dr inż. Marta Chodyka[1], Łukasz Wielgosz[2],
mgr inż. Sebastian Sawczuk[3]
Akademia Bialska im. Jana Pawła II, Biała Podlaska

https://doi.org/10.29316/9788368103281_11

BEZPIECZEŃSTWO I ODPORNOŚĆ MIKROKLASTRÓW RASPBERRY PI W ZDECENTRALIZOWANYCH SIECIACH BLOCKCHAIN – STUDIUM PRZYPADKU MONERO

SECURITY AND RESILIENCE OF RASPBERRY PI MICROCLUSTERS IN DECENTRALIZED BLOCKCHAIN NETWORKS – A CASE STUDY OF MONERO

Streszczenie

Rozdział monografii analizuje możliwość wykorzystania mikroklastra Raspberry Pi do obsługi zdecentralizowanej sieci blockchain na przykładzie kryptowaluty Monero, ze szczególnym uwzględnieniem aspektów bezpieczeństwa i odporności systemu. Zweryfikowano, czy klaster zbudowany z niskokosztowych mikrokomputerów jest w stanie zapewnić bezpieczne i niezawodne funkcjonowanie w sieci blockchain oraz jak radzi sobie z awariami węzłów. Przeprowadzono eksperyment z klastrem złożonym z pięciu Raspberry Pi 5 skonfigurowanych w środowisku Kubernetes (k3s) – jeden węzeł nadrzędny i cztery podrzędne. Na każdym węźle uruchomiono kontenery z oprogramowaniem

[1] https://orcid.org/0000-0002-8819-2451, e-mail: m.chodyka@dyd.akademiabialska.pl
[2] student, https://orcid.org/0009-0006-6732-3447, e-mail: wielgosz13245@stud.akademiabialska.pl
[3] e-mail: s.sawczuk@dyd.akademiabialska.pl

górniczym Monero (XMRig) w celu symulacji procesu wydobycia. Monitorowano parametry wydajności (hashrate, obciążenie CPU, pobór mocy, temperatury) podczas stopniowego dodawania kolejnych węzłów oraz symulowano awarię jednego z nich. Klaster osiągnął łączną moc obliczeniową ~1,6 kH/s przy pełnej konfiguracji (5 węzłów), a wzrost wydajności był bliski liniowemu wraz z dołączaniem kolejnych jednostek. Zużycie energii rosło proporcjonalnie do obciążenia – od ~2 W na węzeł w spoczynku do ~6-8 W pod pełnym obciążeniem. System okazał się stabilny termicznie (temperatury <60°C) i wysoce efektywny energetycznie (~50 H/s na wat mocy). Symulacja awarii ujawniła spadek wydajności klastra proporcjonalny do utraconego węzła oraz brak automatycznego przełączenia obciążenia na pozostałe jednostki, co wskazuje na ograniczoną odporność na uszkodzenia w obecnej konfiguracji. Mimo to klaster zachował ciągłość pracy pozostałych węzłów, potwierdzając potencjał zastosowań w środowiskach wymagających taniej, rozproszonej infrastruktury obliczeniowej.

Słowa kluczowe: Raspberry Pi, mikroklaster, Monero, blockchain, bezpieczeństwo, odporność, Kubernetes, hashrate, RandomX, sieci rozproszone

Abstract

This book chapter examines the feasibility of employing a Raspberry Pi micro-cluster to operate a decentralized blockchain network using the Monero cryptocurrency, with particular emphasis on system security and resilience. The study verifies whether a cluster built from low-cost single-board computers can provide secure and reliable blockchain operations and how it copes with node failures. An experiment was conducted with a cluster of five Raspberry Pi 5 boards configured under Kubernetes (k3s) – one master node and four worker nodes. Each board ran containers with the Monero mining software XMRig to simulate the mining process. Performance metrics (hash-rate, CPU load, power consumption, temperature) were monitored while successively adding nodes, and a failure of one node was simulated. Results show that the fully configured cluster (five nodes) achieved a cumulative computational power of approximately 1.6 kH/s, with nearly linear performance growth as additional units were added. Energy consumption increased proportionally with load-rising from ~2 W per node at idle to ~6-8 W under full load. The system remained thermally stable (temperatures < 60 °C) and highly energy-efficient (~50 H/s per watt). The failure simulation revealed a drop in cluster performance proportional to the lost node and no automatic workload fail-over, indicating limited fault tolerance in the current configuration. Nonetheless, the remaining nodes maintained continuous operation, confirming the potential of such clusters for environments that require low-cost, distributed computing infrastructure.

Keywords: Raspberry Pi, micro-cluster, Monero, blockchain, security, resilience, Kubernetes, hash-rate, RandomX, distributed networks

Wstęp

Rozproszona technologia blockchain zyskała reputację jako fundament bezpiecznych, odpornych na awarie systemów transakcyjnych pozbawionych centralnego punktu kontroli (Nakamoto, 2008). W tradycyjnych sieciach finansowych czy krytycznej infrastrukturze teleinformatycznej awaria pojedynczego, scentralizowanego serwera może skutkować poważnym zakłóceniem usług. Blockchain dzięki swojej decentralizacji i mechanizmom konsensusu oferuje model, w którym wiele niezależnych węzłów przechowuje i weryfikuje kopie tego samego rejestru, eliminując pojedynczy punkt awarii i utrudniając manipulacje danymi (Nofer i in, 2017). Z tego powodu rozważane jest zastosowanie technologii blockchain do zwiększenia odporności cybernetycznej infrastruktury krytycznej – jej niezmienność i rozproszenie mogą wzmocnić bezpieczeństwo systemów kluczowych dla społeczeństwa (Govea i in, 2014). Równocześnie jednak klasyczne wdrożenia blockchain (np. Bitcoin) wiążą się z rosnącymi wymaganiami sprzętowymi i energetycznymi, które nie zawsze są akceptowalne w kontekście zrównoważonego rozwoju czy zastosowań w urządzeniach o ograniczonych zasobach. Monero (ticker: XMR) jest przykładem kryptowaluty, która od początku projektowana była z myślą o maksymalnej descentralizacji wydobycia oraz ochronie prywatności użytkowników. W przeciwieństwie do Bitcoina czy wielu innych monet, gdzie dominują wyspecjalizowane układy ASIC i duże farmy górnicze, Monero wykorzystuje algorytm Proof-of-Work o nazwie RandomX, zoptymalizowany pod kątem wydajności na procesorach ogólnego zastosowania (CPU). RandomX cechuje się wysoką odpornością na implementacje ASIC poprzez zastosowanie losowego wykonywania kodu i intensywnego wykorzystania pamięci, co sprawia, że przewaga sprzętu specjalizowanego nad standardowymi CPU zostaje zminimalizowana (Trail of Bits, 2019). Dzięki tej filozofii „egalitarnego" wydobycia, do sieci Monero mogą dołączać nawet użytkownicy dysponujący niedrogim sprzętem, takim jak komputery osobiste czy mikrokontrolery, bez obawy o bycie zdominowanym przez potężne kopalnie. Dodatkowym wyróżnikiem Monero z punktu widzenia bezpieczeństwa jest nacisk na anonimowość transakcji – wykorzystanie unikalnych rozwiązań kryptograficznych (podpisów pierścieniowych,

jednorazowych adresów stealth czy ukrytych kwot dzięki RingCT) sprawia, że Monero bywa określane „królem prywatności" wśród kryptowalut (Hoskinson, 2021). Te cechy czynią Monero interesującym obiektem badań nad możliwościami uruchamiania sieci blockchain na nietypowych platformach sprzętowych. Wraz z pojawieniem się wydajniejszych generacji mikrokomputerów jednopłytkowych, takich jak Raspberry Pi 4 i 5, otworzyła się możliwość tworzenia niewielkich, tanich klastrów obliczeniowych zdolnych do wykonywania względnie złożonych zadań. Raspberry Pi w kolejnych wersjach zyskały coraz mocniejsze procesory (np. czterordzeniowy Cortex-A76 2,4 GHz w RPi 5) oraz większą ilość pamięci RAM, co czyni je potencjalnie zdolnymi do obsługi wymagających algorytmów, takich jak RandomX (Raspberry Pi Foundation, 2023). W literaturze odnotowano wiele eksperymentów z klastrami Raspberry Pi, od małych grup kilku urządzeń po instalacje liczące dziesiątki, a nawet setki węzłów. Badania wskazują, że choć moc obliczeniowa takiego mikroklastra jest ograniczona, może on stanowić skalowalne i odporne komponenty obliczeniowe na brzegu sieci (edge), zachowując przy tym niski koszt wdrożenia (Komnison i in, 2019). Mikroklastry znalazły zastosowanie m.in. w edukacji (do nauki programowania równoległego i architektur rozproszonych), w systemach IoT, a nawet w badaniach z obszaru bezpieczeństwa. W kontekście kryptowalut, pomysł wykorzystania klastrów Raspberry Pi idzie w parze z rosnącym zainteresowaniem infrastrukturą chmurową rozproszonej na brzegu sieci. Dzięki konteneryzacji i narzędziom orkiestracji, takim jak Kubernetes, można zarządzać wieloma węzłami jako jednolitą platformą usługową. DeCarlo (DeCarlo, 2018) zademonstrował, że użycie lekkiej dystrybucji Kubernetes (k3s) na klastrze Raspberry Pi pozwala uruchomić rozproszoną sieć kryptowaluty z mechanizmami automatycznego równoważenia obciążenia, zwiększając niezawodność i czas bezawaryjnej pracy takiego systemu.

Łącząc powyższe trendy, celem niniejszego badania było sprawdzenie czy sieć blockchain Monero – znana z odporności na specjalizowany sprzęt i przyjazna dla CPU – może funkcjonować na mikroklastrze Raspberry Pi oraz jakie wyzwania bezpieczeństwa i niezawodności się z tym wiążą. Szczególnie istotne było zbadanie odporności systemu na awarie: w klasycznym ujęciu sieć blockchain jest odporna na wyłączenie pojedynczego węzła (reszta

sieci nadal działa poprawnie), jednak w konfiguracji klastrowej, gdzie wiele urządzeń współpracuje w ramach jednego „węzła logicznego", pojawia się pytanie o wewnętrzną tolerancję uszkodzeń. Rozdział koncentruje się zatem na dwóch aspektach: wydajność i skalowalność – jak dodawanie kolejnych mikrokomputerów wpływa na moc obliczeniową i wykorzystanie zasobów, oraz bezpieczeństwo i niezawodność – jak klaster reaguje na awarie części składowych oraz jakie wnioski płyną z tego dla projektowania odpornych systemów rozproszonych.

Architektura rozwiązania i metodologia badawcza

Platforma sprzętowa: Do eksperymentu zbudowano klaster złożony z pięciu jednopłytkowych mikrokomputerów Raspberry Pi 5. Wybór najnowszego modelu podyktowany był jego ulepszonymi parametrami – m.in. szybszym czterordzeniowym procesorem 64-bit ARM, zwiększoną przepustowością I/O oraz obsługą interfejsu GbE, co redukuje wąskie gardła poprzednich generacji (Raspberry Pi Foundation, 2023). Każdy z Raspberry Pi został wyposażony w 8 GB RAM oraz kartę microSD klasy U3 (64 GB) na system operacyjny i dane blockchain. Ze względu na wysokie obciążenie obliczeniowe związane z wydobywaniem Monero, wszystkie jednostki chłodzone były aktywnie za pomocą miniaturowych wentylatorów, aby utrzymać stabilne temperatury pracy (~60°C). Poszczególne Raspberry Pi połączono w sieć lokalną za pośrednictwem przełącznika Gigabit Ethernet, co zapewniło niskie opóźnienia i wystarczającą przepustowość dla komunikacji w klastrze. Oprogramowanie i konfiguracja klastra: Na wszystkich węzłach zainstalowano minimalistyczny system Raspberry Pi OS Lite (64-bit) oraz lekką platformę orkiestracji kontenerów Kubernetes k3s. Jeden z mikrokomputerów pełnił rolę węzła nadrzędnego (master) klastra k3s – uruchomiono na nim serwer Kubernetes zarządzający harmonogramem zadań i stanem klastra. Pozostałe cztery Raspberry Pi działały jako węzły robocze (workery) dołączone do klastra. Takie rozproszone środowisko pozwoliło traktować sumę zasobów wszystkich urządzeń jako wspólną pulę obliczeniową. W kontenerach Docker na węzłach wdrożono instancje oprogramowania XMRig – popularnego open-source narzędzia do wydobywania Monero (wersja skompilowana dla

architektury ARM64). Kontenery zostały przygotowane tak, by w momencie uruchomienia automatycznie dołączać do publicznego poolu miningowego MoneroOcean, co umożliwiało wspólne wydobywanie i śledzenie udziału klastra w rozwiązywaniu bloków. Uruchomienie górników w kontenerach pod kontrolą k3s zapewniało, że zadania były zarządzane centralnie – węzeł master przydzielał kontenery (pody) do dostępnych Raspberry Pi zgodnie z zadeklarowaną konfiguracją. Początkowo skonfigurowano wdrożenie tylko jednego poda XMRig (na jednym z węzłów), następnie stopniowo skalowano deployment zwiększając liczbę replik, aby sprawdzić wpływ liczby równoległych instancji (pracujących na odrębnych urządzeniach) na łączny hashrate i obciążenie systemu. Pomiary i monitorowanie: W trakcie eksperymentu kluczowe metryki systemowe były monitorowane z użyciem wielu narzędzi. Do bieżącego śledzenia wykorzystania procesora, pamięci i wątków na poszczególnych Pi użyto htop, zaś do analizy zużycia energii i stanu zasilania – powertop. Ponadto wdrożono kontenery z Prometheus i Grafana w celu zbierania i wizualizacji danych telemetrycznych z węzłów (CPU load, temperatura, zużycie RAM, wykorzystanie sieci). Z poziomu MoneroOcean rejestrowano akceptowane przez pulę udziały (shares) oraz raportowany przez pulę łączny hashrate klastra (uśredniany w czasie). Dodatkowo logi tekstowe XMRig z każdego poda zbierano i analizowano pod kątem stabilności pracy (zanotowane bieżące wartości H/s każdego wątku, błędy połączeń itp.). Wszystkie pomiary prowadzono w sieci testowej przez kilkanaście godzin ciągłej pracy, co umożliwiło osiągnięcie stanu ustabilizowanego wydobycia i temperatur. Plan testów: Scenariusz eksperymentu podzielono na etapy odpowiadające zwiększaniu skali i badaniu odporności:

Test wydajności skalowania: Uruchomiono klaster z początkowo 1 aktywnym węzłem górniczym (1 pod XMRig na jednym Raspberry Pi). Następnie co ok. 30 minut dodawano kolejną instancję górnika (na kolejnym Pi), aż do wykorzystania wszystkich 5 urządzeń. Na każdym poziomie (1 do 5 węzłów) notowano stabilny hashrate łącznie i per węzeł, średnie obciążenie CPU, wykorzystanie pamięci oraz łączny pobór mocy przez cały klaster. Dzięki temu określono skalowalność rozwiązania – czy przyrost mocy obliczeniowej jest liniowy, malejący czy występują ewentualne efekty synergii lub ograniczenia zasobów.

Test odporności na awarię: Po osiągnięciu pełnej konfiguracji (5 działających podów) przeprowadzono symulację awarii jednego z węzłów. W praktyce zrealizowano to poprzez odizolowanie węzła master od sieci (odłączenie przewodu Ethernet), co odpowiada utracie węzła zarządzającego klastrem. Taki scenariusz jest skrajnie niekorzystny, ponieważ zakłada awarię centralnego komponentu odpowiedzialnego za koordynację – pozwala jednak ocenić, jak system rozproszony zachowa się przy utracie jednostki pełniącej kluczową rolę. Monitorowano natychmiastowy wpływ na hashrate (zarówno globalny klastrowy, jak i poszczególnych pozostałych węzłów), stan podów oraz próby ich ewentualnej relokacji przez Kubernetes. Dodatkowo zbadano logi górników i poolu pod kątem błędów komunikacji z utraconym węzłem.

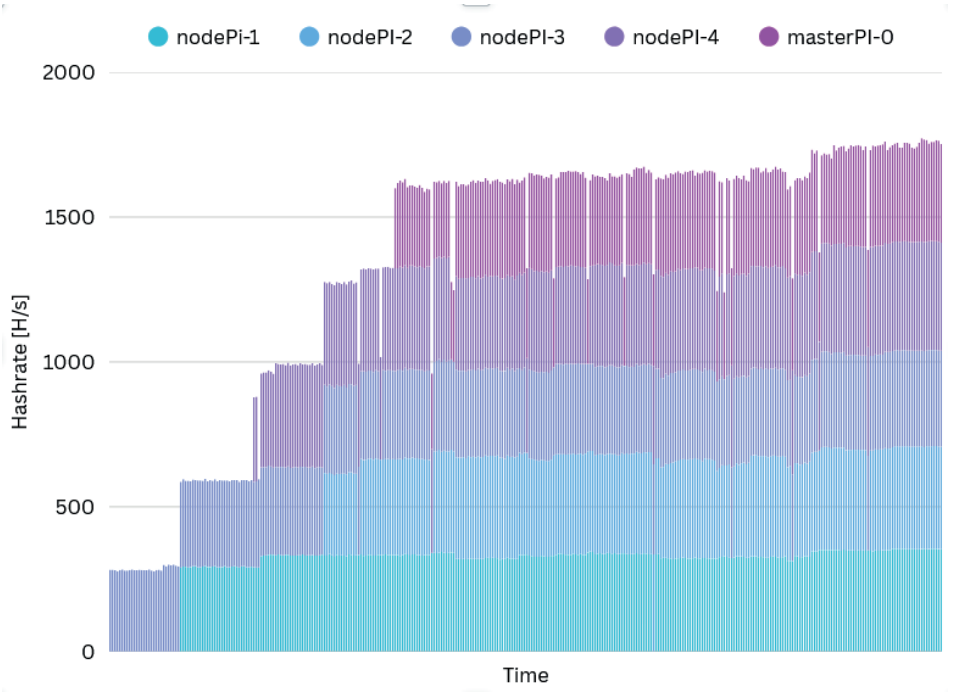
Test ciągłości pracy: W ramach oceny niezawodności, klaster (już po przywróceniu połączenia mastera) pozostawiono w trybie 5 węzłów na dalsze kilka godzin, by obserwować stabilność wydobycia, ewentualne fluktuacje hashrate oraz temperatur. Sprawdzono, czy nie występują zjawiska degradacji wydajności w czasie (np. throttling CPU z powodu temperatur lub spadek hashrate wskutek błędów w oprogramowaniu). Zwracano uwagę na spójność danych – czy wszystkie instancje pozostają zsynchronizowane z siecią Monero, co potwierdzały komunikaty o nowych blokach i transakcjach w logach XMRig.

Zgromadzone dane z powyższych etapów posłużyły do sformułowania wniosków dotyczących zarówno osiągalnej wydajności takiego mikroklastra, jak i jego zachowania w obliczu zakłóceń.

Wyniki i dyskusja

Eksperyment potwierdził, że nawet niewielki klaster Raspberry Pi jest w stanie skutecznie uczestniczyć w procesie wydobycia kryptowaluty Monero. W momencie uruchomienia pierwszego węzła (1 pod na pojedynczym Raspberry Pi) uzyskano moc obliczeniową rzędu ~300 H/s. Włączenie drugiego węzła spowodowało wzrost łącznego hashrate do około 600 H/s, a z trzecim węzłem przekroczono 900 H/s. Ostatecznie, przy pięciu jednocześnie pracujących instancjach górnika, klaster osiągnął szczytowy hashrate wynoszący ok. 1,8 kH/s, ze średnią ustaloną na poziomie 1,66 kH/s. Zależność mocy od

liczby węzłów była bliska liniowej – każda dodatkowa jednostka wnosiła porównywalny przyrost (~300 H/s) do sumarycznej wydajności.



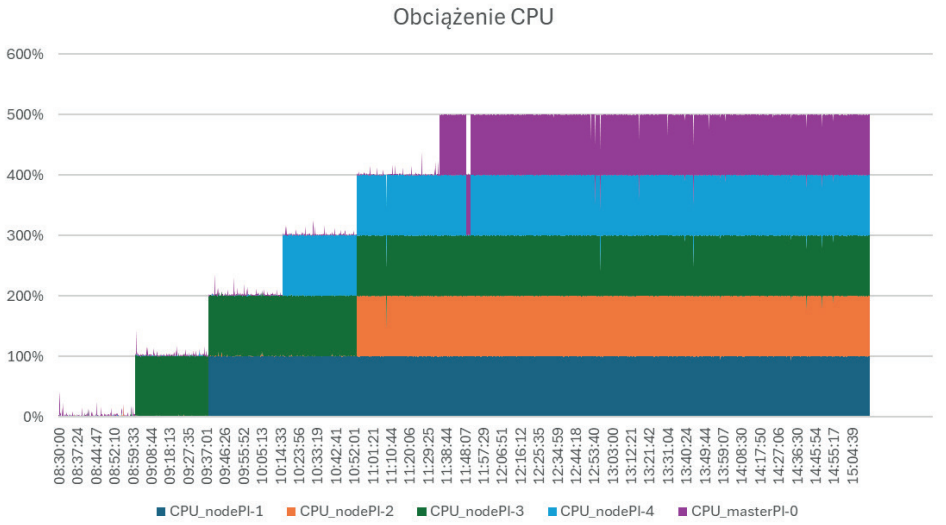
Rysunek 1. Wykres przedstawiający hashrate dla poszczególnych workerów
Źródło: opracowanie własne.

Co istotne, nie zanotowano drastycznych spadków efektywności pojedynczych Raspberry Pi wraz ze wzrostem liczby uczestniczących węzłów, co sugeruje minimalny narzut komunikacyjny przy tak małej skali systemu. Zaobserwowano jednak pewne skokowe zwiększanie hashrate zamiast idealnie płynnej linii, co wynika z sekwencyjnego dołączania węzłów i czasu potrzebnego na ich pełne rozpędzenie (wydajność wzrastała stopniowo po starcie nowej instancji). Podobne zjawisko nieliniowości raportowali również inni badacze dla większych klastrów – wraz z dodawaniem wielu węzłów marginalny przyrost mocy może maleć z uwagi na narzuty koordynacji i ograniczenia sieciowe (Komninos i in., 2019). W opracowanym pięciowęzłowym klastrze efekt degradacji skalowania okazał się jednak pomijalny, co wynika z relatywnie niskiego ruchu sieciowego generowanego przez górników

Monero (komunikaty o nowych blokach i udziały mają małą objętość) oraz wystarczającej przepustowości sieci LAN. W kontekście realnego udziału w sieci Monero, klaster podczas ~12 godzin pracy rozwiązał łącznie 6253 udziały (shares), co odpowiada w przybliżeniu 5000 rzeczywistych transakcji przetworzonych w blockchain (liczba transakcji została oszacowana na podstawie udziału klastra w znajdowaniu bloków w puli). Wynik ten stanowi dowód na pełną funkcjonalność mikroklastra jako węzła obliczeniowego – wszystkie instancje były poprawnie zsynchronizowane z globalną siecią i dokonywały prawidłowej walidacji/obsługi transakcji. Należy jednak podkreślić, że z punktu widzenia wydajności bezwzględnej taka konfiguracja pozostaje daleko w tyle za profesjonalnymi farmami miningowymi. Uzyskany hashrate 1,8 kH/s to ułamek promila globalnej mocy sieci Monero, co przekłada się na marginalne korzyści ekonomiczne. W trakcie testu klaster wygenerował około 0,0001 XMR nagrody, co przy bieżącym kursie (~400 EUR/XMR) odpowiada jedynie ~0,04 EUR – wartość dalece niewystarczającą do pokrycia kosztów energii czy sprzętu. Ten wynik nie jest zaskoczeniem, gdyż celem projektu nie była opłacalność wydobycia, lecz ocena technicznych możliwości i ograniczeń. Potwierdza on jedynie, że w warunkach komercyjnych takie rozwiązanie nie konkuruje z wydajnym sprzętem, lecz znajduje niszę w zastosowaniach eksperymentalnych i edukacyjnych.

Monitorowanie klastra ujawniło istotną korelację pomiędzy obciążeniem obliczeniowym a zużyciem zasobów sprzętowych. Uruchomienie procesu XMRig na danym węźle powodowało natychmiastowy wzrost wykorzystania CPU do blisko 100% na wszystkich czterech rdzeniach oraz zauważalny skok zużycia pamięci (do ok. 250 MB na pod, głównie z tytułu alokacji datasetu RandomX w RAM). Wykres obrazujący obciążenie procesora w czasie pokazuje wyraźne piki w momentach dołączania kolejnych węzłów,

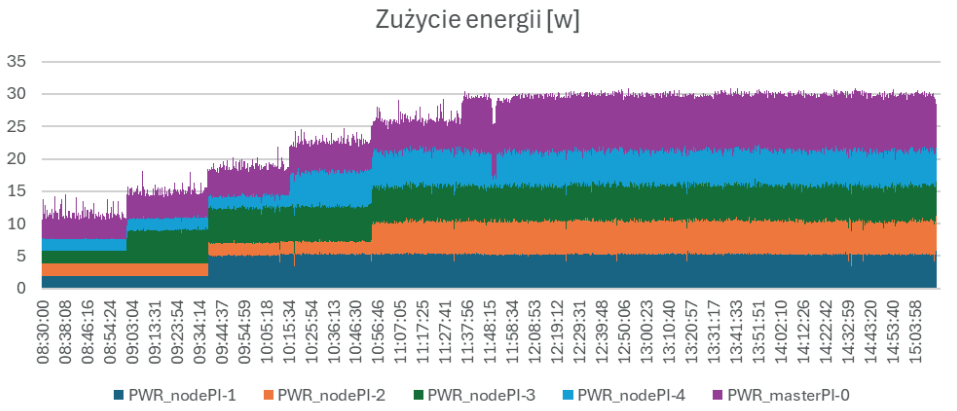
po czym następuje stabilizacja na wysokim poziomie użycia przedstawiono na rysunku 2.



Rysunek 2. Wykres obciążenia CPU podczas uruchomiania i pracy poszczególnych węzłów
Źródło: opracowanie własne.

Każdy nowy węzeł dodaje obciążenie niezależnie od pozostałych, co wskazuje na minimalne wzajemne wpływy procesów górniczych uruchomionych na osobnych urządzeniach. Ciekawym spostrzeżeniem jest drobny efekt synergii – zaobserwowano, że dodanie nowego węzła niekiedy nieznacznie podnosiło hashrate już pracujących sąsiadów (o 1-2%), co można tłumaczyć lepszym wykorzystaniem wspólnych zasobów sieciowych bądź mechanizmami boost w CPU przy pełnym obciążeniu wszystkich rdzeni. Efekt ten jednak mieści się w granicach błędu pomiarowego i wymagałby dalszych badań. Od strony poboru mocy klaster okazał się zaskakująco energooszczędny. W stanie spoczynku pojedyncze Raspberry Pi pobierało ~1,5-2,0 W (przy uruchomionym systemie i usługach pomocniczych, lecz bez aktywnego miningu). Natomiast w trakcie intensywnego wydobycia każdy węzeł zwiększał zużycie energii do średnio ~5 W, a maksymalnie około 6-6,5 W. Wyróżniał się jedynie węzeł nadrzędny (masterPI-0), który jako jedyny dźwigający procesy

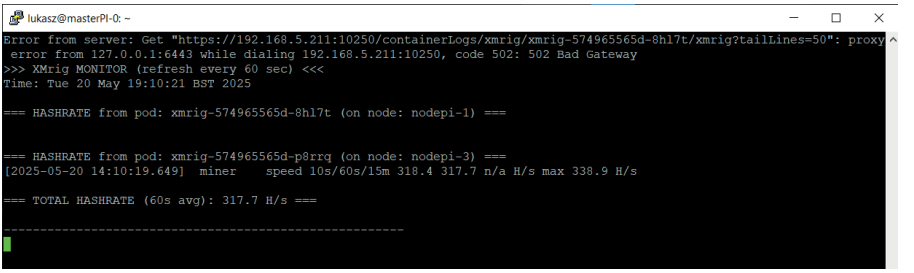
zarządzające klastrem osiągał szczytowo do ~8 W – prawdopodobnie z uwagi na dodatkowe obciążenie związane z obsługą procesu kube-apiserver i koordynacji podów, co zostało przedstawione na rysunku 3.



Rysunek 3. Wykres przedstawiający pobór energii podczas pracy klastra (moc w watach)
Źródło: opracowanie własne.

Łączna konsumpcja mocy całego klastra, z pięcioma aktywnymi węzłami, wyniosła około 24,6 W średnio, przy maksymalnym zaobserwowanym poborze ~33,5 W (krótkotrwałe piki przy pełnym obciążeniu wszystkich jednostek). Oznacza to, że na każdy pobierany wat mocy przypadało w systemie około 50 H/s mocy haszującej. Dla porównania, wysokowydajne koparki GPU czy ASIC osiągają co prawda setki kilohashy na sekundę, ale kosztem kilkuset watów, przez co ich efektywność energetyczna bywa na poziomie 20-30 H/s/W lub niższym. Opracowany mikroklaster okazał się zatem bardzo efektywny energetycznie, co można przypisać energooszczędnej architekturze ARM i relatywnie niskim częstotliwościom pracy. Wynik ~50 H/s/W przewyższa typowe wartości dla starszych CPU używanych w miningowych (np. wiele procesorów x86 osiąga 30-40 H/s/W przy RandomX). Oczywiście, absolutna moc jest niska, ale z punktu widzenia zielonego IT jest to rezultat obiecujący – wskazuje, że rozdrobnione, niskonapięciowe węzły mogą konkurować pod względem efektywności energetycznej z większymi jednostkami. Warto dodać, że dzięki wspomnianemu aktywnemu chłodzeniu,

temperatury wszystkich Raspberry Pi utrzymywały się stabilnie w okolicach 55-60°C pomimo długotrwałego maksymalnego obciążenia. Zapobiegło to zjawisku thermal throttling, czyli ograniczania taktowania CPU przy przegrzewaniu, co oznacza, że klaster pracował cały czas ze szczytową wydajnością bez przerw czy spadków spowodowanych przegrzaniem – a to istotne dla nieprzerwanego i bezpiecznego działania systemu (brak ryzyka uszkodzenia termicznego komponentów). Odporność na awarie i ciągłość działania: Kluczowym elementem badania była symulacja awarii w celu oceny wewnętrznej odporności mikroklastra. Wybrano skrajny scenariusz – utratę łączności z węzłem master – aby przeanalizować zachowanie systemu w warunkach braku centralnej koordynacji. Bezpośrednim skutkiem takiej awarii było zatrzymanie procesu orkiestracji: węzeł nadrzędny przestał być osiągalny dla agentów klastrowych na pozostałych Raspberry Pi. Natychmiastową obserwacją był spadek łącznego hashrate mniej więcej o wartość odpowiadającą wydajności jednego węzła (ok. 300 H/s). Logi XMRig na pozostałych dostępnych węzłach odnotowały utratę jednego z podów – instancja przypisana wcześniej do nodePI-1 (pracująca pod kontrolą mastera) zakończyła pracę i nie została automatycznie zrestartowana gdzie indziej. Wynik przedstawiono na rysunku 4.



Rysunek 4. Logi XMrig, awaria jednego z nodów pracujących w systemie
Źródło: opracowanie własne.

Pool MoneroOcean również zgłaszał brak odpowiedzi od jednego workera. Oznacza to, że awaria węzła zarządzającego spowodowała utratę części zadań obliczeniowych klastra, co było oczekiwane, ale istotne jest, że reszta systemu kontynuowała pracę. Trzy z czterech pozostałych podów (na

nieuszkodzonych węzłach) dalej wydobywały Monero, utrzymując ~75% poprzedniego hashrate. Nie zaobserwowano rozproszenia czy desynchronizacji – te węzły w dalszym ciągu komunikowały się z p2p siecią Monero i pool'em, co świadczy o pewnej rezyliencji: pojedyncza awaria nie zatrzymała całkowicie działania usługi blockchain. Niemniej, test ujawnił poważne ograniczenie: w obecnej konfiguracji Kubernetes (k3s) nie podjął próby automatycznej re-schedulacji utraconego poda na innym dostępnym węźle. Innymi słowy, klaster nie zaadaptował się do awarii przez uruchomienie brakującej instancji górnika np. na jednym z działających Raspberry Pi. Taka sytuacja wynika prawdopodobnie z faktu, że w architekturze k3s z jednym masterem, utrata mastera w praktyce „zamraża" funkcję kontrolera – bez niego nowe decyzje odnośnie podów nie są podejmowane. W typowych wdrożeniach produkcyjnych Kubernetes zaleca się konfigurację wysokiej dostępności (HA) dla warstwy zarządzającej, np. poprzez kilkuetapowy klaster masterów, tak by awaria jednego nie wstrzymywała całego systemu (DeCarlo, 2018). W eksperymencie celowo pominięto tę komplikację, aby ocenić najgorszy przypadek. Wynik wskazuje, że aby mikroklaster był rzeczywiście odporny na uszkodzenia, należałoby zaimplementować mechanizmy redundancji w warstwie orkiestracji lub zastosować alternatywne rozwiązania (np. lekkie schedulery rozproszone oparte na konsensusie w stylu Raft, albo ręczne przełączenie roli mastera na inny węzeł). Pomimo braku automatycznego self-healingu, pozytywnym aspektem jest, że pozostałe węzły utrzymały swoje zadania i po przywróceniu połączenia z masterem system mógł zostać przywrócony do pełnej funkcjonalności bez restartu całości. W praktyce po ponownym dołączeniu węzła nadrzędnego klastra, brakujący pod XMRig został ponownie zdeployowany (zgodnie z deklarowanym stanem 5 replik) i hashrate powrócił do poprzedniego poziomu. Incydent ten pokazał jednak, że obecny mechanizm nie gwarantuje ciągłości pełnej wydajności w momencie awarii – klaster działa dalej, ale w trybie zdegradowanym. Z punktu widzenia bezpieczeństwa systemów krytycznych, gdzie często zakłada się tolerancję na awarię N-1 elementów bez utraty funkcji, należałoby ten mankament wyeliminować. Możliwe sposoby to:

- wprowadzenie zapasowego mastera,

- wykorzystanie bardziej autonomicznych agentów (każdy węzeł sam utrzymuje zadanie niezależnie),
- architektura leader election pomiędzy węzłami.

Podsumowując testy niezawodności, można stwierdzić, że mikrosystem oparty o Raspberry Pi jest w stanie kontynuować pracę nawet w warunkach częściowej awarii, ale wymaga pewnych usprawnień by osiągnąć pełną odporność systemową. Warto zauważyć, że sama sieć blockchain Monero jako całość pozostała niewzruszona utratą jednego węzła – z perspektywy globalnej decentralizacji był to incydent bez znaczenia. To podkreśla, że odporność na poziomie makro (całej sieci blockchain) jest wysoka, natomiast odporność na poziomie mikro zależy od wewnętrznej architektury. W kontekście zabezpieczania takich mikroklastrów przed awariami, należałoby także rozważyć kwestie zasilania awaryjnego (UPS dla każdego node'a lub przynajmniej mastera) – w przeprowadzonym badaniu brak zasilania zapasowego oznaczałby natychmiastowe przerwanie działania całego klastra w razie zaniku prądu. Dla infrastruktury krytycznej byłby to poważny problem, stąd w realnych wdrożeniach konieczne jest projektowanie warstw redundancji zarówno w sprzęcie, jak i oprogramowaniu.

Analizując aspekty bezpieczeństwa, poza odpornością na awarie fizyczne, warto odnieść się do innych wymiarów bezpieczeństwa takiego rozwiązania. W trakcie eksperymentu skupiono się na wydajności, jednak pewne obserwacje można poczynić. Po pierwsze, klaster operował w środowisku lokalnym z zaufaną siecią – w otwartym środowisku produkcyjnym należałoby zadbać o zabezpieczenia sieciowe (firewalle, segmentacja sieci dla węzłów, szyfrowanie ruchu między nimi) aby zapobiec atakom na węzły (np. przejęcie kontroli nad którymś Raspberry Pi mogłoby posłużyć do manipulacji wynikami wydobycia lub jako przyczółek do ataków na inne systemy). Same protokoły Monero są już zbudowane tak, by zapewnić integralność i uwierzytelnienie (transakcje podpisane kryptograficznie, konsensus PoW chroniący przed nadużyciami), więc atakujący nie mógłby łatwo zmienić danych w blockchainie. Mógłby jednak próbować zakłócić pracę klastra – np. przez atak DoS na węzeł master lub zapełnienie go złośliwymi podami w Kubernetes. Te wektory można ograniczyć, stosując odpowiednie polityki

bezpieczeństwa K8s (ograniczenia dostępu, monitorowanie anomalii). Drugim aspektem jest bezpieczeństwo fizyczne i niezawodność komponentów: Raspberry Pi nie są urządzeniami klasy przemysłowej o podwyższonej niezawodności, dlatego w zastosowaniach produkcyjnych należy liczyć się z możliwością częstszych usterek (np. zużycie karty SD, awarie zasilania). Badanie wskazuje, że nawet przy intensywnej pracy i braku przerw, wszystkie jednostki działały stabilnie (co potwierdza dojrzałość RPi 5 jako platformy). Niemniej do systemów, gdzie bezpieczeństwo jest krytyczne (np. przetwarzanie danych medycznych, sterowanie ruchem), dobór sprzętu powinien uwzględniać te czynniki – być może lepszym rozwiązaniem byłyby tu dedykowane moduły obliczeniowe Raspberry Pi z bardziej niezawodną pamięcią eMMC i zasilaniem redundantnym. Porównując rozwiązanie z tradycyjnymi infrastrukturami, interesującym ćwiczeniem jest porównanie zaproponowanego mikroklastra z klasycznymi serwerami stosowanymi jako węzły blockchain. Typowy węzeł Monero uruchomiony na serwerze x86 (np. 8-rdzeniowym) mógłby osiągnąć kilkukrotnie wyższy hashrate niż pojedynczy Pi, ale kosztem również wyższego zużycia energii (rzędu 100 W). Aby uzyskać ~1,6 kH/s jak opracowany klaster, wystarczyłby współczesny procesor klasy desktop, jednak wówczas występuje pojedynczy punkt awarii – uszkodzenie tego serwera zatrzymuje cały węzeł. W opracowanym rozwiązaniu awaria pojedynczego Raspberry Pi redukuje wydajność, ale nie eliminuje systemu całkowicie, co jest cenne z perspektywy odporności (system degraduje się stopniowo, a nie binarnie). Dodatkowo mikroklaster można łatwo rozproszyć geograficznie (np. łącząc węzły przez VPN), zwiększając odporność na awarie lokalne (pożar, zalanie jednej serwerowni itp.). Jest to zgodne z duchem decentralizacji – wiele małych node'ów zamiast jednego dużego. Zgodnie z badaniami (Govea i in., 2024), integracja blockchain z rozproszonymi węzłami IoT może znacząco podnieść odporność i bezpieczeństwo systemów w smart cities, choć wymaga pokonania wyzwań skalowalności i standaryzacji. Otrzymane w przeprowadzonym badaniu wyniki wpisują się w ten obraz – pokazują wykonalność takiej integracji na małą skalę oraz wskazują obszary wymagające dalszych usprawnień (automatyzacja redundancji, zabezpieczenia przed awariami zasilania, ochrona przed atakami sieciowymi na poziomie orkiestracji).

Podsumowanie

Przeprowadzone badanie wykazało, że mikroklaster Raspberry Pi może z powodzeniem pełnić rolę węzła w zdecentralizowanej sieci blockchain takiej jak Monero. Pomimo ograniczeń sprzętowych, pięć współpracujących minikomputerów było w stanie osiągnąć zauważalną moc obliczeniową ~1,6 kH/s i utrzymać stabilne wydobycie kryptowaluty przez dłuższy czas. Otrzymane wyniki dowodzą, iż nawet przy minimalnych zasobach obliczeniowych możliwe jest stworzenie w pełni funkcjonalnego, rozproszonego systemu zdolnego do integracji z globalnym blockchainem (Monero) – potwierdzając tym samym zasadność założeń algorytmu RandomX o inkluzywności słabszych urządzeń. Z perspektywy inżyniersko-technologicznej, klaster ujawnił bardzo dobrą skalowalność w małej skali – wydajność rosła niemal liniowo wraz z dodawaniem kolejnych węzłów, a narzut związany z koordynacją nie zdławił mocy obliczeniowej. Efektywność energetyczna okazała się wręcz atutem – przy łącznym poborze mocy ~30 W klaster uzyskał wskaźnik ~50 H/s na wat, co czyni go konkurencyjnym wobec wielu tradycyjnych konfiguracji pod względem hashowania na jednostkę energii. Takie rozwiązanie jest zatem atrakcyjne w kontekście zastosowań, gdzie dostępna jest ograniczona moc lub gdzie priorytetem jest niski koszt operacji (np. zasilanie z paneli słonecznych w odległych lokalizacjach). Modularność i łatwość skalowania mikroklastra pozwala elastycznie dostosować jego wydajność do potrzeb – można zwiększyć liczbę węzłów dla większej mocy lub redukować dla oszczędności energii, bez konieczności inwestycji w drogi sprzęt. Z punktu widzenia bezpieczeństwa i niezawodności, wyniki wskazują na pewną wrodzoną odporność takiego rozproszonego węzła, ale i obszary wymagające usprawnienia. Pozytywne jest to, że awaria pojedynczego węzła nie eliminuje całkowicie funkcji – pozostałe węzły kontynuują pracę i utrzymują sieć przy życiu. W kontekście systemów krytycznych oznacza to potencjalnie większą ciągłość działania w porównaniu z jednolitą jednostką obliczeniową. Jednak brak automatycznego przejmowania zadań przez inne węzły w razie awarii ujawnia, iż obecna architektura nie spełnia rygorystycznych wymogów wysokiej dostępności. Aby mikroklaster mógł być uznany za w pełni odporny, konieczne byłoby zastosowanie mechanizmów redundancji

w warstwie zarządzania (np. klaster HA dla Kubernetes) oraz zabezpieczenie przed awariami zasilania i komunikacji. W praktyce, biorąc pod uwagę wyniki, rekomenduje się:

1. Wprowadzenie redundancji mastera: co najmniej dwa węzły Raspberry Pi powinny pełnić rolę współdzieloną zarządzającą (np. wykorzystując zewnętrzny etcd i tryb HA Kubernetes), by awaria jednego nie zatrzymała harmonogramu podów.
2. Mechanizmy autoskalowania i autorestartu: włączyć polityki Pod Disruption Budget i restartPolicy tak, aby w razie wykrycia niedziałającego poda, klaster próbował go odtworzyć na innym dostępnym węźle.
3. Zabezpieczenia fizyczne: zastosować zasilanie awaryjne (UPS) przynajmniej dla głównych elementów, a także monitoring sprzętu (watchdogi) by zautomatyzować restart zawieszonych jednostek.
4. Aktualizacje i hardening: regularnie aktualizować oprogramowanie (zarówno system Raspberry Pi OS jak i kontenery), a także stosować zasady hardeningu (np. uruchamianie kontenerów z najmniejszymi uprawnieniami, izolacja sieciowa) by minimalizować ryzyko ataków.

Mikroklastry oparte na tanich urządzeniach jak Raspberry Pi mogą znaleźć zastosowanie w wielu dziedzinach jako tania i elastyczna platforma blockchain. W środowiskach badawczych i dydaktycznych umożliwiają studentom i inżynierom eksperymentowanie z prawdziwymi sieciami blockchain w skali laboratoryjnej, bez konieczności posiadania kosztownego sprzętu – rozwiązanie udowodniło, że nawet wymagająca sieć Monero może działać na takim klastrze, co otwiera drogę do badań nad optymalizacją algorytmów konsensusu czy zarządzania energią w sieciach blockchain. W obszarze IoT, gdzie miliardy urządzeń wymieniają dane i wymagają zabezpieczonej rejestracji zdarzeń, integracja lekkich węzłów blockchain z urządzeniami brzegowymi może zapewnić niezawodną i odporną na manipulacje platformę wymiany informacj. Klaster Raspberry Pi mógłby pełnić rolę lokalnego rejestratora transakcji np. w inteligentnej sieci energetycznej czy systemie czujników miejskich, zapewniając ciągłość działania nawet przy awarii części sensorów. Dodatkowo, architektury odporne na awarie przy użyciu taniej infrastruktury stają się możliwe do wdrożenia w miejscach, gdzie dotąd barierą

był koszt – np. w małych gminach do prowadzenia lokalnych rejestrów (kataster, rejestry mieszkańców) w sposób rozproszony i odporny na utratę danych. Badanie pokazuje, że technicznie jest to realne, choć wymaga dbałości o szczegóły implementacyjne. Na koniec warto zaznaczyć, że trend polegający na demokratyzacji dostępu do technologii blockchain poprzez wykorzystanie urządzeń powszechnie dostępnych i energooszczędnych wpisuje się w szerszy obraz zmian w IT. Wraz z postępem technologicznym, różnica między profesjonalnym sprzętem a hobbystycznym zaciera się – Raspberry Pi 5 jest kilkukrotnie wydajniejszy od swoich poprzedników (Komninos i in., 2019) i przyszłe generacje mogą jeszcze bardziej zwiększyć tę wydajność. Można sobie wyobrazić, że za kilka lat mikroklastry staną się jeszcze potężniejsze, umożliwiając np. działanie lokalnych mini-sieci blockchain obsługujących konkretne społeczności czy zadania (np. wymiana energii między sąsiadującymi domami w ramach smart grid). Już dziś jednak, jak wykazano, nawet obecne rozwiązania pozwalają budować działające, choć niewielkie, „ekosystemy blockchain" w skali mikro, które odzwierciedlają cechy dużych systemów – decentralizację, odporność na cenzurę i (w pewnym stopniu) odporność na awarie – za ułamek kosztów i z minimalnym zużyciem energii.

Bezpieczeństwo i odporność mikroklastrów Raspberry Pi w sieciach blockchain wydaje się obiecujące, o ile uwzględni się specyfikę tych platform. Monero jako studium przypadku udowodniło, że prywatna i zdecentralizowana kryptowaluta może być obsługiwana przez tani, rozproszony sprzęt, co potwierdza zamierzenia jej twórców co do egalitarności protokołu. Jednocześnie doświadczenia wyniesione z testów awaryjnych podkreślają, że osiągnięcie pełnej niezawodności wymaga uzupełnienia architektury o mechanizmy tolerancji uszkodzeń. Niniejsza praca dostarcza podstaw zarówno inżynierom, jak i badaczom bezpieczeństwa do dalszego rozwijania koncepcji odpornych mikro-infrastruktur blockchain – czy to w celu zabezpieczenia krytycznych danych, czy zapewnienia ciągłości działania rozproszonych usług w skali lokalnej. W dobie rosnących zagrożeń cybernetycznych, rozwiązania łączące blokową odporność (blockchain) z modułową redundancją (mikroklastry) mogą odegrać istotną rolę w kształtowaniu przyszłych, bardziej odpornych systemów informacyjnych.

Literatura

1. Govea, J., Gaibor-Naranjo, W., Villegas-Ch., W. (2024). Securing Critical Infrastructure with Blockchain Technology: An Approach to Cyber-Resilience. *Computers, 13*(5), 122. https://doi.org/10.3390/computers13050122
2. Nofer, M., Gomber, P., Hinz, O., Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering, 59*(3), 183-187. https://doi.org/10.1007/s12599-017-0467-3

Netografia

1. DeCarlo, P. (2018). *Cryptocurrency Node Cluster with K8s on Raspberry Pi.* Hackster.io. Pobrane z: https://www.hackster.io/pjdecarlo/cryptocurrency-node-cluster-with-k8s-on-raspberry-pi-696933 (data dostępu: 10.12.2025).
2. Hoskinson, C. (2021). *Why Monero Is the Privacy King of Cryptocurrencies*. Medium (data dostępu: 10.12.2025).
3. Komninos, A., Simou, I., Gkorgkolis, N., Garofalakis, J. (2019). *Performance of Raspberry Pi Micro-Clusters for Edge Machine Learning in Tourism*. In Proc. AmI-2019. CEUR Workshop Proceedings, 2492, 8-18. Pobrane z: http://ceur-ws.org/Vol-2492/paper2.pdf (data dostępu: 10.12.2025).
4. Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. (White paper). Pobrane z: https://bitcoin.org/bitcoin.pdf (data dostępu: 10.12.2025).
5. Raspberry Pi Foundation. (2023). *Introduction to Raspberry Pi 5 and Its New Features. Raspberry Pi Documentation*. Pobrane z: https://www.raspberrypi.com/news/introducing-raspberry-pi-5 (data dostępu: 10.12.2025).
6. Trail of Bits. (2019). *State of the Art Proof-of-Work: RandomX*. Trail of Bits Blog. Pobrane z: https://blog.trailofbits.com/2019/07/02/state-of-the-art-proof-of-work-randomx (data dostępu: 10.12.2025).