

МОДЕЛЮВАННЯ ЕНЕРГОСПОЖИВАННЯ БАГАТОЯДЕРНОГО ПРОЦЕСОРА КЛІЄНТСЬКОГО ПРИСТРОЮ ПІД ЧАС ОНЛАЙН-З'ЄДНАННЯ**І. М. Журавська, К. О. Обухова, В. Ю. Савінов***Чорноморський національний університет імені Петра Могили*

Анотація. Під час пандемії користувачі змушені працювати вдома через відеоконференції, інші онлайн-засоби. При цьому, наприклад, застосунок Zoom може діагностувати недостатню швидкість та не встановлювати зв'язок. В роботі показано, що причиною може бути перенавантаження кластера багатоядерного процесора (CPU) клієнтського пристрою. Шляхом моделювання енергоспоживання кожним ядром CPU з архітектурою *big.LITTLE* досягнуті результати, за якими можливо регулювати режими енергоживлення CPU. Це дозволить забезпечити безперервність сеансу зв'язку при відеоконференціях у виробництві, віддаленому наданні освітніх та медичних послуг.

Ключові слова: моделювання енергоспоживання, CPN Tools, багатоядерні процесори, онлайн відеоконференції.

Вступ

Прогноз розвитку суспільства, розповсюдження у світі коронакризи спричиняє подальше поширення комп'ютерних онлайн-технологій, стимулює пошуки вдосконалення шляхів оновлення існуючих та впровадження новітніх технологій спілкування у всіх галузях. Не оминає це і таку галузь, як охорона здоров'я. Принципово важливим на сьогодні є орієнтація на розвиток системи сімейних лікарів. Остання безумовно приводить до потреби віддаленої діагностики, профілактики та реабілітації, у т. ч. з використанням вдома у пацієнтів малогабаритних приладів медичного-призначення на базі комп'ютерно-інтегрованих технологій. Крім того, проведення лікування здійснюється не контролювано сімейним лікарем, а у медичних приватних центрах, кабінетах, фітнес-центрах тощо на приватних тренажерах. Така, з одного боку, неконтрольованість, а з іншого – і неможливість контролю, приводить до розбалансованості оздоровчого процесу і, безумовно, буде давати збої. Як показують дослідження, створення мереж та засобів включення до них приладів індивідуального призначення, віддалений контроль, запис та накопичення біометричних параметрів пацієнтів є задачею, що принципово реалізується. У зв'язку з цим, у статті зосереджено увагу на проблемах підвищення надійності та стабільності технологій комунікації (відеозв'язок, відеоконференції, тощо) за рахунок моделювання енергоспоживання багатоядерного процесора клієнтського пристрою під час онлайн-з'єднання.

1. Аналіз останніх досліджень і публікацій

На сучасному етапі розвитку систем керування необхідно забезпечити моніторинг режимів роботи віддалених малогабаритних приладів відновлення рухів пацієнтів протягом всього часу виконання завдання, особливо на перших етапах пост-інсультної та пост-інфарктної реабілітації поза межами закладів охорони здоров'я [1]. З урахуванням ємності акумуляторів сучасних гаджетів, до яких надходять та узагальнюються показники під час віддалених сеансів реабілітації, такий час становить від 20 до 240 хвилин.

В такому разі треба приділити увагу контролю та зменшенню енергоспоживання мобільного пристрою, за допомогою якого здійснюється супроводження реабілітаційної місії, особливо якщо це пов'язано з переглядом та передачею відеоконтенту. При цьому необхідно чергувати режими керування та обробки інформації [2].

Також важливим є забезпечення стабільної та тривалої роботи мобільних пристроїв для використання сучасних засобів комунікації. На сьогодні все частіше у одного сімейного лікаря виникає необхідність проведення індивідуальних, а іноді і декількох індивідуальних або групових консультацій одночасно, у форматі відеоконференції чи за допомогою спеціалізованих застосунків, що використовують хмари з шифруванням потоків даних [3].

Прикладом може бути проведення лікарем групових занять для пост-інсультної та пост-інфарктної реабілітації через такі застосунки, як Zoom, Microsoft Teams, Google Meet, Skype тощо.

Для реалізації такого онлайн-спілкування з'являються нові архітектури персональних гаджетів, що забезпечують кращу продуктивність. Зменшення енергоспоживання цих архітектур є

однією з головних проблем досягнення високопродуктивних обчислень. Сучасні напрямки досліджень спрямовані на розробку нових програмних та апаратних методів для досягнення найкращих показників ефективності та компромісів з енергією [4].

Найбільш розповсюдженими багатоядерні мобільні системи на базі асиметричної архітектури big.LITTLE, що розроблена компанією ARM Holdings. Ця технологія дозволяє поліпшити енергоефективність багатоядерного процесора та досягти 30 % енергозбереження в середньому шляхом об'єднання в одному чіпі [5]:

- економічних у плані енергоспоживання та більш повільних процесорних ядер (LITTLE, «маленькі»);

- ядер, що мають велику обчислювальну продуктивність, але більш вимогливі до ресурсів (big, «великі»).

Як правило, тільки один з кластерів є активним в певний момент часу. Також позитивною відмінністю є дуже швидке перенесення робочих навантажень між ядрами різних кластерів (big і LITTLE).

Так, у новітніх 8-ядерних процесорах (наприклад, Exynos 9820, Qualcomm® Snapdragon™ 855 за архітектурою ARM Cortex-A55 та A76, тощо) всі вісім ядер не беруть участь у координуваних навантаженнях. Зазначений чіпсет Snapdragon 855 складається з 4 «big» (з частотою до 2,85 ГГц за A76 для пікових навантажень) та 4 «LITTLE» ядер (з частотою 1,8 ГГц-за A55).

Також використовуються чіпсети з трикластерною архітектурою ARM DynamIQ, що складаються відповідно з двох ядер Exynos M4 (для виконання ресурсомістких завдань), двох ядер ARM Cortex-A75 (для оптимальної продуктивності) і чотирьох ядер Cortex-A55 – для виконання поточних завдань. Це дозволяє підвищити обчислювальну потужність пристрою та зменшити надмірні витрати енергії

Більш того, у трикластерній архітектурі реалізовано новий інтелектуальний планувальник завдань, який допомагає досягти балансу між продуктивністю і енергозбереженням, а також збільшити час автономної роботи.

У технології big.LITTLE завдання розміщуються на ядрах процесора в залежності від планувальника (англ. scheduler), реалізованого в CPU. Існують наступні три алгоритми розміщення задач по ядрах [6]:

- перемикання по кластерам;
- вбудований комутатор;
- неоднорідна багатопроцесорна обробка (рис. 1).

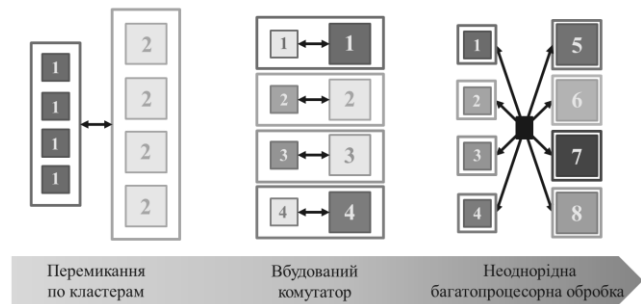


Рис. 1. Алгоритми розміщення задач по ядрах за технологією big.LITTLE

Термін роботи акумуляторної батареї мобільного пристрою можна подовжити вручну, керуючи апаратними компонентами, такими як GPS, 3G, Wi-Fi, Bluetooth, відключаючи їх, коли вони не використовуються. Однак, такий підхід ненадійний, оскільки користувачам складно постійно тримати на контролі використання різних функцій апаратних модулів, наприклад:

- стежити за тим, які компоненти повинні бути активними в даний час;

- контролювати, які функції необхідно відключати, щоб уникнути надлишкового витрати обмежених ресурсів мобільного пристрою.

Однак, функції енергозбереження, що почали вбудовувати в останні версії ОС, також не завжди доцільно використовувати. Часто трапляються такі ситуації, коли навпаки потрібно максимально використати ресурси мобільного пристрою, але при цьому отримувати актуальну інформацію щодо стану заряду акумулятора.

Слід зауважити, що існуючі моделі аналізу процесів у мобільних пристроях не підходять повною мірою для сучасних пристроїв на основі багатоядерних процесорів. Вони можуть давати великі похибки (до 34 %) і великі відхилення точності оцінки (понад 30 %) для різних типів робочих навантажень на багатоядерних смартфонах [7, 8].

Програмні засоби моніторингу (так звані «бенчмарки») – це найкращий спосіб проаналізувати, що відбувається зі смартфоном, коли виникає якась проблема. Тому для покращення ефективності використання мобільних пристроїв створюється велика кількість моніторингових систем, спеціалізованих кожна під окрему операційну систему. Для ОС Android серед них є такі, що використовуються найчастіше, і вони є легкодоступними в PlayMarket: Simple System Monitor, OS Monitor, CPU-Z, CPUMonitor, CPU/GPU Meter, System Monitor та інші. Подібні програми мають достатню кількість відмінностей в інтерфейсі, що впливає на їх використання, та різний функціонал, який відрізняється не тільки

між різними версіями застосунків, а й між платними та безкоштовними версіями.

Однак, в існуючих бенчмарках неналежна увага приділена відстеженню продуктивності роботи процесора з деталізацією навантаження кожного окремого ядра [9, 10]. Крім того, ідентичні системи, що містять номінально однакові процесори, можуть демонструвати значну різницю у споживанні енергії [11].

Також важливою проблемою є енергоспоживання вбудованими у мобільні пристрої компонентами та датчиками, які надають користувачам більше можливостей. Розвиток технологій смартфонів збільшив енергоспоживання цих пристроїв. Але використання смартфонів у деяких технологіях має здійснюватися багато днів без підзарядки. На жаль, батарею необхідно заряджати кілька разів на день при звичайному використанні. У багатьох випадках від смартфона вимагається забезпечення як продуктивності, так і енергоефективності. До того ж, треба обирати час (паузу) між опитуванням датчиків таким чином, щоб досягати конвергенції вбудованих та високопродуктивних обчислень.

Тому розробка енергоекономних систем ядерного моніторингу та моделювання енергоспоживання багатоядерного процесора клієнтського пристрою під час онлайн-з'єднання є актуальною задачею.

2. Розробка моделі для діагностики енергоспоживання багатоядерних процесорів

Процесор є основним джерелом енергоспоживання в мобільних пристроях. Моделювання енергоспоживання – це ключова технологія, що дозволяє зрозуміти витрати заряду акумулятора процесором, а також важливий інструмент управління живленням на пристрої.

Існуючі моделі енергоспоживання базуються на основі тільки частоти і використання CPU, але не враховують час, проведений в режимі очікування. Зазвичай вони використовують лінійну модель потужності процесора. Для кожної частоти процесора вони оцінюють енергоспоживання ядра процесора, як показано нижче:

$$P_{core} = \alpha_{freq_i} \cdot U_{core} + \beta_{freq_i}, \quad (1)$$

де U_{core} – навантаження ядра CPU, а α_{freq_i} і β_{freq_i} – два постійних параметра, значення яких визначаються за допомогою лінійної регресії в процесі генерації моделі.

Однак, існуючі моделі, що оцінюють енергоспоживання одноядерного процесора, не підходять для сучасних багатоядерних процесорів. Зокрема, тому що енергоспоживання CPU в 4-

ядерних (та більше) процесорах мобільних пристроїв з різними наборами мікросхем демонструє великий діапазон варіацій, навіть коли частота і навантаження CPU фіксовані.

Це було доведено наступним експериментом, у ході якого за допомогою спеціальної програми генератора робочого навантаження періодично виконувались безперервні обчислення з подальшим періодом простою на тестовому смартфоні. Керуючи співвідношення між періодами простою до періодами активного виконання обчислень, програма генератора робочого навантаження генерує робочі навантаження з різним навантаженням процесор мобільного пристрою.

Під час безперервних обчислень застосунок виконує складний цикл обчислення великого простого числа. Змінюючи кількість зайнятих циклів, можна контролювати тривалість кожного безперервного періоду обчислень. У результаті було виявлено, що регулюючи довжину кожного безперервного обчислення, енергоспоживання ядра CPU демонструє великий діапазон варіацій, навіть коли робоча частота ядер процесора та їх навантаження були фіксованими.

Отже основним недоліком існуючих моделей є те, що вони оцінюють енергоспоживання на основі лише тільки частоти і навантаження CPU без врахування часу, проведеного в стані простою, яке відіграє вирішальну роль у витраті енергії сучасними багатоядерними процесорами.

Для того щоб визначити, яка статистика станів простою процесора повинна використовуватися в якості змінної регресійної моделі, спочатку потрібно розглянути показник T_{C_i} , що відображає загальну тривалість часу, протягом якого ядро CPU знаходиться в стані простою C_i в секунду, коли частота f і коефіцієнт використання U фіксовані. Отже, якщо загальний час простою процесора в секунду становить T_{idle} , то від дорівнює:

$$T_{idle} = \sum_i T_{C_i}. \quad (2)$$

Проте, T_{C_i} не є хорошим показником енергоспоживання процесора, тому що зазвичай ядра процесора витрачають більше часу, залишаючись в глибоких станах простою.

Показник E_{C_i} , який представляє собою кількість записів для стану простою C_i в секунду. Порівняно з T_{C_i} , показник E_{C_i} означає менші енерговитрати на переходи від робочого стану до простою, може допомогти зекономити чимало

енергії. Однак, тільки E_{C_i} також не є хорошим предиктором енергоспоживання процесора, оскільки він не має прямого зв'язку з енергозбереженням за рахунок станів простою.

Тому далі слід приділити увагу середній тривалості входу для стану простою C_i , що позначається ED_{C_i} та визначається за виразом (3):

$$ED_{C_i} = \frac{T_{C_i}}{E_{C_i}}, \quad (3)$$

Зазвичай, ED_{C_i} вважається хорошим показником для моделювання енерговитрат пристрою, оскільки включає в себе як тривалість простою, так і витрати при переході з одного стану в інший. Однак, ED_{C_i} може мати похибки, що виникають від коливань. Такі коливання в свою чергу виникають через спорадичні записи стану очікування C_j , коли CPU переходить в стан C_i більшу частину часу.

Щоб уникнути зазначеної проблеми з показником ED_{C_i} , доцільно застосувати ваговий коефіцієнт w_i , який позначає той час, протягом якого процесор знаходиться в стані C_i протягом всього періоду простою. Таким чином формується середньозважена тривалість входу:

$$WED_{C_i} = w_i \cdot ED_{C_i}, \quad (4)$$

$$\text{де } w_i = \frac{T_{C_i}}{T_{idle}}.$$

Тоді енергоспоживання одного ядра процесора, що працює на частоті f , моделюється наступним чином:

$$P_{core} = \sum_i \beta_{C_i} \cdot WED_{C_i} + \beta_U \cdot U + c, \quad (5)$$

де WED_{C_i} – це середньозважена тривалість входу для стану простою; C_i , β_{C_i} і β_U – коефіцієнти WED_{C_i} і навантаження U ; c – константа.

Для кожної частоти f процесора, коефіцієнти та константа отримуються при виконанні лінійного регресійного аналізу даних, що містять різні T_{C_i} (показник загальної тривалості часу, протягом якого ядро CPU знаходиться в стані простою в секунду) та U , і відповідного P_{core} .

Далі модель енергоспоживання для одного ядра процесора може бути розширена до багатоядерного варіанту. Тоді приріст потужності на одне ядро обчислюється наступним чином:

ного варіанту. Тоді приріст потужності на одне ядро обчислюється наступним чином:

$$P_{core} = \frac{P_{CPU} - P_{BL,N_C}}{N_C}, \quad (6)$$

де N_C – кількість активних ядер, P_{BL,N_C} – основна потужність CPU з N_C активними ядрами, P_{CPU} – загальна потужність процесора.

Модель енерговитрат P_{CPU} багатоядерного процесора мобільних пристроїв формується наступним чином:

$$P_{CPU} = P_{BL,N_C} + \sum_i^{N_C} P_{\Delta,core,U_i,f_i}, \quad (7)$$

де $P_{\Delta,core,U_i,f_i}$ – приріст потужності на кожне i -те ядро CPU, коли воно працює на частоті f_i з навантаженням U_i . Для кожної частоти f_i значення $P_{\Delta,core,U_i,f_i}$ можна розрахувати, використовуючи описану вище модель енергоспоживання для одного ядра процесора. Тоді як P_{BL,N_C} – це постійне значення, яке можна заздалегідь виміряти.

3. Моделювання структури застосунку для діагностики енергоспоживання багатоядерних процесорів засобами CPN Tools

На основі опрацьованого матеріалу, було розроблено базову модель застосунку, що розробляється, за допомогою інструменту CPN Tools для моделювання та аналізу кольорових мереж Петрі (рис. 2 – рис. 3).

У початковому маркуванні активні два переходи «Takes resources» (обрати ресурси) та «Collects data» (збирати дані), які підсвічено зеленим кольором. Отже, резервування ресурсів процесора активними застосунками і збирання даних про поточний стан компонентів мобільного пристрою – паралельні події, і вони можуть відбуватися одночасно в будь-якому порядку.

«CPU» та «Battery» використовуються для забезпечення можливості спрацьовування переходів «Takes resources» і «Collects data». «Apps» використовується для забезпечення можливості спрацьовування переходу «Returns resources» (повернення ресурсів), щоб застосунки могли повернути ресурси процесора, які було зарезервовано для виконання певної дії.

Отже, застосунки можуть працювати за умови, що є вільні ресурси CPU та заряд батареї. Як тільки стає недостатньо ресурсів процесора, перехід «Takes resources» буде чекати, коли вони звільняться.

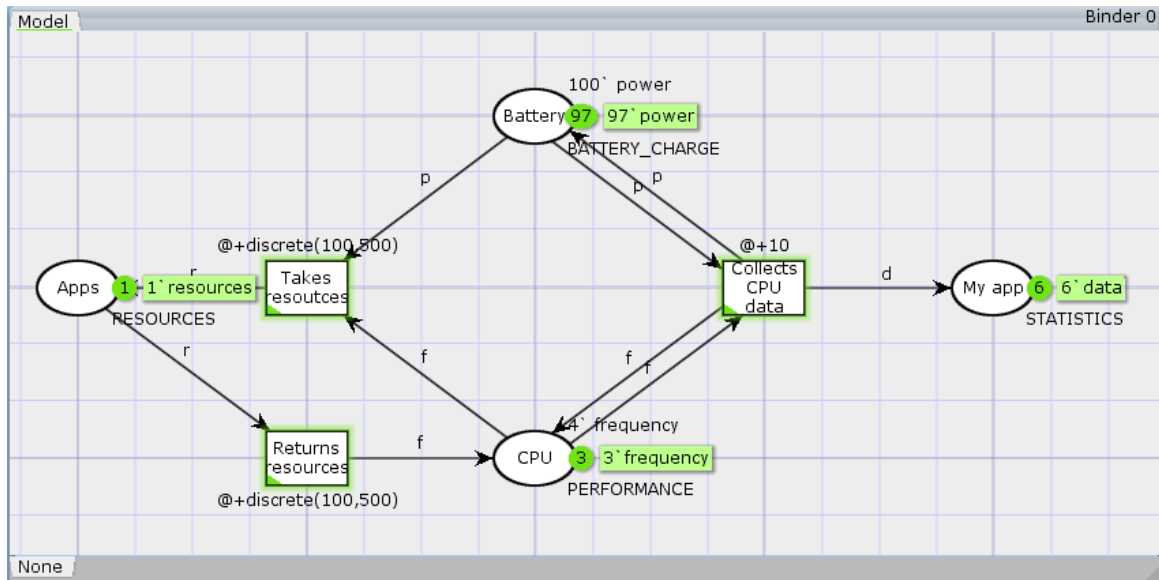


Рис. 2. Базова модель процесу роботи застосунку

```

▼colset BATTERY_CHARGE = unit with power;
▼colset PERFORMANCE = unit with frequency;
▼colset RESOURCES = unit with resources;
▼colset STATISTICS = unit with data;
▼var p: BATTERY_CHARGE;
▼var f: PERFORMANCE;
▼var r: RESOURCES;
▼var d: STATISTICS;

```

Рис. 3. Декларовані описи множин для моделі

У разі, якщо закінчився заряд батареї, розроблена модель зупиняється. Перехід «Collects data» буде виконуватися завжди, в незалежності від того, чи є вільні ресурси CPU, доки є заряд батареї. Тому у цьому випадку використовуються двонаправлені дуги. Решта дуг у моделі – односторонні.

Дуга, що з'єднує стан з переходом, витягує фішку відповідної множини кольорів. Фішка витягується відповідно до напису вхідної дуги переходу.

У представленій моделі всі написи представлені змінними відповідної множини. Так, біля вхідної дуги переходу «Returns resources» є напис «r»; «r» – змінна множини кольорів RESOURCES; таким чином беруться ресурси, які потім буде повернено до стану «CPU».

Вихідні дуги переходів створюють нові фішки. Нова фішка може бути створена (заново) або може збігатися з будь-якою фішкою, витягнутою з вхідної позиції. Усі переходи моделі створюють нові фішки. Наприклад, перехід «Takes resources» може виконатися при умові наявності двох різних фішок (power та frequency) на вхідних дугах. На вихідній дузі цього переходу створюється нова фішка типу «resources».

4. Розробка та тестування програмного застосунку для діагностики навантаження багатоядерного процесору мобільних об'єктів

Таким чином, в якості альтернативи існуючим бенчмаркам, був розроблений планувальник завдань ядра, який працює безпосередньо з ядрами і визначає, на якому з них виконується кожен процес/потік. Це потрібно, наприклад, для непарного компонування ядер (2 «великих» і 4 «малих»).

На рис. 4 наведений фрагмент програмного коду, на рис. 5 – одне з вікон інтерфейсу розробленого діагностичний застосунку для смартфонів під керуванням ОС Android 7 та вище з деталізацією використання кожного ядра.

Результати проведених досліджень показників навантаженості 8-ядерного обчислювального процесора, наведені у табл. 1. Дані у таблиці нормуються та відображаються у процентах (без символу %) таким чином, що усі значення у одному рядку повинні дорівнювати 100 %.

```

private RandomAccessFile reader;
reader = new
RandomAccessFile("/sys/device/system/cpu/cpu" + numCpuCore +
"/cpufreq/scaling_cur_freq", "r");

reader = new
RandomAccessFile("/sys/class/thermal/thermal_zone" + numZone +
"/temp", "r");

```

Рис. 4. Фрагмент коду відстеження частоти кожного ядра і показників термальних зон CPU



Рис. 5. Діагностичний програмний інтерфейс споживання енергії пристроєм

Найбільш важлива інформація знаходиться в першому рядку табл. 1, демонструючи суму показників навантаження CPU з урахуванням навантаженості кожного з ядер. Кожний наступний рядок починається з показників */cpuN* для кожного ядра процесора окремо.

Таким чином, у табл. 1 наведені значення кожного можливого стану 8 ядер процесора мобільних пристроїв.

Значення у стовпчику *User* – це час, витрачений на виконання на *un-niced* процеси користувача. Ця категорія належить до призначених для користувача процесів, які були запущені без явного пріоритету планування. Високе використання CPU в цій категорії може вказувати на неконтрольований процес.

Категорія *Nice* відображає час, витрачений на виконання процесів у режимі користувача «*niced*». Під *niced*-процесами розуміються процеси, які виконуються у пріоритеті, відмінному від пріоритету за замовчуванням. Як і *User*, це поле відображає завдання, не пов'язані з ядром. На відміну від *User*, пріоритет планування для цих завдань було встановлено за допомогою *Nice*.

Таблиця 1

Показники поядерної навантаженості пристрою на базі мобільного процесора

Core №	Cluster (thermal zone)	User	Nice	System	Idle	I/OWait	IRQ	Sort IRQ	Steal
cpu (avg value)	–	6,8	0,8	4,9	87,3	0,0	0,0	0,2	0
cpu0	0	10,7	1,2	9,2	77,7	0,4	0,0	0,8	0
cpu1		8,1	0,9	6,7	84,0	0,0	0,0	0,3	0
cpu2	1	8,9	1,0	6,0	83,9	0,0	0,0	0,2	0
cpu3		8,2	0,9	5,6	85,2	0,0	0,0	0,1	0
cpu4	2	2,0	0,4	1,1	96,5	0,0	0,0	0,0	0
cpu5		1,8	0,3	0,9	97,0	0,0	0,0	0,0	0
cpu6		2,1	0,3	0,9	96,7	0,0	0,0	0,0	0
cpu7		1,7	0,3	0,8	97,2	0,0	0,0	0,0	0

Процеси зі значенням *Nice* від 1 до 20 мають знижений пріоритет. Такі процеси, які споживають багато процесорного часу, як правило, не створюють проблем, тому що завдання з підвищеним пріоритетом отримують обчислювальну потужність своєчасно. Однак, якщо завдання з підвищеним пріоритетом (між мінус 1 і мінус 20) займають непропорційну кількість CPU, вони можуть легко вплинути на чуйність системи. Багато процесів з найвищим пріоритетом планування (мінус 19 або мінус 20 – залежно від системи) породжуються ядром для виконання важливих завдань, які впливають на стабільність системи.

System – час, витрачений на виконання у режимі ядра або в керуючому режимі. Відображає системні процеси. Більшість програм містять як користувальницькі компоненти, так і компоненти ядра. Коли ядро створює системні виклики, перевіряє привілеї або взаємодіє з пристроями від імені застосунку, тут відображається використання процесора. Коли процес виконує не ядром, він буде відображатися або в показнику *User*, або в *Nice*, якщо його пріоритет був заданий за допомогою *Nice*.

Стовпчик *Idle* демонструє час, витрачений на обробку простою ядра, тобто, коли жоден процес не виконується. Цей показник відображає від-

соток часу, протягом якого процесор був доступний, але простоював. Вважається, що система розумно використовує CPU, якщо сума *User*, *Nice* та *Idle* близька до 100 %.

Категорія *I/OWait* містить час очікування до завершення введення-виведення. Показник повідомляє, коли процесор почав операцію читання або запису і очікує завершення операції введення-виведення. Як і в стовбчику *Idle*, стрибки тут вважаються нормою. Але якщо показник повідомляє про часті або тривалі обробки якихось процесів, це може вказувати на «зависле» завдання.

Стовпчик *IRQ* демонструє час, що витрачається на обслуговування апаратних (фізичних) переривань.

SoftIRQ відображає час, витрачений на обслуговування програмних переривань, що відбуваються процесами, а не фізичними пристроями. На відміну від апаратних переривань, які відбуваються на рівні процесора, програмні переривання відбуваються на рівні ядра.

Категорія *Steal* містить час, який використовував гіпервізор. Значення *Steal* повідомляє, як довго віртуальний процесор очікує відповіді фізичного процесора, коли гіпервізор обслуговує свої завдання або інший віртуальний процесор. Зазвичай *Steal* в ОС Android дорівнює «0».

На підставі отриманих даних щодо навантаження процесора виконується обрахування навантаженості ядер у відсотках за наступним алгоритмом згідно з формулами (8)–(10):

$$\begin{aligned} \text{WorkTime} &= \text{User} + \text{Nice} + \text{System} + \\ &+ \frac{1}{I/oWait} + \text{IRQ} + \text{SoftIQR} + \text{Steal} +, \quad (8) \\ &+ \text{GuestTime} + \text{GuestNiceTime} \end{aligned}$$

$$\text{TotalTime} = \text{WorkTime} + \text{Idle}, \quad (9)$$

$$\text{Usage} = \text{WorkTime} / \text{TotalTime}. \quad (10)$$

Під час оптимізації пристрою також важливо відстежувати частоту роботи кожного ядра та інформацію з датчиків, які відстежують температуру процесора (див. рис. 4, б).

Одними з головних переваг розробленої програми-планувальника є більш тонкий контроль робочих навантажень, які переносяться між ядрами, їх рівномірний розподіл і резервування окремих ядер під ресурсомісткі завдання. Оскільки планувальник виконує безпосередню міграцію завдань між ядрами, накладні витрати ядра знижуються, і підвищується енергоефективність пристрою. Розроблене ПЗ придатне здійснювати моніторинг навантаження CPU та представляти зібрані дані у формі діаграм. Це дозволить користувачеві відстежувати певну частоту за певний проміжок часу та визначати, який програмний

застосунок найбільше напружує пристрій. Крім того, планувальник буде відстежувати температуру пристрою та його стан його батареї, оскільки важливо зберігати всі дані, що передаються.

Так, наприклад, якщо пристрій одночасно і здійснює відеозапис, і передає відеозапис через онлайн-з'єднання для узагальнення у «хмарні» базі даних, але при тривалому використанні телефон розряджається, то в цьому випадку велика кількість інформації може бути втрачена. В такій ситуації, при моніторингу планувальник розподілить навантаження на більш потужні – big – ядра для швидкої передачі та збереження інформації. Крім того, для мінімізації навантаження CPU, розроблена програма не «висить» постійно у треї, і не буде додатково навантажувати процесор.

Висновки

Розроблений програмний застосунок використовує нову модель енергоспоживання, оптимізовану для сучасних багатоядерних процесорів. Використання цієї моделі дозволяє збільшити час використання пристрою без підзарядки до 6,8%, за рахунок прийняття своєчасних рішень щодо керування навантаженням процесора (напр. відключення процесів та компонентів, які перевантажують ЦПУ або працюють у фоновому режимі).

Однією з головних переваг розробленого застосунку-моніторингу є більш тонкий контроль робочих навантажень, які переносяться між ядрами, їх рівномірного розподілу під час виконання ресурсомістких завдань, таких як проведення онлайн відеоконференцій чи використання спеціалізованих медичних застосунків, що здійснюють зберігання даних про пацієнтів у «хмарних» базах даних з шифруванням потоків даних. Оскільки планувальник виконує пряму міграцію завдань між ядрами, накладні витрати на ядро зменшуються, а енергоефективність пристрою збільшується.

Інша відмінність від існуючих застосунків полягає в тому, при використанні розробленого застосунку смартфон не перевантажується та не впливає на запущені процеси, тобто не гальмує їх. Це було перевірено виконанням 60 запусків підряд розробленого застосунку, при цьому проведений тест не привів до значного розрядження батареї. При проведенні подібного тесту для стандартного планувальника, вже при 15 запусках батарея смартфона розрядилася на 35 %.

Створений програмний застосунок здатен відстежувати навантаження на процесор (для кожного окремого ядра процесора та узагальнені результати) без використання діаграм. Такий підхід значно зменшує споживання й без того дефіцитної енергії мобільного пристрою та підвищує

стабільність роботи, що дозволяє, наприклад, забезпечити безперервне з'єднання з віддаленими користувачами (співробітниками офісів, пацієнтами на домашній реабілітації або учнів/студентів) у режимі відеоконференцій за допомогою сучасних спеціалізованих ресурсоємних застосунків та збільшити кількість одночасних підключень з одного інтернет-з'єднання, підключеного до провайдера.

Подяки

Автори дякують керівництву Чорноморського національного університету ім. Петра Могили та науковому керівнику НДР держ. реєстр. № 0121U109898 проф. Трунову О. М. за надану підтримку при підготовці матеріалів та за придбання обладнання.

Список використаної літератури

1. Trunov, A. Forming a methodology for transforming a model as the basis for expanding its informativeness [Text] / A. Trunov // *Eastern-European Jnl. of Enterprise Technol.* – 2019. – Vol. 5, no. 4 (101). – P. 34–43. DOI: 10.15587/1729-4061.2019.181866
2. Shurbin, V. Hardware-software complex to diagnostic and rehabilitation the patients with damages of cervical-thoracic spine and hand nerves [Text] / V. Shurbin, I. Zhuravska, V. Polianichkin et al. // *Electronics and Nanotechnology (ELNANO) : Proc. of the IEEE 40th Internat. Conf., Kyiv, Ukraine, Apr. 22–24, 2020.* – P. 493–498. DOI: 10.1109/ELNANO50318.2020.9088866
3. Trunov, A. The mesh technology for improving of end nodes updating for Lorawan network [Text] / A. Trunov, A. Macheniuk // *Computer Sci. and Information Technol. (CSIT) : Proc. of the IEEE 18th Internat. Sci. and Techn. Conf., Lviv, Ukraine, 11–14 Sept. 2018.* – P. 164–167. DOI: 10.1109/STC-CSIT.2018.8526649
4. Garcia, A. M. The impact of CPU frequency scaling on power consumption of computing infrastructures [Text] / A. M. Garcia, M. Da Silva Serpa, D. Griebler, et al. // *Lect. Notes in Computer Sci. (LNCS).* – Springer, LNCS, 2020. – Vol. 12254. – P. 142–157. DOI: 10.1007/978-3-030-58817-5_12
5. Se, W. L. Power-efficient Big.LITTLE core assignment scheme for task graph based real-time smartphone applications [Text] / W. L. Se, K. Donghoo, H. L. Sung, // *Mobile Internet Security.* Springer, 2020. – P. 73–84. DOI: 10.1007/978-981-15-9609-4_6
6. Vasilakis, E. Modeling energy-performance tradeoffs in ARM big.LITTLE architectures [Text] / E. Vasilakis, I. Sourdis, V. Papaefstathiou, et al. // *Power and Timing Modeling, Optimization and Simulation : Proc. of the IEEE 27th Internat. Symp., Thessaloniki, Greece, 25–27 Sept. 2017.* DOI: 10.1109/PATMOS.2017.8106950
7. Obukhova, K. Diagnostics of power consumption of a mobile device multi-core processor with detail of each core utilization [Text] / K. Obukhova, I. Zhuravska, V. Burenko // *Adv. Trends in Radioelectronics, Telecomm. and Computer Engin. (TCSET): Proc. of the IEEE 15th Internat. Conf., Lviv-Slavske, Ukraine, 25–29 Feb. 2020.* – P. 368–372. DOI: 10.1109/TCSET49122.2020.235456
8. Nikov, K. Evaluation of hybrid run-time power models for the ARM Big.LITTLE architecture [Text] / K. Nikov, J. Nunez-Yanez, M. Horsnell // *Embedded and Ubiquitous Computing (EUC) : Proc. of the IEEE 13th Internat. Conf. on, Porto, Portugal, 21–23 Oct. 2015.* – P. 205–210. DOI: 10.1109/EUC.2015.32
9. Ukhina, H. Express-analysis of stability of fourth order frequency-dependent rearrangeable components [Text] / H. Ukhina, V. Sytnikov // *Electronics and Information Technol. (EIT'2016) : Proc. of the IEEE Internat. Conf., Odessa, Ukraine, 23–27 May 2016.* – 3 p. DOI: 10.1109/ICEAIT.2016.7500998
10. Kataoka, H. Power consumption and computation models of a server with a multi-core CPU and experiments [Text] / H. Kataoka, D. Duolikun, T. Enokido, et al. // *Adv. Information Networking and Applic. Workshops (WAINA) : Proc. of the IEEE 29th Intern. Conf., Gwangju, Korea (South), 24–27 March, 2015.* – P. 217–222. DOI: 10.1109/WAINA.2015.127
11. Von Kistowski, J. Variations in CPU power consumption [Text] / J. Von Kistowski, H. Block, J. Beckett, et al. // *Performance Engin. : Proc. of the ACM Internat. Conf., Delft, Netherlands, 12–18 March 2016.* – 13 p. DOI: 10.1145/2851553.2851567

References

1. Trunov, A. (2019), "Forming a methodology for transforming a model as the basis for expanding its informativeness," *Eastern-European Jnl. of Enterprise Technol.*, vol. 5, no. 4 (101), pp. 34–43. DOI: 10.15587/1729-4061.2019.181866
2. Shurbin, V., Zhuravska, I., Polianichkin, V., Hnesdilov, M., Boiko, A., and Burenko, V. (2020), "Hardware-software complex to diagnostic and rehabilitation the patients with damages of cervical-thoracic spine and hand nerves," *IEEE 40th Internat. Conf. on Electronics and Nanotechnology (ELNANO)*, Kyiv, Ukraine, Apr. 22–24, 2020, pp. 493–498. DOI: 10.1109/ELNANO50318.2020.9088866

3. Trunov, A. and Macheniuk, A. (2018), "The mesh technology for improving of end nodes updating for Lorawan network," *IEEE 18th Internat. Sci. and Techn. Conf. on Computer Sci. and Information Technol. (CSIT)*, Lviv, Ukraine, Sept. 11–14, 2018, pp. 164–167. DOI: 10.1109/STC-CSIT.2018.8526649
4. Garcia, A. M., Da Silva Serpa, M., Griebler, D., et al. (2020), "The impact of CPU frequency scaling on power consumption of computing infrastructures," *In book: Lect. Notes in Computer Sci. (LNCS)*. Springer, LNCS, vol. 12254, pp. 142–157. DOI: 10.1007/978-3-030-58817-5_12
5. Se, W. L., Donghoo, K., and Sung, H. L. (2020), "Power-efficient Big.LITTLE core assignment scheme for task graph based real-time smartphone applications," *In book: Mobile Internet Security*. Nov. 2020, pp. 73–84. DOI: 10.1007/978-981-15-9609-4_6
6. Vasilakis, E., Sourdis, I., Papaefstathiou, V., Psathakis, A. and Katevenis, M.G.H. (2017), "Modeling energy-performance tradeoffs in ARM big.LITTLE architectures," *IEEE 27th Internat. Symp. on Power and Timing Modeling, Optimization and Simulation*, Thessaloniki, Greece, Sept. 25–27, 2017. DOI: 10.1109/PATMOS.2017.8106950
7. Obukhova, K., Zhuravska, I., and Burenko, V. (2020), "Diagnostics of power consumption of a mobile device multi-core processor with detail of each core utilization," *IEEE 15th Internat. Conf. on Adv. Trends in Radioelectronics, Telecomm. and Computer Engin. (TCSET)*, Lviv-Slavske, Ukraine, Feb. 25–29, 2020, pp. 368–372. DOI: 10.1109/TCSET49122.2020.235456
8. Nikov, K., Nunez-Yanez, J., and Horsnell, M. (2015), "Evaluation of hybrid run-time power models for the ARM Big.LITTLE architecture," *IEEE 13th Internat. Conf. on Embedded and Ubiquitous Computing (EUC)*, Porto, Portugal, Oct. 21–23, 2015, pp. 205–210. DOI: 10.1109/EUC.2015.32
9. Ukhina, H. and Sytnikov, V. (2016), "Express-analysis of stability of fourth order frequency-dependent rearrangeable components," *IEEE Internat. Conf. on Electronics and Information Technol. (EIT'2016)*, Odessa, Ukraine, May 23–27, 2016, 3 p. DOI: 10.1109/ICEAIT.2016.7500998
10. Kataoka, H., Duolikun, D., Enokido, T., and Takizawa, M. (2015), "Power consumption and computation models of a server with a multi-core CPU and experiments," *IEEE 29th Intern. Conf. on Adv. Information Networking and Applic. Workshops (WAINA)*, Gwangju, Korea (South), 24–27 March, 2015, pp. 217–222. DOI: 10.1109/WAINA.2015.127
11. Von Kistowski, J., Block, H., Beckett, J., et al. (2016), "Variations in CPU power consumption," *ACM Internat. Conf. on Performance Engin.*, Delft, Netherlands, March 12–18, 2016, 13 p. DOI: 10.1145/2851553.2851567

MODELING THE POWER CONSUMPTION OF A CLIENT DEVICE MULTICORE PROCESSOR DURING ONLINE INTERCONNECTION

I. M. Zhuravska, K. O. Obukhova, V. Yu. Savinov

Petro Mohyla Black Sea National University

Abstract. *During a pandemic, a large number of users are forced to work not at regular workplaces, but at home, using video conferencing and other online means of intercommunication. In this case, not only desktop computers but also mobile devices are often used. Wherein, for example, the application Zoom can diagnose insufficient connection speed, even if one session is established from one Internet connection. The paper shows that the reason may be the overload of the multicore processor's cluster on the client device. However, the functions of energy saving, embedded in the last actual versions of operation systems, also do not provide complete success. Often, it is needed maximum using the resource of the mobile application, but at the same, time it is necessary to keep up-to-date information about the battery charge. Increasing the number of cores in modern processors does not significantly improve the quality of communication in online communication, because the same type (multimedia) applications are still distributed to the same cluster. In this study, by modeling the power consumption of each core of such a processor, built on the big.LITTLE architecture, the results are achieved, according to which it is possible to adjust the power supply modes of a multi-core processor. The developed software application uses a new power consumption model optimized for modern multi-core processors. Using this model allows you to increase the time of use of the device without recharging to 6.8%, by making timely decisions about CPU load management (e.g., disabling processes and components that overload the CPU or run in the background). The positive difference between the developed software application and the existing schedulers is that this application does not overload client devices and does not affect running processes, i.e., does not slow them down. The power-efficient CPU loading*

on the client device will ensure the non-breaking of the communication session during video conferences providing processes in the production industries, the remote provision of educational and medical services.

Keywords: modeling the power consumption, CPN Tools, multicore CPU, online video conference.

МОДЕЛИРОВАНИЕ ЭНЕРГОПОТРЕБЛЕНИЯ МНОГОЯДЕРНОГО ПРОЦЕССОРА КЛИЕНТСКОГО УСТРОЙСТВА ПРИ ОНЛАЙН-СОЕДИНЕНИИ

И. Н. Журавская, Е. А. Обухова, В. Ю. Савинов

Черноморский национальный университет имени Петра Могилы

Аннотация. Во время пандемии большое количество пользователей вынуждена работать дома, с использованием видеоконференций и других онлайн-средств. При этом, например, приложение Zoom может диагностировать недостаточную скорость и не устанавливать связь. В работе показано, что причиной может быть перегрузка кластера многоядерного процессора (CPU) клиентского устройства. Путем моделирования энергопотребления каждым ядром CPU, построенного по архитектуре big.LITTLE, достигнуты результаты, на основании которых можно регулировать режимы энергопотребления CPU. Это позволит осуществлять непрерывные сеансы связи при видеоконференциях, обеспечивающих процессы в производственных отраслях, а также удаленное предоставление образовательных и медицинских услуг.

Ключевые слова: моделирование энергопотребления, CPN Tools, многоядерные процессоры, онлайн видеоконференции.

Отримано 10.05.2021



Журавська Ірина Миколаївна, Черноморський національний університет імені Петра Могили, доктор технічних наук, професор, в. о. професора кафедри комп'ютерної інженерії. Вул. 68 Десантників, 10, Миколаїв, 54003, Україна, E-mail: iryna.zhuravska@chmnu.edu.ua, тел. +38-067-912-34-57

Iryna Zhuravska, Petro Mohyla Black Sea National University, Dr. of Science (Techn.), Professor, Acting Professor of the Department of Computer Engineering, 68 Desantnykiv str., 10, Mykolaiv, 54003, Ukraine

ORCID ID: 0000-0002-8102-9854



Обухова Катерина Олександрівна, Черноморський національний університет імені Петра Могили, магістр з комп'ютерних наук, в. о. викладача кафедри комп'ютерної інженерії. Вул. 68 Десантників, 10, Миколаїв, 54003, Україна, E-mail: katherineobukhova@gmail.com, тел. +38-066-814-42-54

Kateryna Obukhova, Petro Mohyla Black Sea National University, Master of Computer Science, Professor, Acting Lecturer of the Department of Computer Engineering, 68 Desantnykiv str., 10, Mykolaiv, 54003, Ukraine

ORCID ID: 0000-0001-8793-7055



Савінов Володимир Юрійович, Черноморський національний університет імені Петра Могили, кандидат технічних наук, старший викладач кафедри комп'ютерної інженерії. Вул. 68 Десантників, 10, Миколаїв, 54003, Україна, E-mail: volodymyr.savinov@chmnu.edu.ua, тел. +38-093-998-47-16

Volodymyr Savinov, Petro Mohyla Black Sea National University, Ph.D. (Techn.), Senior Lecturer of the Department of Computer Engineering, 68 Desantnykiv str., 10, Mykolaiv, 54003, Ukraine

ORCID ID: 0000-0002-0862-5879