Для перевірки можна використовувати програму HwInfo, яка показує частоти для кожного ядра. Паралельно можна запустити будь-якої бенчмарк або стрес-тест (наприклад cpu-z), щоб навантажити процесор. Якщо все пройшло вдало - частота кожного ядра буде дорівнює максимальному значенню турбо-буста процесора.

Порівнюємо анлок турбо-буста та стокового біоса, за допомогою програми Corona BenchMark бачимо що в рендері процесор з модифікованим біосом показує себе набагато краще , тим самим випереджаючи на 40 секунд стоковий біос.

В деяких іграх описаний метод показує високий приріст продуктивності , а в деяких ні. Це обумовлено оптимізацію ігор під наш процесор, а саме тактовою частотою процесора, деякі оптимізовані під високу частоту процесора , а інші такої оптимізації не мають, але все таки процесор з розблокованим TurboBoost, показує кращий результат.

Розглянутий метод дозволяє підвищити продуктивність обчислювальної системи на 20-50%,за допомогою видалення мікрокоду , що в свою чергу дозволить підвищує ефективність використання зазначених систем для вирішення задач кіберзахисту.

Список використаних джерел

1. https://uk.wikipedia.org/wiki/Haswell
2. https://ru.wikipedia.org/wiki/Extensible_Firmware_Interface
3. https://forums.overclockers.ru/viewtopic.php?f=1&t=479847&start=7560
4. http://xeonlive.ru/instruktsii/afuwin-proshivka-bios-iz-pod-windows

---

УДК 004.056.55

KUBERNETES - ОРКЕСТРАЦІЇ КОНТЕЙНЕРІВ, ЩО ЗАБЕЗПЕЧУЄ ВІДМОВОСТІЙКІСТЬ ТА ДОСТУПНІСТЬ ДОДАТКУ

Ткач Ю. М., д.п.н., проф.,
завідувач кафедри кібербезпеки та математичного моделювання,
Клименок В. О., здобувач вищої освіти гр. МКБп-201
Національний університет "Чернігівська політехніка"

Kubernetes - це потужна система з відкритим кодом, спочатку розроблена Google, для управління контейнерними програмами в кластерному середовищі. Вона спрямована на забезпечення кращих способів управління пов'язаними, розподіленими компонентами та послугами у різноманітній інфраструктурі.

Kubernetes, на своєму базовому рівні, є системою для запуску та координації контейнерних програм через кластер машин. Це платформа, призначена для повного управління життєвим циклом контейнерних програм та служб, використовуючи методи, що забезпечують передбачуваність, масштабованість та високу доступність.

Користувач Kubernetes може визначити, як повинні працювати програми та способи їх взаємодії з іншими програмами чи зовнішнім світом. Є можливість масштабувати сервіси вгору або вниз, виконувати оновлення, а також переключати трафік між різними версіями програм, щоб перевірити функції або відмовити розгортання. Kubernetes надає інтерфейси та складні примітивні елементи платформи, які дозволяють визначати та керувати своїми програмами з високим ступенем гнучкості, потужності та надійності.

Контейнери - це хороший спосіб об'єднати та запустити ваші програми. У production середовищі потрібно керувати контейнерами, в яких запущені програми, і переконатися, що немає простоїв. Наприклад, якщо контейнер виключається, потрібно запускати інший контейнер.

Kubernetes надає фреймворк для стійкого запуску розподілених систем. Він дбає про масштабування та відновлення програми після відмови, надає схеми розгортання тощо. Наприклад, Kubernetes може легко керувати “canary” розгортанням для системи.

Kubernetes надає наступні переваги:

1. **Автоматичний пошук сервісів** та **балансування навантаження**. Kubernetes може виставити назовні контейнер, використовуючи ім'я DNS або використовуючи власну IP-адресу. Якщо трафік до контейнера високий, Kubernetes може розподіляти мережевий трафік, щоб додаток був стабільним.
2. **Організація зберігання**. Kubernetes дозволяє автоматично монтувати довільну систему зберігання даних, наприклад, локальні сховища, загальнодоступні хмарні постачальники тощо.
3. **Автоматизовані розгортання** та їх відміна. Є можливість описати бажаний стан для розгорнутих контейнерів за допомогою Kubernetes, і він може змінити фактичний стан до бажаного стану з контрольованою швидкістю. Наприклад, можна автоматизувати Kubernetes для створення нових контейнерів для розгортання, видалення існуючих контейнерів та прийняття всіх їх ресурсів до нового контейнера.
4. **Ефективний запуск контейнерів.** Kubernetes має кластер вузлів, який він може використовувати для запуску контейнерних завдань. Користувач повідомляє Kubernetes, скільки процесора та пам'яті (ОЗУ) потрібно кожному контейнеру. Kubernetes може помістити контейнери на вузли, щоб ефективно використовувати ресурси.
5. **Самовідновлення та самолікування**. Kubernetes самостійно перезапускає контейнери, які виходять з ладу, замінює контейнери, вбиває контейнери, які не відповідають на визначену користувачем перевірку стану здоров’я, та не надає доступ клієнтам, поки вони не будуть готові до обслуговування.
6. **Управління секретами та конфігурацією.** Kubernetes дозволяє зберігати та керувати конфіденційною інформацією, такою як паролі, токени OAuth та ключі SSH. Можна розгортати та оновлювати секрети та конфігурацію програми, не відновлюючи образи контейнера.

Головний вузол (master) - панель керування для всього кластера Kubernetes. Компоненти головного вузла можна запускати на будь-якому вузлі кластера. Ключовими компонентами є:

1. Сервер **API**. Точка входу для всіх команд REST, єдиний компонент головного вузла, який є доступним для користувача.
2. Сховище даних (**etcd**). Потужне, послідовне та доступне сховище ключових значень, яке використовується кластером Kubernetes.
3. Планувальник (**scheduler**). Спостерігає за новоствореними подами та призначає їх вузлам. Розгортання подів і служб на вузлах відбувається через планувальник.
4. **Менеджер контролерів**: запускає всі контролери, які обробляють рутинні завдання кластера.
5. Робочі вузли (**worker nodes**). Робочі вузли містять усі необхідні служби для управління мережею між контейнерами, зв’язку з головним вузлом та призначення ресурсів запланованим контейнерам.
6. **Docker.** працює на кожному робочому вузлі та завантажує зображення та стартові контейнери.
7. **Kubelet**. Відстежує стан поду та гарантує, що контейнери працюють. Він також взаємодіє зі сховищем даних, отримуючи інформацію про сервіси та записуючи деталі про нещодавно створені.
8. **Kube-proxy**. Мережевий проксі-сервер і балансувальник навантаження для сервісу на одному робочому вузлі. Він відповідає за маршрутизацію трафіку.
9. **Kubectl**: інструмент CLI для спілкування користувачів із сервером API Kubernetes.

Поди (**pods**) - це найменші, найосновніші об’єкти для розгортання в Kubernetes. Pod представляє один екземпляр запущеного процесу у кластері. Поди містять один або кілька контейнерів, наприклад, контейнери Docker. Коли Pod запускає кілька контейнерів,

контейнери управляються як єдине ціле і діляться ресурсами одне з одним. Як правило, запуск декількох контейнерів в одному модулі є просунутим варіантом використання.

Сервіси (**services**) - абстрактний спосіб виставити програму, що працює на наборі Pods, як послугу мережі. Kubernetes надає подам їх власні IP-адреси та єдине DNS-ім'я для набору Pods, і може балансувати навантаження між ними.

Kubernetes - це проект, який дозволяє користувачам запускати масштабовані, високодоступні контейнерні робочі навантаження на абстрагованій платформі. Незважаючи на те, що архітектура та набір внутрішніх компонентів Kubernetes спочатку можуть здатися важкими та об’ємними, їх потужність, гнучкість та надійний набір функцій не мають собі рівних у світі з відкритим кодом. Розуміючи, як поєднуються основні блоки, можна почати розробляти системи, які повністю використовують можливості платформи для управління та масштабування робочих навантажень.

Список використаних джерел

1. Kubernetes Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://kubernetes.io/docs/home/

2. Ellingwood J. An Introduction to Kubernetes [Електронний ресурс] / Justin Ellingwood. – 2018. – Режим доступу до ресурсу: https://www.digitalocean.com/community/tutorials/an-introduction-to-kubernetes

3. Gerrard A. What Is Kubernetes? An Introduction to the Wildly Popular Container Orchestration Platform [Електронний ресурс] / Ali Gerrard. – 2019. – Режим доступу до ресурсу: https://blog.newrelic.com/engineering/what-is-kubernetes/