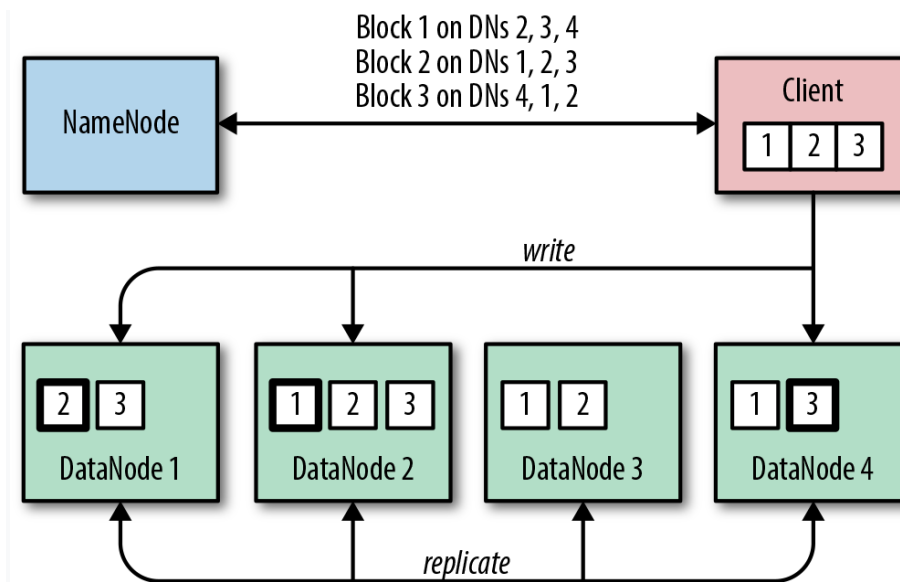


Tier 1 (4 pkt)

- Przygotuj i zaprezentuj zadanie umieszczone w Jupyter Notebook Task 1
- Obliczenie liczby Pi + równoległe sortowanie przez scalanie (szczegóły w noteboku)

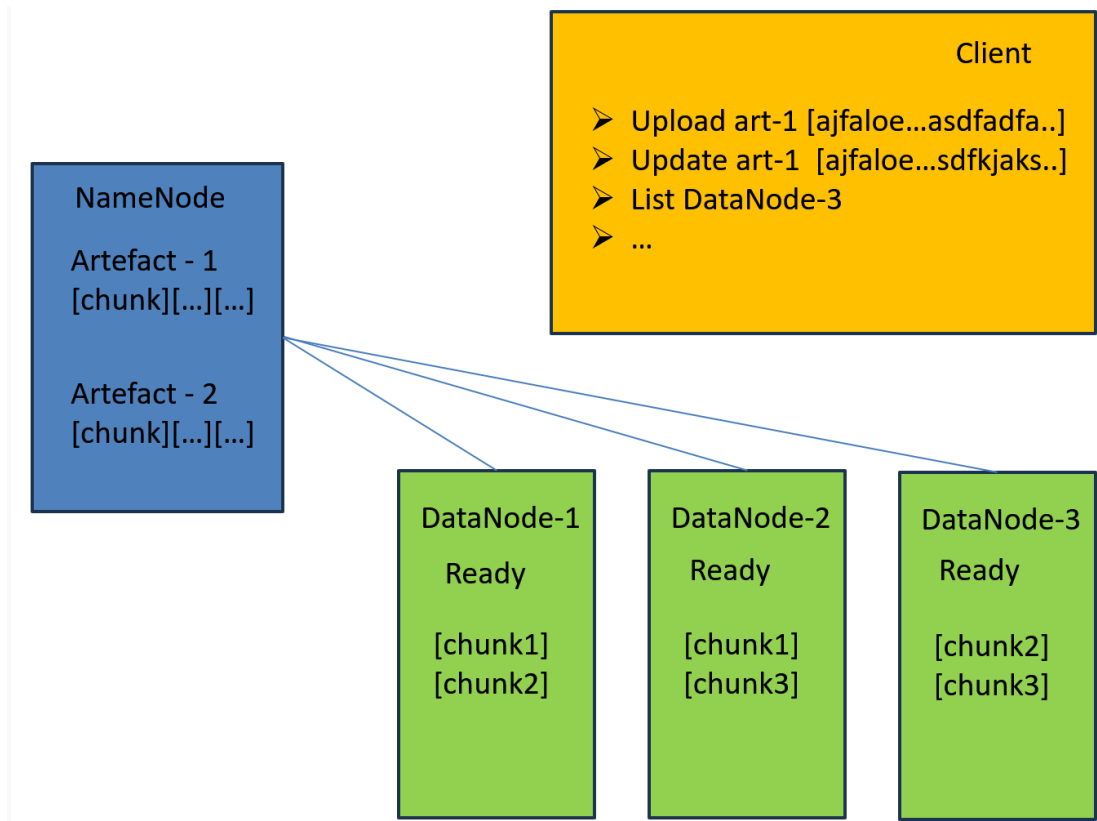
Tier 2 (6-9 pts)

- Zaimplementuj rozproszony system do przechowywania artefaktów na bazie rozproszonego systemu plików
- Schemat HDFS poniżej



- Artefakt składa się z nazwy i długiego napisu, który jest rozbijany na węzły
- Jest jeden name-node i kilka storage-node - wszystkie węzły to aktorzy
- *Całość implementacji może być przedstawiona w Jupyter Notebook'u*
- Aktor name-node dystrybuje artefakty do węzłów storage-node w chunkach/blokach i monitoruje je
- Istnieje kilka kopii takich bloków w systemie (aby poprawić wysoką dostępność i transfer danych - podobnie jak w HDFS)
- Węzły storage-node mogą ulec awarii (zmiana statusu); węzeł name-node śledzi wszystkie fragmenty i zarządza kopiami (wszystkie dane, operacje i stan systemu powinny być spójne)

- Wszystkie węzły powinny mieć metody dostarczające możliwość wylistowanie najważniejszych danych (np. gdzie przechowywane są chunki/bloki)
- Poniżej schemat systemu



- Podczas oddawania zadania proszę o przygotowanie następujących testów. Użytkownik ma możliwość: stworzenia, aktualizacji, skasowania, pobrania artefaktu. Operacje powinny być weryfikowane przez wylistowanie stanu węzłów.

Punktacja Tier 2:

- Bazowy scenariusz opisany wyżej : 6-7 pkt
- Dołożenie jednego elementu z Ray Actor API o którym nie mówiliśmy na zajęciach (<https://docs.ray.io/en/latest/ray-core/actors.html>): +1 pkt
- Dołożenie rozproszenie środowiska (patrz niżej): +1 pkt

Tier 3 (9-10 pkt)

- Przygotuj przykładowy projekt realizujący scenariusz ML w wykorzystaniem modułów Ray AI Runtime

- W projekcie należy wykorzystać w miarę unikalne zbiory danych (raczej nie: MNIST, irysy itp.)
- Należy użyć 3 różnych modułów, i w miarę możliwości przygotować testy/ewaluacje
- Jakiekroki były potrzebne aby przenieść projekt do Ray AIR, jakie były problemy

Punktacja Tier 3:

- Przygotowania i prezentacja projektu: 9 pkt
- Dołożenie rozproszenie środowiska (patrz niżej): +1 pkt

Środowisko rozproszone:

- Przygotowane zadanie należy uruchomić w środowisku rozproszonym (może być elementarne środowisko: 1 head node + 1 worker)
- Można użyć docker-compose (<https://github.com/MarvinSt/ray-docker-compose>), trzeba tylko podmienić wersję pythona i Ray Framework, oraz być może zmienić środowisko w `.env`
- Można użyć kubernetes (<https://docs.ray.io/en/latest/cluster/kubernetes/index.html>), klaster lokalny lub w chmurze
- Można też uruchomić w chmurze (<https://docs.ray.io/en/latest/cluster/vms/user-guides/launching-clusters/index.html>) - ta opcja wydaje się być najbardziej złożona