ВИКОРИСТАННЯ ОДНОПЛАТНИХ КОМП'ЮТЕРІВ ДЛЯ МОДЕЛЮВАННЯ ПРОЦЕСІВ ШАХТНИХ ВЕНТИЛЯЦІЙНИХ МЕРЕЖ

Новіков В. Г., Ковальов С. О.

Виконано аналіз одноплатних комп'ютерів на базі Ordroid XU4. Зроблено портування програмного забезпечення (ПЗ) моделювання динамічних процесів шахтних вентиляційних мереж ШВМ на одноплатний комп'ютер. ПЗ має 4 режими моделювання: для одного елементу ШВМ, для повітряної секції з чотирьох елементів, для газово-повітряної секції з чотирьох елементів та для чотирьох елементної мережі з динамікою повітряних мас. Виконано відлагодження та модернізування ПЗ на одноплатному комп'ютері. Розглянуто способи збереження отриманих даних з використанням NoSQL баз даних (БД) на прикладі ElasticSearch. Приведено пакетний спосіб передачі інформації до БД з використанням JSON файлів.

Ключові слова: одноплатний комп'ютер, моделювання, шахтні вентиляційні мережі, бази даних, NoSQL, ElasticSearch, JSON.

Вступ. Шахтна вентиляційна система – система з'єднаних між собою підземних виробок шахти, що забезпечує спрямований рух повітря для провітрювання. Включає джерела тяги (вентилятори), вентиляційні регулятори і споруди, шляхи витоку повітря. Рух повітря у вентиляційній мережі підтримується за допомогою напору (або розрідження), що створюється одним або декількома вентиляторами [1].

В сучасних вугільних шахтах вентиляційна мережа містить до 300-500 гілок, в рудних - до 1000. Найпростішим елементом шахтної вентиляційної системи можна вважати гілку, яку можна уявити трубою (рис 1). Основні параметри гілки вентиляційної мережі (аеродинамічний опір – r , витрати повітря – q і втрати тиску – h , в кожній гілці) пов'язані між собою співвідношеннями виду $h = r \cdot q^2$ (гілки з турбулентним рухом повітря) або $h = r \cdot q$ (гілки з ламінарним рухом повітря). Вентиляційна мережа в цілому характеризується параметрами (в разі одного вентилятора головного провітрювання):

- сумарна витрата повітря – Q ;
- сумарна втрата тиску – H ;
- загальний опір – R ,

які пов'язані рівнянням $H = R \cdot Q^2$, званім характеристикою вентиляційної системи [2].

Рисунок 1 – Елемент вентиляційної системи

Елементи вентиляційної системи можна об'єднати у найпростішу секцію (рис 2а) або мережу (рис 2б).

Рисунок 2 – Секція (а) та мережа (б) вентиляційної системи.

Особливою відмінністю секції або мережі є наявність декількох елементів, кожен з яких має свій опір. В кожному з елементів ШВМ можуть знаходитись:

- перешкоди;
- елементи зворотної тяги;
- інший склад повітряних мас.

У якості перешкод з елементами зворотної тяги можуть виступати шахтні машини та агрегати, що використовуються у виробництві, активна система охолодження, яка спрямовує потік повітря на зустріч потоку ШВМ.

Процес зміни руху повітряних мас не може відбуватися миттєво, саме тому, це може призвести до негативних або, навіть, катастрофічних наслідків (недостатність кисню, великий рівень метану тощо). Тому було розроблено додаток, який дозволяє моделювати ці процеси та попереджати надзвичайні ситуації.

Додаток працює на UNIX подібній системі персонального комп'ютера. Та для

використання додатку в реальних умовах необхідно перенести додаток на більш мобільну платформу. Саме тому було обрано використання одноплатних комп'ютерів.

Метою роботи є портування та модернізування існуючого ПЗ на одноплатний комп'ютер.

Задачі роботи:

1. Аналіз ПЗ для моделювання динамічних процесів ШВМ;
2. Огляд та налагодження платформи одноплатного комп'ютера ODROID XU4;
3. Портування та налагодження ПЗ для моделювання динамічних процесів ШВМ на ОК ODROID XU4;
4. Розгортання NoSQL бази даних ElasticSearch на платформі ODROID XU4;
5. Реалізація та відлагодження функції запису даних до бази даних;
6. Дослідження продуктивності роботи із БД.

Особливості роботи додатку ПЗ моделювання динамічних процесів ШВМ. Додаток має чотири режими роботи:

- одиночний елемент з динамікою повітряних мас;
- чотирьох-елементна секція з динамікою повітряних мас;
- чотирьох-елементна секція з динамікою газових мас;
- чотирьох-елементна мережа з динамікою повітряних мас.

Результати роботи ПЗ, на початковому етапі, зберігаються у вигляді csv файлів. Приклад наповнення результуючого файлу наведено на рис. 3.

```
it;H;r_reg;Q
0;0,000000;0,000000;0,000000
1;12,440000;0,000000;0,134997
2;12,440000;0,000000;0,269912
3;12,440000;0,000000;0,404648
4;12,440000;0,000000;0,539113
5;12,440000;0,000000;0,673212
6;12,440000;0,000000;0,806853
7;12,440000;0,000000;0,939945
8;12,440000;0,000000;1,072398
9;12,440000;0,000000;1,204124
10;12,440000;0,000000;1,335037
11;12,440000;0,000000;1,465054
12;12,440000;0,000000;1,594091
13;12,440000;0,000000;1,722072
14;12,440000;0,000000;1,849919
```

Рисунок 3 – Приклад наповнення результуючого файлу ПЗ.

Цей файл дуже складний в сприйнятті інформації людиною. Саме тому було вирішено відобразити інформацію у вигляді графіків. Для цього було використано програмне забезпечення UNIX систем – GNU_plot.

GNU_plot – вільна програма для створення дво- і тривимірних графіків. Вона має власну систему команд, може працювати в режимі командної строки або виконувати скрипти, які зчитуються з файлів. GNU_plot виводить графіки як безпосередньо на екран, так і в файли різних графічних форматів, таких як PNG, EPS, SVG, JPEG та інших [3].

Для обробки інформації з csv-файлу програмою GNU_plot, необхідно створити скрипт, в якому позначити ім'я файлу, розділовий символ у рядку, тип даних та всі інші параметри, необхідні для побудови графіку.

Результати побудови графіків для чотирьох-елементної мережі наведено на рис. 4.

Рисунок 4 – Графіки GNU_plot, виконання четвертої функції ПЗ

Як можна помітити, GNU_plot дозволив візуалізувати отримані данні, та вони стали більш наочними. Також використання GNU_plot дозволяє експортувати графіки у графічні файли для подальшого аналізу або роздрукування.

Огляд та налагодження платформи одноплатного комп'ютера ODROID XU4. Кінцева платформа, на якій повинно працювати ПЗ – ODROID XU4. ODROID – це серія одноплатних комп'ютерів і планшетних ПК, створених компанією Hardkernel Co., Ltd., що розташована в Південній Кореї. Незважаючи на те, що ім'я ODROID є портмантом Open + Android, обладнання насправді не відкрите, тому що частина дизайну зберігається компанією. ODROID XU4 (рис. 5) – це нове покоління обчислювальних пристроїв з більш потужним, енергозберігаючим обладнанням та меншим форм-фактором ніж його попередники. Запропонувавши підтримку з

відкритим вихідним кодом, плата може запускати різні версії Linux, включаючи останню версію Ubuntu 16.04 та Android 4.4 KitKat і 7.1 Nougat. Використовуючи інтерфейси eMMC 5.0, USB 3.0 та Gigabit Ethernet, ODROID XU4 має чудову швидкість передачі даних, яка потрібна для підтримки високої потужності обробки на пристроях ARM. В табл. 2 представлені технічні характеристики одноплатного комп'ютера ODROID-XU4 [4].

Рисунок 5 – Одноплатний комп'ютер ODROID XU4

Таблиця 2 – Технічні характеристики ODROID XU4

Тип характеристики	Опис
Процесор	Samsung Exynos5 Octa ARM Cortex™-A15 Quad 2Ghz and Cortex™-A7 Quad 1.3GHz CPUs
Пам'ять	2Гб LPDDR3 O3Y на 933МГц (14.9Гб/с пропускна здатність пам'яті)
3D акселератор	Mali-T628 MP6(OpenGL ES 3.0/2.0/1.1 and OpenCL 1.1 Full profile)
Відео	підтримує 1080p через кабель HDMI (контейнер формату H.264 + AAC на основі MP4)
USB3.0 хост	SuperSpeed USB стандарту А x 2 порти, максимальне завантаження: загальний 2Amp для 2-х портів хоста USB 3.0
USB2.0 хост	Високошвидкісний стандартний роз'єм типу X 1 портів, максимальна навантаження: 500 мА/порт
Wi-Fi	USB IEEE 802.11b / g / n 1T1R WLAN з антеною (USB-модуль)

Портування та налагодження ПЗ моделювання динамічних процесів ШВМ на ОК ODROID XU4. ПЗ моделювання динамічних процесів ШВМ на початковому етапі було розроблено для виконання на UNIX подібних системах. На платформі ОК ODROID XU4 встановлено саме таку систему. Тому, складності портування програмного забезпечення були пов'язані з використанням іншої версії компілятора GPP. При портуванні виникли незначні помилки компілятора, в більшості пов'язані з різницею бібліотек або типу змінних. Різницю в архітектурі було вирішено завдяки інтегрованим засобам компілятора, а всі інші

незначні помилки вирішувались за мірою їх виникнення.

При налагодженні програмного забезпечення було поставлено декілька додаткових задач:

- виконати експорт до результуючого файлу часу у форматі HH:mm:ss, відштовхуючись від системного часу;
- виконати експорт до результуючого файлу часу, відштовхуючись від заданого часу;
- підрахувати час виконання кожного експерименту для кожної функції ПЗ.

Розгортання NoSQL бази даних ElasticSearch на платформі ODROID XU4. Оскільки збереження інформації у файлах не

є найкращим рішенням, було прийнято рішення використовувати для цього БД. Проаналізувавши існуючі рішення було обрано Elasticsearch [5], перевагами якого є:

- безкоштовність;
- швидкість роботи;
- легка масштабованість;
- висока відмовостійкість;
- велика кількість типів даних.

ElasticSearch - це платформа для розподіленого пошуку та аналізу даних в режимі реального часу. Вона користується великою популярністю завдяки простоті, потужності і масштабованості.

ElasticSearch підтримує операції REST, що дозволяє застосовувати методи HTTP (GET, POST, PUT, DELETE тощо).

ElasticSearch є вільним і відкритим програмним забезпеченням відомої компанії Elastic. Тому ElasticSearch відмінно підходить для використання в самих різних умовах [6].

ElasticSearch можна завантажити з сайту проекту в форматі zip, tar.gz, deb і rpm. В системі Ubuntu краще використовувати пакет deb, який встановить все необхідне для коректної роботи ElasticSearch.

Файли Elasticsearch знаходяться в каталозі /etc/elasticsearch.

Файл elasticsearch.yml містить серверні налаштування Elasticsearch. Тут зберігаються всі параметри програми. У цьому файлі потрібно відредагувати змінні node.name і cluster.name. Змінна node.name задає ім'я сервера, а cluster.name – ім'я кластера, до якого належить сервер.

Значення cluster.name використовується функцією автоматичного виявлення і підключення серверів Elasticsearch в кластер. Таким чином, значення за замовчуванням допускає в кластер небажані сервери, знайдені в цій мережі. Тому потрібно відредагувати даний файл в текстовому редакторі та змінити два параметри: cluster.name та node.name.

Після цього Elasticsearch можна вважати налаштованим та готовим до роботи [7].

Реалізація та відлагодження функції запису даних до бази даних.

Щоб записати до бази даних (відмічена червоним) в таблицю (відмічена жовтим) під індексом (відмічене зеленим) даних (відмічені синім) необхідно виконати команду, наведену на рис. 6 [8].

```
xvelv@ubuntu:~/Desktop/MegaSerial$ curl -XPUT 'localhost:9200/modeling/a111/17pratty' -d '{
  "time": "11:11:11",
  "H": "0,111111",
  "r_req": "0,222222",
  "Q": "0,333333"
}'
```

Рисунок 6 – Команда запису інформації до ElasticSearch

Для того, щоб отримати данні за заданим записом від ElasticSearch необхідно виконати команду, наведену на рис. 7. Відповідь також зображено на цьому рисунку.

```
xvelv@ubuntu:~/Desktop/MegaSerial$ curl -XGET 'localhost:9200/modeling/a111/17pratty'
{
  "_index" : "modeling",
  "_type" : "a111",
  "_id" : "1",
  "_version" : 27,
  "found" : true,
  "_source" : {
    "time": "11:11:11",
    "H": "0,111111",
    "r_req": "0,222222",
    "Q": "0,333333"
  }
}
```

Рисунок 7 – Команда отримання даних від ElasticSearch та її відповідь

Вигляд, у якому зберігаються данні в ElasticSearch:

- «_index» – назва бази даних;
- «_type» – назва таблиці;
- «_id» – ідентифікатор запису;
- «_version» – яка версія інформації зберігається за даним ідентифікатором;
- «found» – признак того, що інформація знайдена;
- «_source» – інформація, що зберігається.

Для програмного збереження інформації у ElasticSearch спочатку було прийнято рішення використовувати саме цей підхід.

Але після тестування отриманого додатку, було виявлено, що після налаштування послідовного (запис за записом) додавання інформації до ElasticSearch на віртуальній машині, час виконання програми зріс більше ніж у 30 разів. Результати зображені на рис. 8.

Через це було обрано додавання інформації до бази даних за допомогою JSON файлів.

JSON - текстовий формат обміну даними, заснований на JavaScript. Як і багато інших текстових форматів, JSON легко може бути прочитаним людиною. Незважаючи на походження від JavaScript, формат вважається незалежним від мови і може використовуватися практично з будь-якою

мовою програмування. Для багатьох мов існує готовий код для створення і обробки даних в форматі JSON [9].

```
xvelv@ubuntu: ~/Desktop/Serial (copy)$ su root
rm: cannot remove '.csv': No such file or directory
Please choose the benchmark to execute.

Available options:
-----
1. A single Element with air dynamics (the simplest)
2. A 4-elements Section with air and air dynamics (more complex)
3. A 4-elements Section with air and gas dynamics (more complex)
4. A 4-elements Network with air dynamics

Make you choice now and press enter: 1
Created Air Element: Roadway; l=500, F=7, r=0.24, ro=1.29, K=92.1429
RUN WITHOUT ARGS
Element Roadway: Applied initial ctrl set: H=0, r_reg=0
Element Roadway: Applied ctrl set: H=12.44, r_reg=0
Element Roadway: Applied ctrl set: H=12.44, r_reg=4.5
Element Roadway: Applied initial ctrl set: H=12.44, r_reg=4.5
Element Roadway: Applied ctrl set: H=12.44, r_reg=2.5
Time of the First experiment: 4ms
Time of the Second experiment: 1ms
Time of the Third experiment: 1ms
Time of the All experiments: 7ms
Simulation completed
xvelv@ubuntu:~/Desktop/Serial (copy)$
```

```
{ "index": "modeling", "type": "a111", "id": "517", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "518", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "519", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "520", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "521", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "522", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "523", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "524", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "525", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "526", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "527", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "528", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "529", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "530", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "531", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "532", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "533", "version": 2, "created": false }
{ "index": "modeling", "type": "a111", "id": "534", "version": 2, "created": false }
Time of the First experiment: 142ms
Time of the Second experiment: 31ms
Time of the Third experiment: 44ms
Time of the All experiments: 220ms
Simulation completed
xvelv@ubuntu:~/Desktop/Serial$
```

Рисунок 8 – Порівняння часу виконання ПЗ з послідовним занесенням до ElasticSearch та без занесення до БД

При виконанні ПЗ інформація, що була підрахована на теперішньому етапі, експортується до csv файлу, як це було на початковому етапі, та до JSON файлу в необхідному форматі. Виходячи з цього, кількість записів у JSON файлі така ж, як і в csv файлі та дорівнює кількості секунд у вказаному для моделювання часу. Перед завершенням роботи ПЗ виконується запит до ElasticSearch для передачі сформованого файлу.

Одним з стандартних структур файлів JSON є набір пар «ключ : значення». Саме ця структура використана при створенні файлу програмним забезпеченням (рис. 9) [10].

Дослідження продуктивності роботи із БД. Наступним етапом було зроблено порівняльну характеристику швидкодії ПЗ. Експеримент проводився з та без занесення до БД. Результати наведено в мілісекундах. Першим експериментом було порівняння роботи на ВМ та на платформі одноплатного комп'ютера моделювання першої функції ПЗ. Заданий час моделювання першої функції – 1 година, результати наведено на рис. 10.

```
Open  J1.json (~/.Desktop/MegaSerial) - gedit

{ "index": { "index": "modeling", "type": "a111", "id": "1" } }
{ "Time": "18:10:44", "H": "0,000000", "r_reg": "0,000000", "Q": "0,000000" }
{ "index": { "index": "modeling", "type": "a111", "id": "2" } }
{ "Time": "18:10:45", "H": "12,440000", "r_reg": "0,000000", "Q": "0,134997" }
{ "index": { "index": "modeling", "type": "a111", "id": "3" } }
{ "Time": "18:10:46", "H": "12,440000", "r_reg": "0,000000", "Q": "0,269912" }
{ "index": { "index": "modeling", "type": "a111", "id": "4" } }
{ "Time": "18:10:47", "H": "12,440000", "r_reg": "0,000000", "Q": "0,404648" }
{ "index": { "index": "modeling", "type": "a111", "id": "5" } }
{ "Time": "18:10:48", "H": "12,440000", "r_reg": "0,000000", "Q": "0,539113" }
{ "index": { "index": "modeling", "type": "a111", "id": "6" } }
{ "Time": "18:10:49", "H": "12,440000", "r_reg": "0,000000", "Q": "0,673212" }
{ "index": { "index": "modeling", "type": "a111", "id": "7" } }
{ "Time": "18:10:50", "H": "12,440000", "r_reg": "0,000000", "Q": "0,806853" }
{ "index": { "index": "modeling", "type": "a111", "id": "8" } }
{ "Time": "18:10:51", "H": "12,440000", "r_reg": "0,000000", "Q": "0,939945" }
{ "index": { "index": "modeling", "type": "a111", "id": "9" } }
{ "Time": "18:10:52", "H": "12,440000", "r_reg": "0,000000", "Q": "1,072398" }
{ "index": { "index": "modeling", "type": "a111", "id": "10" } }
{ "Time": "18:10:53", "H": "12,440000", "r_reg": "0,000000", "Q": "1,204124" }
{ "index": { "index": "modeling", "type": "a111", "id": "11" } }
```

Рисунок 9 – Наповнення файлу JSON

Рисунок 10 – Час виконання ПЗ для моделювання одного елемента

Також порівняння часу моделювання було виконано для секції з повітрям на п'ятнадцять хвилин, для секції з повітрям та газом на шість з половиною годин та для мережі на три години вперед. Результати представлені на рис. 11 – 13 відповідно.

Рисунок 11 – Час виконання ПЗ для моделювання секції з повітрям

Рисунок 12 – Час виконання ПЗ для моделювання секції з повітрям та газом

Рисунок 13 – Час виконання ПЗ для моделювання мережі

В загальному випадку різниця у часі з занесенням до БД та без становить 1,5 – 2 рази, а виконання ПЗ на різних платформах від 2 до 7 разів. При цьому найбільша різниця в режимі без занесення до бази даних для секції з повітрям.

Висновок. Результати портування ПЗ моделювання процесів ШВМ наступні:

- програмне забезпечення працює стабільно;

- час моделювання може вважатися прийнятним (наприклад, моделювання секції з повітрям на 15 хвилин вперед виконується за 15 мілісекунд з занесенням до БД).

Збережена у БД інформація може бути в подальшому використана для більш детального аналізу та більш тонкого налаштування ПЗ.

Також слід зазначити що дана тема має актуальні напрямки розвитку, такі як:

- використання спільного серверу збереження даних з багатьох пристроїв;
- прогнозування в режимі реального часу в залежності від даних, отриманих одноплатним комп'ютером з датчиків;
- відображення графіків в реальному часі даних, що зберігаються на сервері, за допомогою інструменту Kibana;
- розпаралелювання ПЗ для підвищення швидкості роботи.

Література

- [1] Шахтна вентиляційна система: пат. 21781 Україна: опубл. 15.06.2001, 2с..
- [2] АЕРОЛОГІЯ ШАХТНИХ ВЕНТИЛЯЦІЙНИХ МЕРЕЖ / В.О. Трофимов та інш. Донецьк: Норд-Прес, 2009. 87 с..
- [3] Gnuplot - Вікіпедія [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Gnuplot>.
- [4] Odroid-XU4 - ODROID Wiki [електронний ресурс]. – Режим доступу: <https://wiki.odroid.com/odroid-xu4/odroid-xu4>.
- [5] Elasticsearch: The Official Distributed Search & Analytics [електронний ресурс]. – Режим доступу: <https://www.elastic.co/products/elasticsearch>.
- [6] Elasticsearch — Вікіпедія [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Elasticsearch>.
- [7] Основы Elasticsearch / Хабр - Habr [електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/280488/>.
- [8] 10 Elasticsearch Concepts You Need to Learn | Logz.io [електронний ресурс]. – Режим доступу: <https://logz.io/blog/10-elasticsearch-concepts/>.
- [9] JSON — Вікіпедія [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/JSON>.
- [10] Введение в JSON - Stas Bagretsov - Medium | Logz.io [електронний ресурс]. – Режим доступу: <https://medium.com/@stason...>

References

- [1] Шахтна вентиляційна система: пат. 21781 Україна: опубл. 15.06.2001, 2с..
- [2] АЕРОЛОГІЯ ШАХТНИХ ВЕНТИЛЯЦІЙНИХ МЕРЕЖ / В.О. Трофимов та інш. Донецьк: Норд-Прес, 2009. 87 с..
- [3] Gnuplot - Вікіпедія [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Gnuplot>.
- [4] Odroid-XU4 - ODROID Wiki [електронний ресурс]. – Режим доступу: <https://wiki.odroid.com/odroid-xu4/odroid-xu4>.
- [5] Elasticsearch: The Official Distributed Search & Analytics [електронний ресурс]. – Режим

- доступу: <https://www.elastic.co/products/elasticsearch>.
- [6] Elasticsearch — Википедія [електронний ресурс]. — Режим доступу: <https://ru.wikipedia.org/wiki/Elasticsearch>.
- [7] Основы Elasticsearch / Хабр - Habr [електронний ресурс]. — Режим доступу: <https://habr.com/ru/post/280488/>.
- [8] 10 Elasticsearch Concepts You Need to Learn | Logz.io [електронний ресурс]. — Режим доступу: <https://logz.io/blog/10-elasticsearch-concepts/>.
- [9] JSON — Википедія [електронний ресурс]. — Режим доступу: <https://ru.wikipedia.org/wiki/JSON>.
- [10] Введение в JSON - Stas Bagretsov - Medium | Logz.io [електронний ресурс]. — Режим доступу: <https://medium.com/@stason...>

Надійшла до редакції 25.04.2019 р.

Новіков Віталій Геннадійович — аспірант, Донецький національний технічний університет (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: vitalii.novikov@donntu.edu.ua.

Ковальов Сергій Олександрович — к. т. н., доцент, Державний вищий навчальний заклад «Донецький національний технічний університет» (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: sergiy.kovalov@donntu.edu.ua

USE OF SINGLE-BOARD COMPUTERS FOR SIMULATION OF MINE VENTILATION NETWORKS

Mine ventilation system is a system of interconnected underground workings of the mine that provides directional air movement for ventilation. The ventilation network of modern coal mines contains up to 300-500 branches, in the ore mines - up to 1000. The process of changing the air masses movement in such systems is quite complicated and can lead to negative or even catastrophic consequences (lack of oxygen, high levels of methane, etc.). Therefore, an software application was developed to simulate ventilation processes and prevent emergencies. The application runs on a UNIX-like personal computer system. But in order to use the application in real conditions, you need to move the application to a more mobile platform. The use of single-board computers was chosen and the task of appropriate porting and upgrading of existing software. An analysis of single board computers based on Ordroid XU4 has been performed. Porting software for simulating dynamic processes of mine ventilation networks (MVN) on a single board computer is made. The software has 4 simulation modes: for one element of the MVN, for the air section of four elements, for the gas-air section of four elements, and for the four element network with the calculation of air masses dynamics. Debugging and upgrading of the software on a single board computer has been completed. Porting errors have compiler errors, most of which are related to the difference in libraries or type of variables. Architectural differences were solved with the help of integrated compiler tools. The following tasks were solved under debugging: exporting to the resulting time file in HH: mm: ss format, starting from both the system time and the specified time; calculating the execution time of each experiment for each software function. The methods of the obtained data storing using NoSQL databases (DB) on the example of ElasticSearch are considered. Studies of different versions of software implementation showed that in the general case, the speed of storing the received data with the entry in the database is more than 1.5 - 2 times compared with the method without recording. A batch method of transferring information to a database using JSON files is provided.

Keywords: single board computer, simulation, mine ventilation networks, databases, NoSQL, ElasticSearch, JSON.

Novikov Vitalii Hennadiiovych — PhD-student, Donetsk National Technical University, (2, Shybankova square, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: vitalii.novikov@donntu.edu.ua.

Kovalov Serhii Oleksandrovych – PhD, Associate Professor, State Higher Education Establishment
“Donetsk National Technical University”, (Shybankova square, 2, Pokrovsk, Donetsk region, 85300,
Ukraine).
E-mail: sergiy.kovalov@donntu.edu.ua