадкового сигналу представлений як: спектральні складові сигналу; верхня частотна складова; кількість складових від 6 до 10; амплітуда; фаза.

Моделювання Cloud системи реального часу. Даний графік (Рис. 1) показує залежність довжини вхідної черги планувальника від часу. Спочатку було запущено 3 клієнта, що призвело до зростання кількості задач в черзі. Між 34 та 35 секундами був запущений агент, який швидко понизив кількість задач у черзі до мінімального значення.

Проведемо тест планувальника з вимкненим алгоритмом балансування, Генерація завдань: 300/с. Агенти: 20.

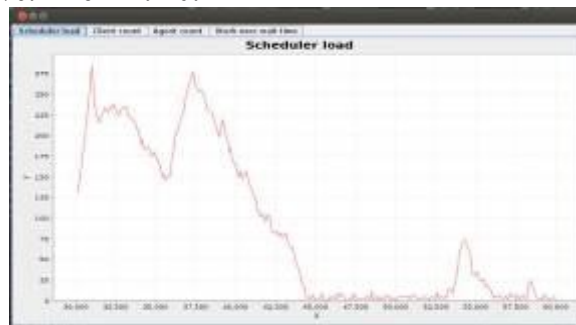


Рис. 2. Залежність довжини вхідної черги від часу

Отримані графіки показують, що ефективність планувальника зростає при збільшенні навантаження та достатній кількості ресурсів агентів. Після вимкнення алгоритму балансування, стабільність планувальника порушилася. Так як все навантаження лягло всього на декількох агентів, то планувальник не може нормально функціонувати в умовах підвищеного навантаження.

Висновки. Розроблено систему, яка складається з основних підсистем: планувальника завдань реального часу, клієнта та агента. Для всіх підсистем виконані етапи створення системних і функціональних вимог, визначено використовувані алгоритми і архітектури. Використана сучасна методологія розробки ПЗ: було розроблено та проаналізовано архітектуру програмного забезпечення та створено необхідні класи та інтерфейси. Також було проведено тестування системи планувальника, це дало змогу оцінити роботу планувальника та зміну його параметрів в залежності від часу та навантаження.

Література

1. Система реального часу [Електронний ресурс]. – Режим доступу: URL: http://ru.wikipedia.org/wiki/Система_реального_времени – Назва з екрану.
2. Планувальник [Електронний ресурс]. – Режим доступу: URL: http://uk.wikipedia.org/wiki/Планувальник_операційної_системи – Назва з екрану.
3. Directed Acyclic Graph Manager [Електронний ресурс]. – Режим доступу: URL: <http://research.cs.wisc.edu/htcondor/dagman/dagman.html> – Назва з екрану.