



В. С. Яковина, Б. В. Угриновський

Національний університет "Львівська політехніка", м. Львів, Україна

МЕТОД ОМОЛОДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID З ВИКОРИСТАННЯМ КОМПЛЕКСНОЇ МОДЕЛІ ЙОГО СТАРІННЯ НА ПІДСТАВІ ЛАНЦЮГА МАРКОВА

Явище старіння програмного забезпечення (ПЗ) полягає в нагромадженні помилок старіння у системі, що працює тривалий час без перезавантаження. Акумуляція помилок спричиняє зменшення швидкодії системи та збільшення відмов старіння, що має негативний вплив на такі характеристики якості ПЗ, як надійність та продуктивність. Мобільні пристрої та системи, зокрема на базі платформи Android, є особливо вразливі до процесу старіння, оскільки в них обмежені апаратні ресурси, їх функціонування залежить від заряду батареї, а активність використання користувачами може відрізнятися у різних випадках. Основним проактивним підходом для протидії негативним ефектам явища старіння є процедура омолодження програмного забезпечення, яка полягає в регулярному очищенні стану системи від нагромаджених помилок старіння. У роботі запропоновано метод омолодження програмного забезпечення для операційної системи (ОС) Android. Цей метод передбачає прогнозування часу виконання омолодження, використовуючи комплексну модель старіння та омолодження на підставі ланцюга Маркова з неперервним часом розподілу. Запропонований метод враховує різні рівні старіння, активність використання мобільного пристрою користувачем та рівень заряду батареї для прогнозування часу омолодження та вибору стратегії омолодження. Градація рівнів старіння в методі дає змогу виконувати планування омолодження тільки на тому етапі, коли в системі спостерігається процес старіння, який може призвести до відмови старіння. Окрім цього, залежно від рівня старіння в момент виконання омолодження, метод передбачає вибір між "теплим" і "холодним" омолодженням, що може забезпечити як ефективність омолодження, так і працездатність системи у разі "теплого" омолодження. Врахування рівня заряду в методі дає змогу планувати омолодження завчасно до повного розряду батареї. Для покращення досвіду користувача запропонований метод враховує активність використання мобільного пристрою користувачем так, щоб виконання омолодження не перешкоджало виконанню процесів і додатків користувача. Метод омолодження може бути реалізований у вигляді програмного комплексу, що складається із двох модулів. Модуль вимірювання метрик старіння дає змогу у фоновому процесі виконувати збір даних та їх статистичне опрацювання. Модуль омолодження безпосередньо виконує кроки методу та саму процедуру омолодження у запланований час, враховуючи отримані дані із модуля вимірювання. Описаний метод омолодження потребує експериментальних досліджень та перевірок ефективності омолодження на реальних даних і в реальних умовах.

Ключові слова: надійність програмного забезпечення; досвід користувача; чинники старіння; метрики старіння; мобільні пристрої, процедура омолодження програмного забезпечення.

Вступ / Introduction

Старіння ПЗ [1, 4, 9, 12] – це процес нагромадження помилок старіння [6] у стані системи, яка працює тривалий час без перезавантаження, що призводить до погіршення швидкодії ПЗ та збільшення інтенсивності відмов старіння [6]. Це явище має негативний вплив на такі характеристики якості [10] ПЗ, як надійність та продуктивність.

Старіння ПЗ спостерігається в багатьох ОС [6, 13], зокрема в мобільних платформах [2, 3, 17]. Широке використання мобільних пристроїв у повсякденному житті вимагає забезпечення високого рівня якості ПЗ мобільних ОС, зокрема Android.

Омолодження ПЗ [1, 4, 9, 13] – це проактивний підхід для протидії ефектам старіння [6], який полягає у виконанні регулярного запланованого очищення нагромаджених помилок шляхом перезавантаження всієї ОС чи її компонент. Мобільні пристрої мають деякі особливості [17], зокрема: залежність від заряду батареї, різна активність використання пристрою різними користувачами, обмежені апаратні ресурси, які відрізняють їх від інших програмно-апаратних систем, таких як персональні комп'ютери чи сервери.

Отже, актуальним науково-прикладним завданням є розроблення методів омолодження ПЗ, які враховували б чинники й умови старіння мобільних пристроїв. У цій роботі запропоновано метод омолодження ПЗ для ОС

Інформація про авторів:

Яковина Віталій Степанович, д-р техн. наук, професор, кафедра систем штучного інтелекту. Email: vitaliy.s.yakovyna@lpnu.ua;
<https://orcid.org/0000-0003-0133-8591>

Угриновський Богдан Володимирович, аспірант, кафедра програмного забезпечення. Email: bohdan.v.uhrynovskiy@lpnu.ua;
<https://orcid.org/0000-0002-4356-192X>

Цитування за ДСТУ: Яковина В. С., Угриновський Б. В. Метод омолодження програмного забезпечення для операційної системи Android з використанням комплексної моделі його старіння на підставі ланцюга Маркова. Науковий вісник НЛТУ України. 2021, т. 31, № 6. С. 97–103.

Citation APA: Yakovyna, V. S., & Uhrynovskiy, B. V. (2021). Software rejuvenation method for Android operating system considering complex aging model based on continuous time Markov chains. *Scientific Bulletin of UNFU*, 31(6), 97–103.

<https://doi.org/10.36930/40310615>

Android, який враховує такі особливості мобільних пристроїв, як стан заряду батареї, активність використання пристрою користувачем, а також різні рівні процесу старіння, які забезпечують вибір оптимальної стратегії і механізму виконання омолодження.

Об'єкт дослідження – процес старіння та омолодження ПЗ.

Предмет дослідження – метод омолодження ПЗ для ОС Android, який враховує чинники й умови старіння мобільних пристроїв.

Мета роботи – розробити метод омолодження ПЗ для протидії явищу старіння в мобільній ОС Android, який враховуватиме такі чинники, як активність використання мобільного пристрою, стан заряду батареї та різні рівні старіння ПЗ.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- проаналізувати наявні метрики старіння, методи омолодження та моделі процесу старіння для мобільної ОС Android;
- описати комплексну модель старіння та омолодження на підставі ланцюга Маркова з неперервним часом розподілу, яка враховує активність використання мобільного пристрою, стан заряду батареї та різні рівні старіння;
- описати інтерфейс та функції модуля вимірювання метрик старіння, зокрема, правила інтерпретації статистичних даних у відповідні стани системи, передбачені комплексною моделлю;
- розробити метод омолодження ПЗ, який використовує комплексну модель та статистичні дані модуля вимірювання метрик для прогнозування часу омолодження, а також виконує вибір стратегії та механізму омолодження.

Наукова новизна отриманих результатів дослідження – вперше розроблено метод омолодження ПЗ для ОС Android з урахуванням активності користувача мобільного пристрою, стану заряду батареї та різних рівнів старіння, а прогнозування часу омолодження виконується з допомогою комплексної моделі на підставі ланцюга Маркова з неперервним часом розподілу.

Практична значущість результатів дослідження – можливе розроблення ПЗ на підставі запропонованого методу, яке дасть змогу зменшити негативний вплив процесу старіння на надійність і продуктивність мобільної системи, а врахування таких чинників, як активність використання та рівень старіння системи покращить досвід користувача UX (англ. – user experience).

Аналіз останніх досліджень та публікацій. Вимірювання метрик старіння [6] дає змогу визначати наявність старіння у системі, а оцінювання трендів – визначати інтенсивність старіння. У попередніх роботах [2, 20] теоретично описано та емпірично обґрунтовано доцільність застосування таких метрик старіння в ОС Android, як:

- Activity Launch Time (ALT) – тривалість відображення екрану додатку з моменту ініціалізації процесу до моменту його першої появи на дисплеї мобільного пристрою;
- Frame Draw Time (FDT) – тривалість відображення кадрів користувацького інтерфейсу додатку;
- Proportional Set Size (PSS) – об'єм оперативної пам'яті використаної процесами;
- Garbage Collector Duration (GCD) – тривалість виконання процедури збирання сміття.

Метрики ALT та FDT дають змогу оцінити продуктивність і погіршення швидкодії користувацьких додатків. ALT найбільш часто використовується в дослідженнях, однак її вимірювання може не відображати реальний стан старіння у системі, тому в роботі [20] бу-

ло запропоновано метрику FDT, яка може бути використана як для оцінки наявності старіння в системі, так і для оцінки активності використання мобільного пристрою користувачем.

Метрики PSS та GCD, водночас, є індикаторами старіння у системних процесах, які можуть бути використані для точнішої оцінки старіння у поєднанні із метриками продуктивності користувацького інтерфейсу. Результати емпіричних досліджень старіння ПЗ з допомогою вимірювань метрик старіння підтверджують наявність старіння у системі Android [3]. У роботах [2, 3, 18] визначено найвразливіші до старіння системні процеси, де метрики PSS і GCD показують постійні тренди збільшення в більшості з виконаних експериментів. Зокрема, системний процес system server – це Java процес, який запускається під час завантаження Android, ініціалізує Android Framework, який містить основні системні служби, такі як Activity Manager та Package Manager. Цей процес також регулює доступ до системних ресурсів.

У роботі [18] розглянуто оцінку "out of memory" (oom_adj_score) для пріоритетизації кожного процесу системи, яка використовується механізмом low-memory killer (LMK), який примусово завершує процеси з найнижчим пріоритетом у разі недостатності пам'яті у критичних ситуаціях. У контексті старіння ця метрика дає змогу оцінювати активність користувача і доцільність застосування омолодження в конкретний момент часу. Зокрема, важливо враховувати частку процесів з пріоритетом групи користувацьких додатків та процесів, виконання яких є важливим для користувача, а їх переривання виконанням омолодження може погіршити надійність ПЗ. Ще одним прикладом використання метрики oom_adj_score є в роботі [5], де для визначення рівня робочого навантаження обчислюють частку користувацьких процесів, що активно використовуються.

ADARTA [5] – це інструмент виявлення старіння та виконання омолодження для ОС Android. ADARTA реалізує метод рухомого вікна для відслідковування і виявлення старіння ПЗ в реальному часі. У разі виявлення старіння, метод виконує прогнозування часу виконання процедури омолодження шляхом обчислення часу до відмови через старіння (англ. – Time To Aging Failure, TTAF) [6]. TTAF обчислюється з допомогою тесту Манна-Кандела та процедури Сена на підставі часових рядів для вимірюваних метрик та їх порогових значень. Недоліком такого підходу виявлення старіння є потреба постійного використання системних ресурсів для виконання обчислень, а обчислення TTAF може не враховувати інші чинники, що впливають на процес старіння. Вирішенням цих недоліків може бути застосування аналітичних моделей [4], які дають змогу моделювати процес старіння і омолодження на підставі статистичних даних вимірюваних метрик старіння. Аналітичні моделі дають змогу уникнути необхідності безперервного аналізу часових рядів, зокрема можуть бути застосовані для раннього прогнозування часу виконання омолодження відразу після запуску мобільного пристрою.

У роботах [15, 16, 19] наведено аналітичні моделі старіння та омолодження для ОС Android на підставі ланцюгів Маркова з неперервним часом розподілу, які враховують різні чинники старіння.

У роботі [15] описано модель, яка враховує активність використання мобільного пристрою користува-

чем. Активність користувача може бути подана двома станами системи, а саме "Active" та "Sleep", стан активного використання мобільного пристрою та стан очікування відповідно. Особливістю цієї моделі є те, що вона описує поведінку системи, де омолодження може почати виконуватись тільки в стані "Sleep", оскільки в режимі очікування виконання процедури омолодження може найменше нашкодити користувачу.

У попередній роботі запропоновано модель [16] для Android з урахуванням активності користувача, яка розширена додатковими станами, що описують три рівні заряду батареї: "Stable Power" (високий рівень заряду), "Low Power" (критично низький рівень заряду батареї), "Off Power" (батарея повністю розряджена). Особливість цієї моделі в тому, що вона передбачає можливість виконання процедури омолодження тільки у стані "Stable Power", оскільки в двох інших станах виконання процедури неможливе або недоцільне.

У моделі старіння та омолодження для ОС Android [19] запропоновано врахувати додаткові рівні старіння, які дали б змогу оптимізувати планування омолодження з точки зору UX та надійності ПЗ. Ця модель описує систему, що може перебувати в таких станах:

- Young – для системи характерні високий рівень продуктивності та низький рівень використання системних ресурсів;
- Aging – у системі спостерігаються тренди погіршення продуктивності й збільшення використання системних ресурсів, а вимірювані метрики розташовані в межах між верхнім граничним значенням "Young" та нижнім значенням порогу переходу в "Old";
- Old – користувач відчуває затримки в UI, а обмеження системних ресурсів може призводити до збоїв в окремих користувацьких додатках, вимірювані метрики перевищують визначені порогові значення;
- Failure – відмова системи внаслідок старіння, яка призводить до перезавантаження пристрою користувачем під час активного використання або системою в режимі очікування;
- Rejuvenation – стан, в якому відбувається виконання процедури омолодження.

Аналіз тестових симуляцій цієї моделі показав, що ефективним є виконання омолодження у стані "Aging", тобто тоді, коли старіння вже спостерігається в системі, але ймовірність відмови через старіння ще невелика.

Окрім цього, ця модель враховує "холодну" та "теплу" техніку омолодження [7]. Виконання процедури "холодного" омолодження полягає в перезавантаженні цілої ОС, що з найбільшою ймовірністю поверне її в стан "Young". Водночас, "тепле" омолодження може бути реалізовано шляхом перезавантаження окремих компонент чи ключових системних процесів, у яких спостерігається старіння. Такий підхід, на відміну від "холодного" омолодження, може бути менш ефективним, але він дає змогу зберегти систему в працездатному стані без виконання перезавантаження. Оскільки ці техніки мають різний рівень ефективності з точки зору омолодження, тому в межах моделі [19] "тепла" техніка може бути застосована на етапі "Aging" для повернення у стан "Young", а "холодне" омолодження може бути застосовано як у стані "Aging", так і в стані "Old". Також, враховуючи вплив техніки виконання омолодження на працездатність системи, у моделі із врахуванням активності використання мобільного пристрою передбачено умову, що під час "холодного" омолодження користувач не може використовувати пристрій, тоді як під час "теплого" омолодження може відбуватись перехід зі стану "Sleep" у стан "Active" і навпаки.

Матеріали та методи дослідження. В основі методу омолодження, який пропонуємо в цій роботі, знаходиться комплексна модель старіння та омолодження на підставі ланцюгів Маркова з неперервним часом розподілу, яка враховує різні рівні старіння [19], активність використання мобільного пристрою користувачем [15], а також стан заряду батареї [16]. Модель описує систему, яка може перебувати в таких станах:

- Active Mobile + Stable Power + Young (AmSpY);
- Active Mobile + Stable Power + Aging (AmSpA);
- Active Mobile + Stable Power + Old (AmSpO);
- Active Mobile + Stable Power + Failure (AmSpF);
- Active Mobile + Stable Power + Rejuvenation (AmSpR);
- Sleep Mobile + Stable Power + Young (SmSpY);
- Sleep Mobile + Stable Power + Aging (SmSpA);
- Sleep Mobile + Stable Power + Old (SmSpO);
- Sleep Mobile + Stable Power + Failure (SmSpF);
- Sleep Mobile + Stable Power + Rejuvenation (SmSpR);
- Active Mobile + Low Power + Young (AmLpY);
- Active Mobile + Low Power + Aging (AmLpA);
- Active Mobile + Low Power + Old (AmLpO);
- Active Mobile + Low Power + Failure (AmLpF);
- Active Mobile + Low Power + Rejuvenation (AmLpR);
- Sleep Mobile + Low Power + Young (SmLpY);
- Sleep Mobile + Low Power + Aging (SmLpA);
- Sleep Mobile + Low Power + Old (SmLpO);
- Sleep Mobile + Low Power + Failure (SmLpF);
- Sleep Mobile + Low Power + Rejuvenation (SmLpR);
- Off Power (Op).

Оскільки процес старіння та омолодження поданий моделлю на підставі ланцюга Маркова [11], то ймовірність перебування системи в i -му стані $P_i(t)$ можна отримати з результату обчислення системи рівнянь Колмогорова-Чепмена:

$$\frac{dP_i(t)}{dt} = -\sum_{j \in S} a_{ij}P_i(t) + \sum_{j \in S} a_{ji}P_j(t), i \in S, \quad (1)$$

де: a_{ij} та a_{ji} – інтенсивності переходів зі стану i у стан j та навпаки, а S – множина всіх можливих станів системи.

Формулу (1) можна подати у матричному вигляді для обчислення вектора ймовірностей перебування системи в кожному із можливих станів у довільний час t :

$$P(t) = P(0) \cdot \exp(Q \cdot t), \quad (2)$$

де: Q – матриця інтенсивностей переходів між станами, а $P(0)$ – початковий вектор ймовірностей.

Оскільки розподіл часу в моделях на підставі ланцюгів Маркова з неперервним часом є експонентним, то середня тривалість переходів між станами обчислюється за такою формулою:

$$T_{avg} = 1/a. \quad (3)$$

Враховуючи попередні результати досліджень та характеристики можливих станів системи, виконання процедури омолодження буде доцільним за умови, коли система перебуває у станах "Sleep", "Stable Power" та "Aging". Тобто процедура омолодження у стані SmSpA не буде мати значного негативного впливу на UX, а також випереджатиме перехід пристрою у вразливий до відмов старіння стан чи до розряду батареї. Виконання процедури омолодження в комплексній моделі описано переходом зі стану SmSpA у стан SmSpR з інтенсивністю a_{AR} (середнім часом T_{AR}). З цього випливає, що прогнозування оптимального часу омолодження полягає у пошуку такого значення $a_{AR}(T_{AR})$, за якого система з найбільшою ймовірністю перебуває в стані SmSpA у момент виконання процедури. Цільова функція пошуку оптимального часу виконання омолодження:

$$\begin{cases} \sum_{i \in S_A} P_i(t) \rightarrow \max; \\ \sum_{j \in S_{FR}} P_j(t) \rightarrow \min; \\ P_{SmSpA}(t) \rightarrow \max; \end{cases} \quad (4)$$

де: S_A – множина працездатних станів системи (AmSpY, AmSpA, AmSpO, SmSpY, SmSpA, SmSpO, AmLpY, AmLpA, AmLpO, SmLpY, SmLpA, SmLpO); S_{FR} – множина непрацездатних станів системи (AmSpF, AmSpR, SmSpF, SmSpR, AmLpF, AmLpR, SmLpF, SmLpR, Op).

Результати дослідження та їх обговорення / Research results and their discussion

Компоненти методу омолодження ПЗ. Для реалізації запропонованого методу омолодження в ОС Android необхідно передбачити комплекс програмних модулів, які забезпечили б виконання методу омолодження, моніторинг процесу старіння у системі та прогнозування оптимального часу виконання омолодження. На рис. 1 показано діаграму компонент методу омолодження для ОС Android, який складається із двох модулів: Aging Metrics Monitor та Rejuvenation Method Module.

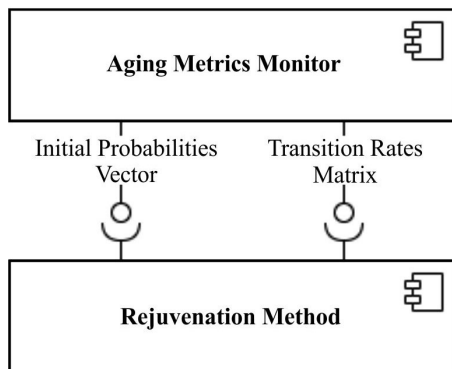


Рис. 1. UML-діаграма компонент методу омолодження ОС Android / UML component diagram of rejuvenation method for Android OS

Модуль вимірювання метрик старіння. Для забезпечення роботи методу омолодження важливим компонентом програмної системи є модуль вимірювання метрик старіння. Основним завданням цього модуля є збір метрик інтерфейсу та системи для формування часових рядів, на підставі яких можна обчислювати інтенсивності переходів між станами, а також визначати стан системи в конкретний момент часу.

Налаштування цього модуля можна описати набором множин, зокрема, для ОС Android:

$M_{Aging} = \{ALT, FDT, PSS, GCT\}$ – метрики старіння;

$M_{Usage} = \left\{ \begin{array}{l} ALT, FDT, \\ \text{частка процесів користувача,} \\ \text{частка використання CPU,} \\ \text{частка суми PSS} \end{array} \right\}$ – метрики активності використання мобільного пристрою;

$M_{Battery} = \left\{ \begin{array}{l} \text{поточний рівень заряду,} \\ \text{прогнозований час розряду} \end{array} \right\}$ – метрики стану заряду батареї;

$M_{Time} = \{1 \text{ хв.}, 5 \text{ хв.}, 10 \text{ хв.}, 30 \text{ хв.}, 60 \text{ хв.}\}$ – часові інтервали між вимірюваннями метрик.

Вибір оптимального інтервалу між вимірюваннями з набору M_{Time} є важливим завданням, що потребує по-

дальших досліджень, оскільки малий інтервал між записами вимагає виділення більшого об'єму пам'яті в сховищі даних, однак надто великий інтервал може негативно вплинути на точність моделювання під час прогнозування часу омолодження.

Сформовані множини часових рядів вимірюваних метрик необхідні для визначення інтенсивностей переходів між можливими станами системи згідно з моделлю прогнозування часу омолодження стану, а саме – для обчислення значень матриці Q у формулі (2). У табл. 1-3 наведено умови значень вимірюваних метрик, за яких можливе перебування в тому чи іншому стані.

Табл. 1. Метрики та їх пороги для визначення рівня старіння системи / Metrics and their thresholds for determining the level of the system aging

Метрика		Стан системи старіння			
Назва	Пріоритет	Young	Aging	Old	Failure
FDT	Високий	≤ 16 мс	> 16 мс & ≤ 40 мс	> 40 мс & ≤ 66 мс	> 66 мс
PSS	Високий	≤ 20 %	> 20 % & ≤ 80 %	> 80 % & ≤ 95 %	> 95 %
ALT	Середній	< 2 с	≥ 2 с & < 5 с	≥ 5 с & < 10 с	≥ 10 с
GCT	Середній	≤ 100 мс	> 100 мс & ≤ 200 мс	> 200 мс & ≤ 300 мс	> 300 мс

Для обчислення інтенсивностей переходів між станами старіння (див. табл. 1) доцільно застосовувати методи для виявлення наявного позитивного тренду для кожного окремого часового ряду, який описує поведінку мобільного пристрою від моменту його увімкнення до вимкнення. Таким методом може бути тест Манна-Кандела. Якщо тренд існує, тоді, застосувавши регресійний метод, можна отримати криву, перетин якої із пороговими значеннями буде давати часові відрізки перебування системи в кожному із станів. Обчисливши середні значення для всіх отриманих часових відрізків та застосувавши формулу (3), можна отримати інтенсивності переходів. Водночас, виставлені пріоритети для метрик необхідні, щоб відкинути ті випадки, коли старіння спостерігається тільки в метриках з низьким пріоритетом. Наприклад, якщо тренд старіння спостерігається тільки в одній із метрик із середнім рівнем, тоді цей випадок можна відкинути і не враховувати в обчисленнях інтенсивностей.

Табл. 2. Метрики та їх пороги для визначення рівня активності використання мобільних пристроїв / Metrics and their thresholds for determining the level of mobile device usage activity

Метрика	Активність використання мобільного пристрою		
	Варіа	Sleep	Active
Назва			
FDT	1,0	0	> 0
ALT	0,5	0	> 0
Частка процесів користувача	0,5	< 20 %	≥ 20 %
Частка використання CPU	0,25	< 20 %	≥ 20 %
Частка суми PSS	0,25	< 20 %	≥ 20 %

Табл. 3. Метрики та їх пороги для визначення стану заряду батареї / Metrics and their thresholds for determining the state of battery charge

Метрика	Стан заряду батареї	Стан заряду батареї		
		High Power	Low Power	Off Power
Назва	Варіа			
Прогнозований час розряду	1,0	> 1 год	≤ 1 год	0
Поточний рівень заряду	0,25	> 20 %	≤ 20 %	0 %

У випадку метрик та порогів активності використання (див. табл. 2) та стану заряду батареї (див. табл. 3) для обчислення матриці інтенсивностей потрібно враховувати частоту змін їх станів упродовж всього циклу функціонування мобільного пристрою. Виставлені ваги метрик дають змогу робити вибір конкретного стану під час вимірювання на користь того, який має найбільшу суму ваг. Наприклад, якщо для визначення рівня активності вимірювані метрики FDT вказують на стан "Active", а решта на стан "Sleep", тоді вибір буде на користь "Sleep", бо сума ваг для цього стану дорівнює 1,5 і більша за вагу FDT. Цей приклад може описувати випадок, коли користувач активував екран телефону і пристрій відобразив певний додаток, однак обчислень не відбулось і можливо, що в наступний момент часу користувач перестане використовувати телефон. Якщо користувач продовжить його використовувати, то використання ресурсів систем виросте, що буде відображено у вимірюваних метриках.

Ще одним важливим завданням модуля моніторингу є визначення поточного стану системи. Для визначення

поточного рівня старіння необхідно обчислити середнє значення для N останніх записів часового ряду метрик M_{Aging} і визначити в який із інтервалів табл. 1 вони входять. Для визначення поточного статусу батареї й активності використання необхідно отримати актуальні значення метрик на цей момент. У такий спосіб буде сформований вектор $P(0)$ і використаний для обчислення формули (2).

Модуль методу омолодження. Виконання методу омолодження передбачається в окремому модулі, який може бути поданий у вигляді системного фонового процесу, що ініціалізується під час запуску мобільного пристрою. Загальну схему методу омолодження подано у вигляді UML діаграми діяльності (рис. 2).

На першому кроці методу омолодження програмного забезпечення необхідно виконати ініціалізацію інтенсивностей матриці переходів для виконання обчислень моделі. Цей крок використовує інтерфейс модуля моніторингу для обчислення матриці інтенсивності переходів Q .

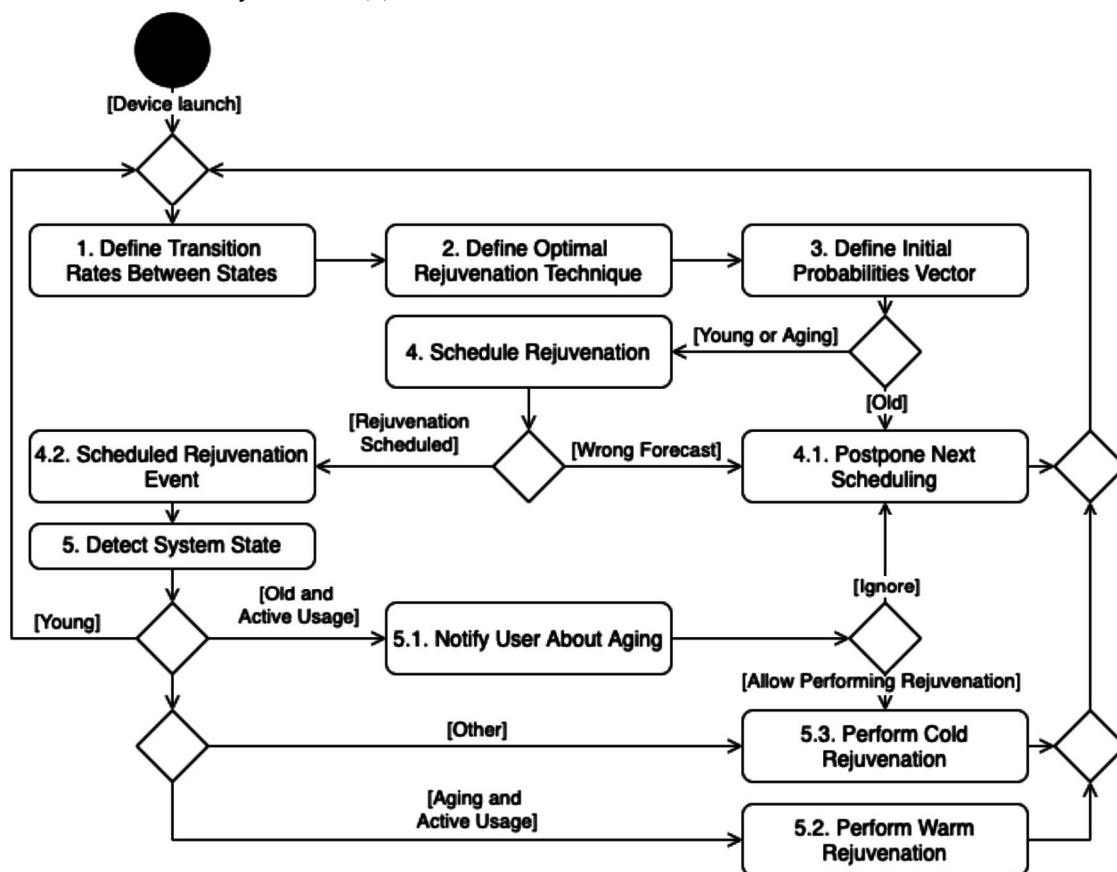


Рис. 2. UML-діаграма діяльності методу омолодження / UML-diagram of the rejuvenation method

Другий крок методу полягає у визначенні оптимальної для користувача техніки виконання омолодження під час прогнозування часу омолодження. Тобто для комплексної моделі необхідно установити інтенсивність переходу між станами "Active" та "Sleep" під час виконання омолодження так, щоб за умови активного використання мобільного пристрою в пріоритеті було застосування "теплого" омолодження:

$$a_{SmAmR} = a_{AmSmR} = 0, \text{ якщо } a_{Am} > a_{Sm};$$

$$a_{SmAmR} = a_{SmAm}, a_{AmSmR} = a_{AmSm}, \text{ якщо } a_{Am} \leq a_{Sm}.$$

На третьому кроці виконується ініціалізація початкового вектора імовірностей $P(0)$ з використанням ін-

терфейсу модуля моніторингу. Цей вектор однозначно визначає стан в якому перебуває система в цей момент.

Виконання кроків 1-3 можливе або після увімкнення мобільного пристрою, або після завершення виконання запланованої процедури омолодження. Ці дві умови означають, що на кроці 3 очікується перебування системи у стані "Young" або "Aging". Якщо система все ж таки перебуває в стані "Old", тоді виконання, аналогічно як і планування, омолодження недоцільне, тому відбувається перехід до кроку 4.1. Така ситуація можлива в умовах, коли мобільний пристрій постійно перебуває у стані "Old" через помилки, не пов'язані із явищем старіння програмного забезпечення.

Якщо система перебуває в одному із станів "Young" чи "Old", тоді виконується крок 4, де обчислюється цільова функція (4) комплексної моделі для визначення оптимального часу омолодження з урахуванням вхідних даних, отриманих на кроках 1-3. Для прогнозованого часу омолодження необхідно передбачити порогові значення, вихід за які означає недостовірність отриманого результату, оскільки модель прогнозування залежить від статистичних даних, а якість статистичних даних – від нагромаджених даних.

Якщо прогнозований час омолодження $T_R < T_{RBottom}$, або $T_R > T_{RTop}$, то наступним потрібно виконувати крок 4.1, який полягає в очікуванні наступної спроби прогнозування омолодження певний час $T_{Postpone}$. Наприклад, якщо $T_{RBottom} = 30$ хв, а $T_R = 16$ хв, тоді омолодження може бути надлишковим і мати більш негативний ефект. Водночас, якщо $T_{RTop} = 7$ днів, а $T_R = 10$ днів, тоді є ймовірність пропустити реальний момент переходу в стан "Old" і саму відмову старіння. Визначення коректних і ефективних значень $T_{RBottom}$, T_{RTop} та $T_{RPostpone}$ потребує додаткових досліджень, зокрема ці значення можуть бути адаптовані під конкретні умови середовища.

Якщо прогнозований час омолодження коректний, тоді виконується крок 4.2, тобто очікується запланована подія початку виконання процедури омолодження. Ця подія ініціалізує виконання п'ятого кроку, який використовує інтерфейс модуля моніторингу старіння і визначає стан системи для прийняття рішення про доцільність виконання процедури омолодження.

Якщо система перебуває у стані "Young", тоді виконання омолодження недоцільне і виконується повернення до першого кроку і повторного планування наступної процедури омолодження. Якщо система перебуває в стані "Old" і користувач активно використовує мобільний пристрій, то процедура "теплого" омолодження буде неефективна, а "холодного" може пошкодити користувачу, тому пропонується в цей момент показати повідомлення з інструкцією про потреба перезавантажити мобільний пристрій, тобто виконати "холодне" омолодження. Якщо система перебуває в стані "Aging" і користувач активно використовує мобільний пристрій, то застосовується процедура "теплого" омолодження. В усіх інших випадках виконується "холодне" омолодження.

Після виконання омолодження будь-якого типу, метод передбачає, що система повернеться до стану "Young", тому важливим завданням є розроблення ефективних і коректних реалізацій методів "холодного", а особливо "теплого" омолодження.

Обговорення результатів дослідження. Запропонований метод омолодження ПЗ враховує такі чинники, що мають прямий чи опосередкований вплив на процес старіння. Врахування активності використання пристрою користувачем дає змогу покращити характеристики UX та надійності, оскільки зменшується ймовірність того, що виконання процедури омолодження перериватиме виконання важливих для користувача процесів та додатків. Планування омолодження з урахуванням інтенсивності розряду батареї дає змогу виконувати омолодження завчасно, до повного розряду батареї, що зменшує ймовірність поганих прогнозів. Подання процесу старіння у вигляді поступових рівнів у моделі та методи дають змогу виконувати прогнозування часу омолодження програмного забезпечення тільки за умови,

що в системі спостерігається процес старіння, однак це ще не призводить до відмов старіння. Окрім цього, залежно від конкретних умов старіння, в яких перебуває система, метод передбачає вибір механізму омолодження, що забезпечує ефективність його виконання та покращує процес старіння та омолодження з точки зору UX.

Ключовими елементами методу, що потребують емпіричної перевірки, є інтерпретація вимірюваних метрик старіння для визначення стану старіння системи та інших станів моделі, а також оптимізація змінних і функцій, що впливають на прогнозування часу виконання омолодження. Водночас, ефективність виконання процедури омолодження визначається його механізмом та прогнозованим часом, що впливає на зміни у вимірюваних метриках. Тобто важливими завданнями майбутніх досліджень є експериментальні перевірки ефективності запропонованого методу омолодження та вибір оптимальних параметрів для конкретних умов та сценаріїв використання мобільного пристрою.

Висновки / Conclusions

У цій роботі запропоновано й описано метод омолодження ПЗ для ОС Android, який використовує комплексну модель старіння та омолодження на підставі ланцюгів Маркова з неперервним часом розподілу для прогнозування часу виконання процедури омолодження. Модель старіння та омолодження в цій роботі – це інструмент для проектування та вибору параметрів методу омолодження ПЗ, що дає змогу формувати технічне завдання на його розроблення. Запропонована модель старіння та омолодження може бути застосована для оцінки показників якості наявного ПЗ. Комплексна модель дає змогу отримувати вирази для показників ефективності омолодження ПЗ.

Метод, аналогічно як і модель, визначає поступові рівні старіння системи, що дає змогу будувати стратегії омолодження, які враховують актуальний стан старіння системи та попереджати перехід системи у стан з високою ймовірністю відмови внаслідок старіння.

Окрім цього, запропонований метод прогнозування враховує такі чинники, як стан заряду батареї та активність використання мобільного пристрою користувачем, що дає змогу покращити UX. Наприклад, у стані активного використання мобільного пристрою його робота не буде перервана процедурою омолодження програмного забезпечення, а користувач отримає повідомлення, що необхідно виконати омолодження для покращення продуктивності системи.

Метод передбачає можливість застосування механізмів "холодного" та "теплого" омолодження для різних умов старіння. Зокрема, використання "теплого" омолодження на ранньому етапі дає змогу уникнути переходу в стан "Old" та необхідності перезавантаження мобільної системи у разі холодного омолодження, цим самим покращується як UX, так і характеристики надійності.

У наступних роботах важливо виконати реалізацію запропонованого методу для ОС Android, перевірити запропоновані стратегії омолодження в реальних умовах та оцінити їх вплив на покращення метрик старіння. Зокрема, важливо перевірити та уточнити такі етапи методу, як прогнозування часу омолодження, конкретні реалізації механізмів "теплого" та "холодного" омолодження, визначення стану системи на підставі запропо-

нованих метрик та їх порогів, інтервалів між вимірюваннями метрик, критеріїв $T_{RBottom}$, T_{RTop} та $T_{RPostpone}$.

Подяки. Держбюджетна НДР "Розроблення інформаційної технології оцінювання та прогнозування надійності програмного забезпечення методами машинного навчання", номер держреєстрації 0121U109527.

References

1. Ahamad, S. (2016). Study of software aging issues and prevention solutions. *The International Journal of Computer Science and Information Security*, 14(8), 307–313.
2. Cotroneo, D., Fucci, F., Iannillo, A. K., Natella, R., & Pietrantuono, R. (2016). Software aging analysis of the Android mobile OS. *IEEE 27th International Symposium on Software Reliability Engineering*, 478–489. <https://doi.org/10.1109/ISSRE.2016.25>
3. Cotroneo, D., Iannillo, A. K., Natella, R., & Pietrantuono, R. (2020). A Comprehensive Study on Software Aging across Android Versions and Vendors. *Empirical Software Engineering*, 25, 3357–3395. <https://doi.org/10.1007/s10664-020-09838-3>
4. Cotroneo, D., Natella, R., Pietrantuono, R., & Russo, S. (2014). A Survey of Software Aging and Rejuvenation Studies. *ACM Journal on Emerging Technologies in Computing Systems*, 10(1). <https://doi.org/10.1145/2539117>
5. Cotroneo, D., Simone, L. D., Natella, R., Pietrantuono, R., & Russo, S. (2019). A Configurable Software Aging Detection and Rejuvenation Agent for Android. *11th Intl Workshop on Software Aging and Rejuvenation (WoSAR)*. <https://doi.org/10.1109/ISSREW.2019.00078>
6. Grottke, M., Jr, R. M., & Trivedi, K. S. (2008). The fundamentals of software aging. *Proceedings of IEEE 19th International Symposium on Software Reliability Engineering*, 1–6.
7. Guo, C., Wu, H., Hua, X., Lautner, D., & Ren, S. (2015). Use Two-Level Rejuvenation to Combat Software Aging and Maximize Average Resource Performance. *IEEE 12th International Conference on Embedded Software and Systems*, 1160–1165.
8. Hrytsiuk, Yu. I., & Nemova, E. A. (2018). Peculiarities of Formulation of Requirements to the Software. *Scientific Bulletin of UNFU*, 28(7), 135–148. <https://doi.org/10.15421/40280727>
9. Huang, Y., Kintala, C., Kolettis, N., & Fulton, N. D. (1995). Software rejuvenation: Analysis, module and applications. *Proceedings of 25th Symposium on Fault Tolerant Computing*, 381–390. <https://doi.org/10.1109/FTCS.1995.466961>
10. ISO 25010. (n.d.). Retrieved from: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?limit=3&limitstart=0/>
11. Norris, J. R. (1997). Markov Chains. *Cambridge University Press*.
12. Pamas, D. L. (1994). Software aging. *Proceedings of 16th International Conference on Software Engineering*, 279–287. <https://doi.org/10.1109/ICSE.1994.296790>
13. Valentim, N. A., Macedo, A., & Matias, R. (2016). A Systematic Mapping Review of the First 20 Years of Software Aging and Rejuvenation Research. *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. <https://doi.org/10.1109/ISSREW.2016.42>
14. Weng, C., Zhao, D., Lu, L., Xiang, J., Yang, C., & Li, D. (2017). A Rejuvenation Strategy in Android. *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 273–279. <https://doi.org/10.1109/ISSREW.2017.50>
15. Xiang, J., Weng, C., Zhao, D., Tiana, J., Xiong, S., Lia, L., Andrzejak, A. (2018). A New Software Rejuvenation Model for Android. *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. <https://doi.org/10.1109/ISSREW.2018.00021>
16. Yakovyna, V. S., & Uhrynovskiy, B. V. (2020). Model starinnia ta omolodzhennia programnoho zabezpechennia dlia platformy android. *XX International Scientific and Technical Conference "Measuring and Computing in Technological Processes"*, 139–140. [In Ukrainian].
17. Yakovyna, V. S., & Uhrynovskiy, B. V. (2020). Starinnia programnoho zabezpechennia mobilnykh dodatkov: analiz problematyky. *Scientific bulletin of Ukrainian National Forestry University*, 30(2), 107–112. <https://doi.org/10.36930/40300219>
18. Yakovyna, V. S., & Uhrynovskiy, B. V. (2021). Doslidzhennia systemnykh protsesiv ta korystuvatskykh dodatkov operatsiinoi systemy android v konteksti yavshcha starinnia programnoho zabezpechennia. *Herald of Khmelnytskyi national university*, 2(295), 64–70. <https://www.doi.org/10.31891/2307-5732-2021-295-2-64-70>
19. Yakovyna, V. S., & Uhrynovskiy, B. V. (2021). Extended software aging and rejuvenation model for Android operating system considering different aging levels and rejuvenation procedure types. *International scientific journal "Computer systems and information technologies"*, 3, 64–76.
20. Yakovyna, V. S., & Uhrynovskiy, B. V. (2021). Metryky interfeisu korystuvacha dlia vyiavlennia yavshcha starinnia programnoho zabezpechennia v mobilnii systemi android. *Journal of Lviv Polytechnic National University "Information Systems and Networks"*, 9, 32–43. <https://doi.org/10.23939/sisn2021.09.032>

V. S. Yakovyna, B. V. Uhrynovskiy

Lviv Polytechnic National University, Lviv, Ukraina

SOFTWARE REJUVENATION METHOD FOR ANDROID OPERATING SYSTEM CONSIDERING COMPLEX AGING MODEL BASED ON CONTINUOUS TIME MARKOV CHAINS

The software aging process consists in accumulation of aging-related errors that cause system performance degradation and aging-related failures rate increasing. Aging phenomenon has a negative impact on such software quality characteristics as reliability and performance. Mobile devices and systems, particularly those based on the Android platform, are vulnerable to software aging effects. Software rejuvenation procedure is a proactive approach of counteracting the aging phenomenon effects, which consists in regular cleaning of the system state from accumulated aging-related errors. The method of software rejuvenation for the Android operating system is proposed in this paper. This method involves predicting of the rejuvenation time, using a complex software aging and rejuvenation model based on the continuous time Markov chain. The proposed method takes into account different software aging levels, user activity of the mobile device and battery level to predict the rejuvenation time and select the rejuvenation strategy. Gradation of aging levels in the method allows to perform rejuvenation only at the stage when the aging process is observed in the system and leads to the aging-related failure. The method involves a choice between warm and cold rejuvenation mechanisms, and in the case of active usage of the mobile phone, the method allows the user to decide for themselves whether to perform or postpone rejuvenation. The rejuvenation method is proposed to be implemented in the form of system component which consists of two modules that run in the background. The aging metrics monitor module regularly collects the data needed to calculate the input data for the rejuvenation and prediction method. The rejuvenation module directly performs the steps of the method, which provide the calculation of the rejuvenation time and the decision to perform a particular rejuvenation mechanism. The rejuvenation method and its adjustment described in the paper require experimental research and testing on real data and in real conditions. In addition, it is important to continue working on detailing individual mechanisms and steps, in particular, calculating the target function for predicting rejuvenation time, mechanisms of warm and cold rejuvenation for the Android operating system, as well as calculating transition rates and defining current system state.

Keywords: software reliability; user experience; aging-related factors; aging-related metrics; mobile devices; software rejuvenation procedure.