

Wydział Fizyki Astronomii i
Informatyki Stosowanej

Kraków, 2017

Autor: Janusz Majchrzak



Seminarium licencjackie:
*Przyśpieszenie obliczeń przy pomocy programowania
równoległego i rozproszonego*

Opiekun pracy: dr Wojciech Palacz

Plan prezentacji:

- Wstępny zarys tematyki
- Motywacja do wybrania tego tematu
- Dostępne technologie
- Opis problemu do rozwiązania
- Podsumowanie



Wydajność

- Kiedy jej potrzebujemy?
- Gdzie jej potrzebujemy?
- Czy zawsze tak samo ją rozumiemy?
- Czy zawsze jest najważniejsza?
- Kompromisy
- Opłacalność
- Trudność



Przyspieszanie naszej aplikacji

Sposoby NIE wymagające zmiany istniejącego kodu

- Aktualizacja / zmiana narzędzi deweloperskich lub środowiska
- Wbudowane mechanizmy optymalizacji
- Wymiana komponentów

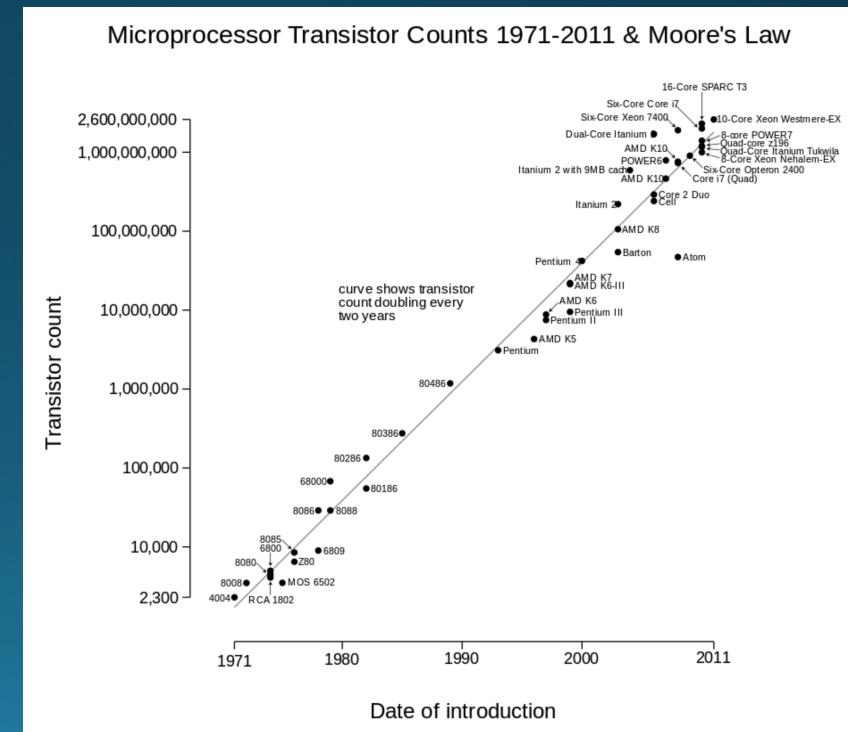
Opłacalność

Sposoby Planowane

- Przemyślana architektura aplikacji
- Zastosowanie optymalnych algorytmów i struktur danych
- Korzystanie z języków niższego poziomu
- Wielowątkowość
- Rozpraszania na inne maszyny

Prawo Moore'a

- Prawo Moore'a – prawo empiryczne, wynikające z obserwacji, że ekonomicznie optymalna liczba tranzystorów w układzie scalonym zwiększa się w kolejnych latach zgodnie z trendem wykładniczym (podwaja się w niemal równych odcinkach czasu).

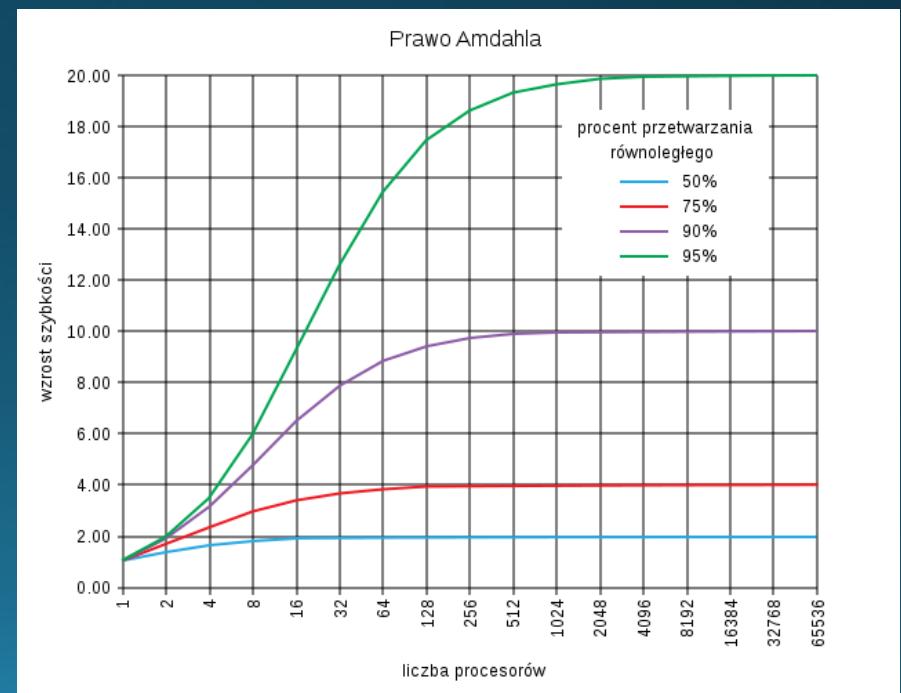


Prawo Amdahla

„(...) Zwiększenie szybkości wykonywania się programu przy użyciu wielu procesorów w obliczeniach równoległych jest ograniczane przez czas potrzebny do sekwencyjnego dzielenia programu. ”

W przypadku zrównoleglania, Prawo Amdahla mówi, że jeśli P jest proporcją programu, który może podlegać zrównolegleniu (np. korzyści z wykonywania równoległego) i $(1 - P)$ jest proporcją części, która nie może zostać zrównoleglona (pozostaje w przetwarzaniu szeregowym), wówczas maksymalne przyspieszenie jakie może być uzyskane przy użyciu N procesorów jest równe:

$$\frac{1}{(1 - P) + \frac{P}{N}}$$



Prawo Gustafsona

Prawo Gustafsona (znane także jako **prawo Gustafsona-Barsisa**) jest prawem w inżynierii komputerowej, które stanowi, że każdy wystarczająco duży problem może być efektywnie zrównoległy. Prawo Gustafsona jest ściśle związane z prawem Amdahla, które określa limit przyspieszenia spowodowanego zrównolegleniem. Zostało po raz pierwszy sformułowane przez J. Gustafsona w 1988 roku

$$S(P) = P - \alpha \cdot (P - 1)$$

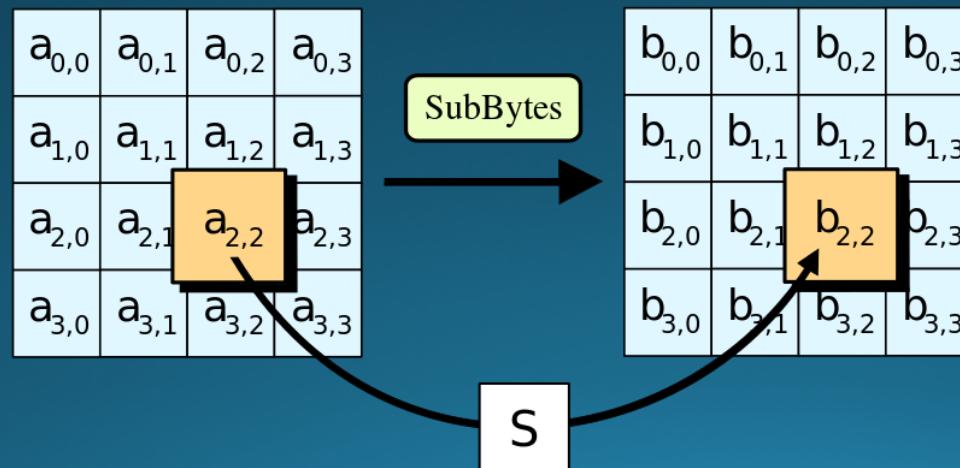
gdzie P jest liczbą procesorów, S jest przyspieszeniem a α częścią procesu której nie da się zrównoleglić.

Advanced Encryption Standard (AES)

AES (ang. *Advanced Encryption Standard*), nazwa oryginalna: **Rijndael** – symetryczny szyfr blokowy przyjęty przez NIST jako standard FIPS-197 w wyniku konkursu ogłoszonego w roku 1997. W 2001 roku został przyjęty jako standard.

AES jest oparty na algorytmie Rijndael'a, którego autorami są Belgijscy kryptografowie, Joan Daemen i Vincent Rijmen. Zaprezentowali oni swoją propozycję szyfru Instytucji NIST w ramach ogłoszonego konkursu. Rijndael jest rodziną szyfrów o różnych długościach klucza oraz różnych wielkościach bloków.

W przypadku AES'a, NIST wybrał trzy algorytmy z rodziny Rijndaela, z których każdy miał tą samą wielkość bloku (128 bitów), ale miały różne długości klucza: 128, 192 i 256 bitów.



Dostępne technologie



openstack.



kubernetes
by Google



docker



Java™



OpenMPI + BOOST

- Wysoka wydajność
- Kontrola nad procesem rozpraszania oraz zarządzania zasobami
- Prostota implementacji
- Wsparcie społeczności
- Rozwiązania powszechnie w przemyśle
- Dzięki wykorzystaniu Boost'a obiektowe API

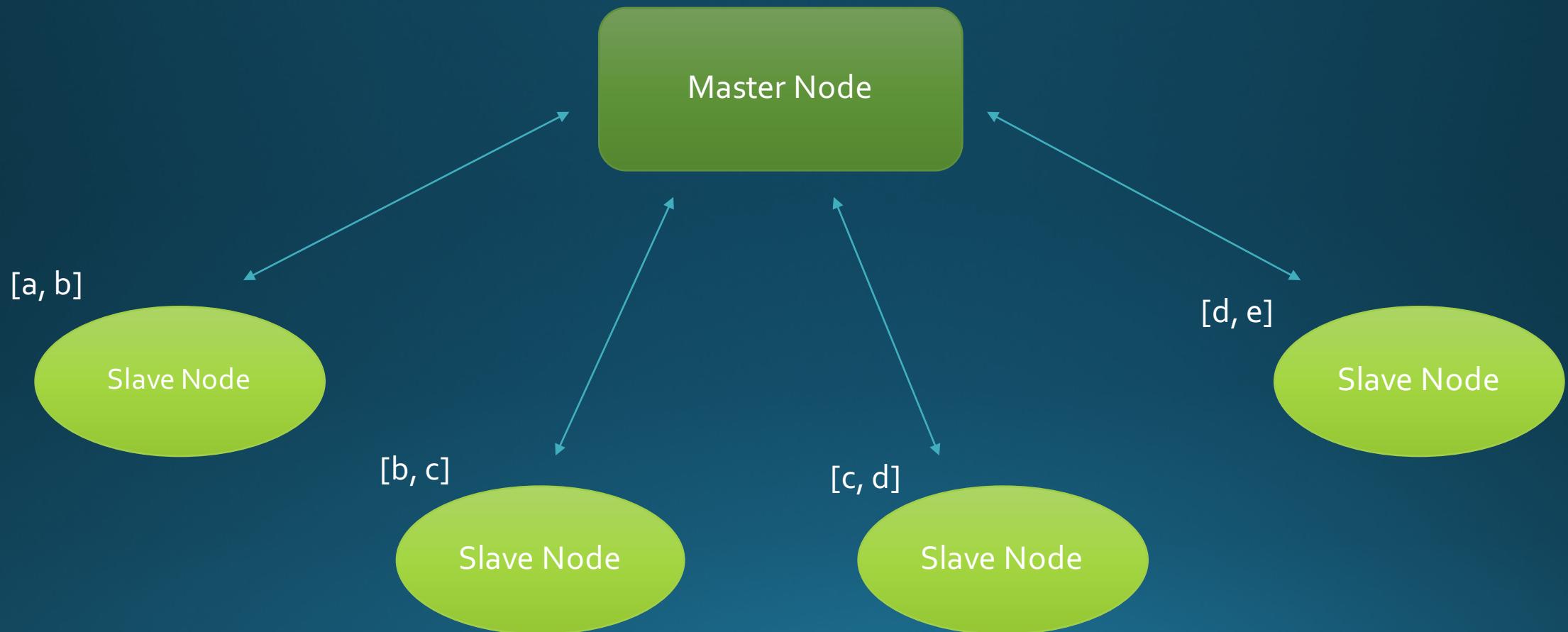
- http://www.boost.org/doc/libs/1_64_0/doc/html/mpi.html
- <https://www.open-mpi.org>

Distributed Decryption System

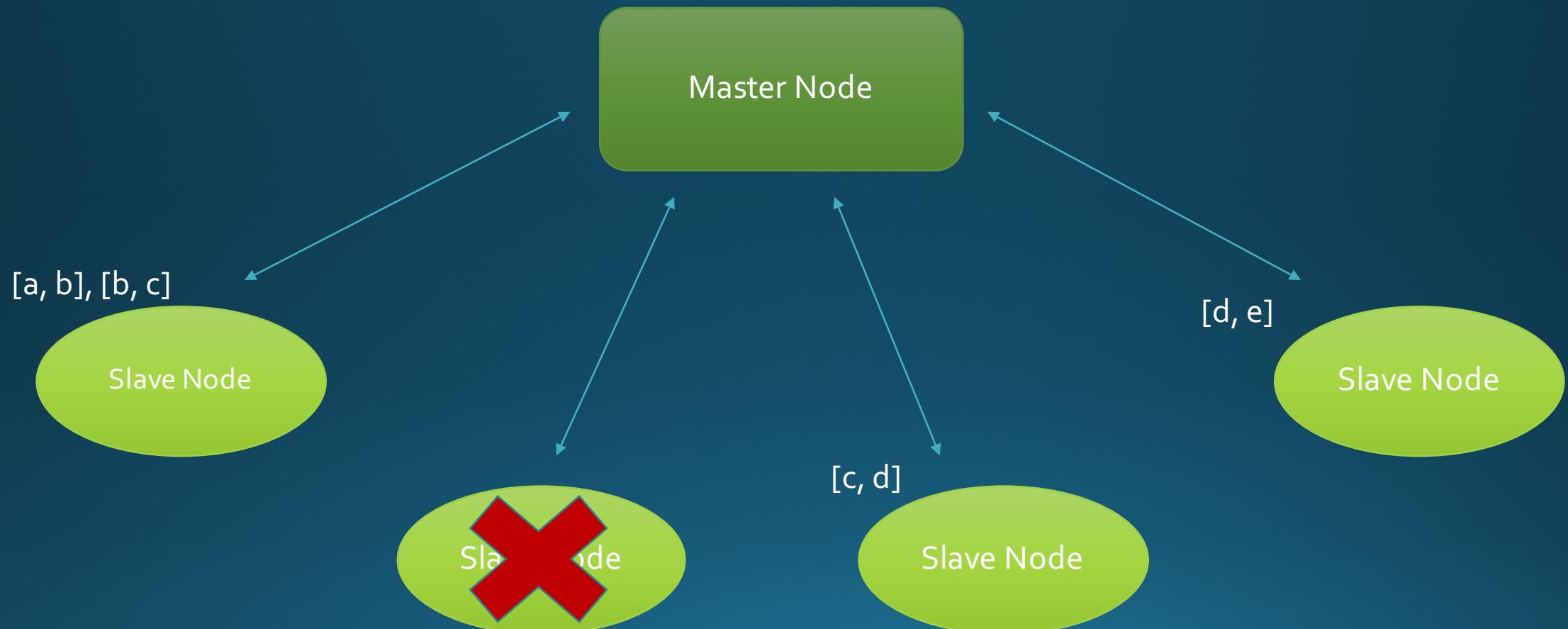
- Modularność – Interfejs przystosowany do tego, aby programista mógł sam pisać moduły odpowiedzialne za przetwarzanie danych
- Skalowalność – System łatwo skaluje się w taki sposób aby wykorzystać możliwie całą moc obliczeniową klastra
- Łatwość w instalacji na maszynach docelowych – dzięki CMake oraz Conan
- Napisany w oparciu o bibliotekę BOOST oraz nowoczesnego C++



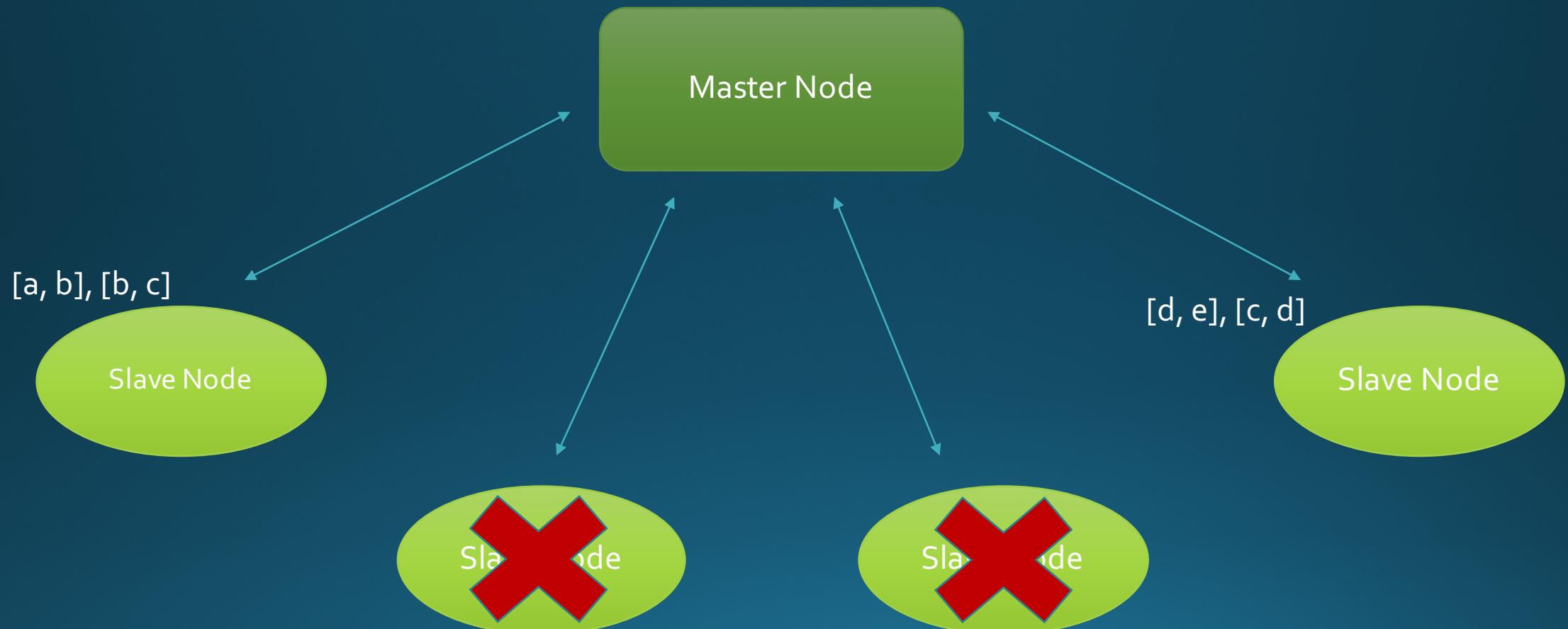
Distributed Decryption System



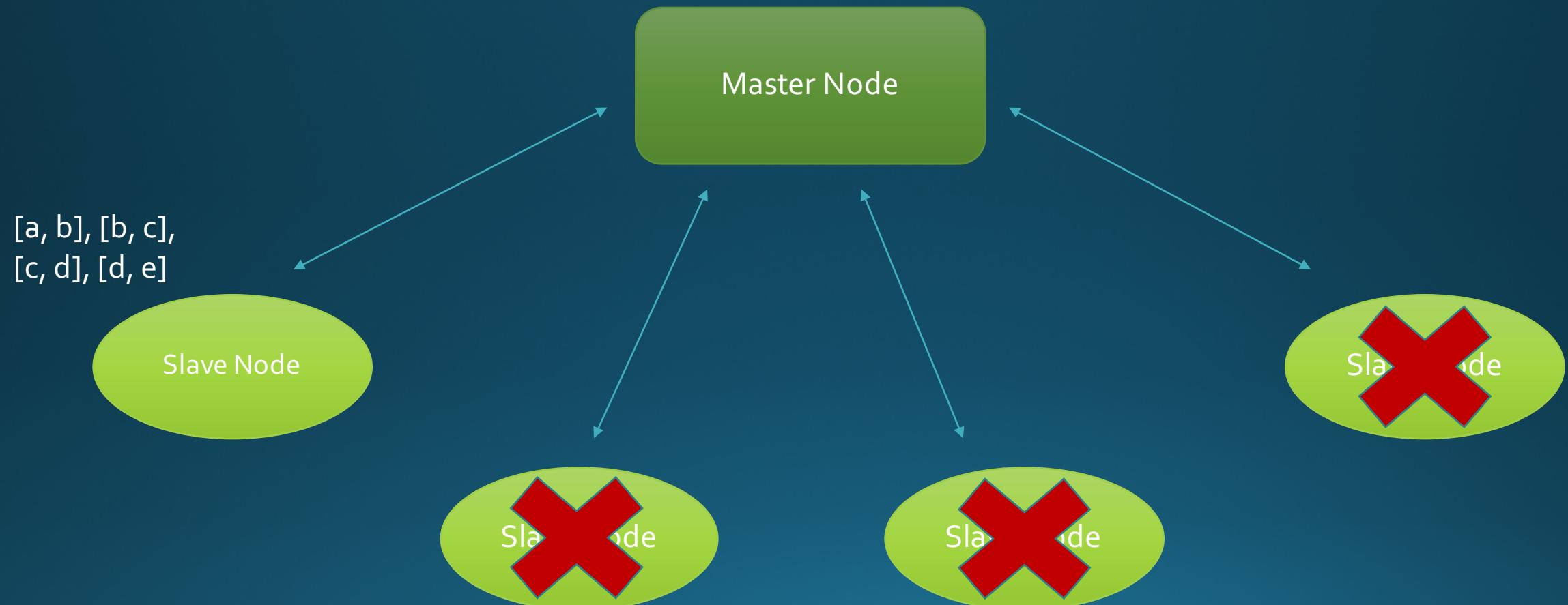
Distributed Decryption System



Distributed Decryption System



Distributed Decryption System



Harmonogram

- Zbieranie informacji (styczeń – luty)
- Prototypowanie kawałków aplikacja (marzec)
- Prototypowanie logiki aplikacji (marzec)
- Okrojona wersja Alfa (kwiecień – maj)
- Testy na prywatnym klastrze RPi (kwiecień – maj)
- Wersja Finalna (maj – czerwic)
- Testy na komputerach pracowni (maj)
- Pisanie tekstu pracy (maj - czerwiec)



*Finalny termin oddania pracy może ulec zmianie

Bibliografia

- <https://www.open-mpi.org>
- https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- https://pl.wikipedia.org/wiki/Prawo_Moore'a
- https://pl.wikipedia.org/wiki/Prawo_Amdahla
- https://pl.wikipedia.org/wiki/Prawo_Gustafsona
- <http://www.boost.org>

Dziękuję za uwagę