

PROCEDURA TESTÓW WYDAJNOŚCIOWYCH

Załącznik do CZĘŚĆ IV SWZ

Ogólne założenia dotyczące przeprowadzenia testów wydajnościowych:

1. **Wydajność zaoferowanych elementów infrastruktury** wskazanych w SWZ musi zostać określona poprzez zamieszczenie usrednionych wyników poszczególnych testów w formularza technicznym oferty oraz w załączniku do formularza technicznego, w którym muszą znaleźć się zarówno cząstkowe wyniki poszczególnych tur testu dla poszczególnych typów testów jak i wyniki usrednione dla poszczególnych typów testów
2. Ponadto **poprawność wykonania testów wydajnościowych dla zaoferowanych elementów infrastruktury** musi zostać potwierdzona poprzez zaprezentowanie /udostępnienie – na wezwanie Zamawiającego, na etapie oceny ofert - wydruku wyników testów wydajnościowych wraz z opisem środowiska testowego – patrz take ppkt 9 a) niniejszego punktu;
3. **Testy wydajnościowe** muszą zostać przeprowadzone dla wszystkich wskazanych elementów zaoferowanych systemów i na wszystkich wskazanych poniżej poziomach:
 - a. **Elementy / bloki systemu składowania dla obliczeń HPC (scratch):**
 - i. Poziom blokowy
 - ii. Poziom plikowy
 - b. **Elementy / bloki systemu składowania danych użytkowników i projektów (project data):**
 - i. Poziom plikowy
 - ii. Poziom obiektowy
4. **Testy wydajnościowe na poziomie blokowym** muszą zostać przeprowadzone dla wszystkich typów komponentów dostarczanej infrastruktury systemu składowania dla obliczeń HPC (scratch), tj.:
 - a. bloku pamięci dyskowych (dla OSS/OST), wyposażonego w pamięci dyskowe i elementy przeznaczone do akceleracji I/O (moduły / podsystem pamięci flash)
 - b. bloku pamięci flash (dla MDS/MDT) - wyposażonego w pamięci flash do składowania meta-danych systemu;
5. **Testy wydajnościowe na poziomie plikowym** muszą zostać przeprowadzone osobno dla wszystkich typów komponentów dostarczanej infrastruktury, włączając:
 - a. System składowania dla obliczeń HPC (scratch) - system plików skonfigurowany z wykorzystaniem:
 - i. minimum 1 bloku pamięci dyskowych (dla OSS/OST), wyposażonego w pamięci dyskowe oraz elementy do akceleracji I/O (moduły / podsystem pamięci flash)
 - ii. minimum 1 bloku pamięci flash (dla MDS/MDT) - wyposażonego w pamięci flash do składowania meta-danych systemu;
 - b. System składowania danych użytkowników i projektów (project data) - system plików skonfigurowany z wykorzystaniem następujących elementów:
 - i. Minimum 1 bloku pamięci dyskowych, wyposażonego w pamięci dyskowe oraz elementy akceleracji I/O (flash) oraz minimum 1 bloku pamięci flash - wyposażonego w pamięci flash do przechowywania / buforowania danych *LUB* *Minimum 1 blok hybrydowego - zawierającego pamięci dyskowe i flash*

2. Testy wydajnościowe muszą potwierdzać spełnianie przez zaoferowane elementy / bloki systemów składowania wszystkich wymagań SWZ dotyczących wydajności, w tym:
 - a. Dla system składowania dla obliczeń HPC (scratch):
 - i. Wydajności [GB/s] strumieniowej zapisu i odczytu danych
 - ii. Wydajności [IOPS] dla losowego zapisu/odczytu danych w ramach plików
 - iii. Wydajności [IOPS] operacji plikowych
 - b. Dla system składowania dla obliczeń HPC (scratch):
 - i. wydajności [GB/s] strumieniowej zapisu i odczytu danych
 - ii. Wydajności [IOPS] dla losowego zapisu/odczytu danych w ramach plików
 - iii. Wydajności [IOPS] operacji plikowych
6. Testy wydajnościowe muszą zostać przeprowadzone zgodnie z dobrymi praktykami w branży IT.
7. Konfiguracja testowa wykorzystana do przeprowadzenia testów wydajnościowych na potrzeby przygotowania oferty musi być *zbliżona do konfiguracji* zaoferowanej systemów:
 - a. Konfiguracja systemu testowego i zaoferowanego musi być porównywalna pod względem liczby i rodzaju serwerów i kontrolerów dyskowych, liczby i rodzaju napędów dyskowych (HDD) i modułów pamięci flash oraz liczby i rodzajów interfejsów I/O do serwerów (np. FC vs Ethernet) a także liczby i rodzajów serwerów i węzłów systemów pamięci masowych.
 - b. Nie będą akceptowane wyniki testu:
 - i. uzyskane dla systemu w konfiguracji z przeważającą lub wyłączną przestrzenią na pamięciach flash (SSD/NVMe) podane jako wyniki testu wydajnościowego dla przestrzeni na dyskach magnetycznych (HDD);
 - ii. uzyskane dla macierzy w konfiguracji o wielokrotnie większej liczbie modułów pamięci flash (SSD/NVMe) czy napędów dyskowych (HDD) niż zaoferowany system (dopuszcza się wyniki testów dla liczby napędów lub dysków większej o maksymalnie 50% niż w zaoferowanym systemie);
 - iii. uzyskane dla wolumenu danych testowych mniejszego niż dwukrotność sumy pojemności pamięci cache i pamięci operacyjnej RAM w serwerach testujących;
8. Konfiguracja testowa wykorzystana do przeprowadzenia testów wydajnościowych na potrzeby przygotowania oferty musi być taka, że wyniki testów przedstawione w ofercie będą możliwe do uzyskania w zaoferowanej konfiguracji systemu.
9. Zamawiający zastrzega sobie prawo weryfikacji czy zaoferowana konfiguracja systemów w rzeczywistości osiąga zadeklarowane w ofercie poziomy wydajności dla poszczególnych typów testów dla poszczególnych elementów infrastruktury. W tym celu Zamawiający może:
 - a. na etapie oceny ofert - zażądać od wykonawcy zapewnienia dostępu do systemu storage konfiguracji adekwatnej do zaoferowanej bądź do zapewnienia dostępu zdalnego lub fizycznego do systemu znajdującego się w laboratorium Wykonawcy lub producenta sprzętu, bądź to poprzez dostawę do siedziby Zamawiającego takiej liczby bloków systemów pamięci masowych wraz z elementami towarzyszącymi (sieć, infrastruktura otoczenia, zasilanie, szafy), jaka jest wymagana do wykonania testu wydajnościowego - w celu przeprowadzenia testów weryfikujących zadeklarowaną wydajność systemów - zgodnie ze zdefiniowaną procedurą testową;

- b. Po dostawie sprzętu - zażądać przeprowadzenia weryfikacji przez Wykonawcę lub dokonać weryfikacji samodzielnie - poprzez uruchomienie testów wydajnościowych na dostarczonych elementach infrastruktury systemów przechowywania danych - zgodnie ze zdefiniowaną procedurą testową;
10. **Konfiguracja i sposób przeprowadzenia testu** wydajnościowego musi **odzwierciedlać realistyczny scenariusz użycia** systemów pamięci masowej w systemach obliczeniowych:
- a. W ramach testu nie może wykorzystywana - do obsługi żądań I/O - wyłącznie pamięci flash i/lub pamięć RAM w systemach, w których realizowane jest buforowanie w pamięciach flash lub RAM danych składowanych na pamięciach dyskowych i/lub inna metoda akceleracji obsługi żądań I/O z wykorzystaniem pamięci flash lub RAM.
 - b. Wielkość wolumenu testowego musi zapewniać wykorzystanie - do obsługi żądań I/O - pamięci dyskowych a nie tylko pamięci flash i RAM) - dotyczy to zarówno składowania i odczytu bloków danych w plikach/obiektach jak i operacji na plikach i ich metadanych, np. modyfikacji nazw, zmiany atrybutów, odczytu katalogów, odczytu meta-danych itp.
 - c. Rozmiar wolumenu testowego danych musi przekraczać dwukrotność sumy pamięci flash i RAM w elementach / blokach pamięci masowych (elementy OST i MDT dla systemu plików dużej wydajności dla obliczeń-scratch, bloki i węzły pamięci dyskowych i flash systemu przechowywania danych użytkowników i projektów - project data) oraz w serwerach testujących, na których uruchomiony jest narzędzie testujące wydajność.
 - d. Ponadto, bufor systemów masowego przechowywania danych na pamięciach SSD/NVME (flash) a także w pamięci RAM serwerów i węzłów systemu przechowywania danych (dotyczy zarówno systemu scratch jak i project data) oraz buforów serwerów testujących wydajność muszą być każdorazowo, przed wykonaniem tury testów wydajnościowych opróżniane lub wymiatane na pamięci dyskowe. Należy tego dokonać stosownym, właściwym dla danego typu systemu pamięci masowej sposobem - np. odpowiednim poleceniem systemowym na poziomie systemów operacyjnych serwerów systemu pamięci masowej i serwerów testujących, dedykowaną funkcją systemu pamięci masowej lub - jeśli nie jest dla danego systemu udokumentowana metoda opróżniania buforów w pamięciach flash i RAM - poprzez restart elementów systemu pamięci masowej przed każdą turą testu.
11. **Konfiguracja i sposób przeprowadzenia testu** wydajnościowego musi być **adekwatna do zaoferowanej konfiguracji systemów** pamięci masowych:
- a. Testowany wolumen danych musi obejmować **co najmniej 20% całkowitej pojemności** testowanego systemu przechowywania danych (tj. pojemności będącej sumą powierzchni pamięci dyskowych i pamięci SSD/NVMe testowanego systemu przechowywania danych).
 - b. Ponadto konfiguracja testu musi skutkować **równomiernym rozłożeniem żądań I/O** po powierzchni testowanego systemu przechowywania danych. Nie jest dozwolone takie zawężanie obszaru powierzchni systemu przechowywania danych wykorzystywanego do testów wydajnościowych, żeby żądania I/O narzędzia testującego wydajność skierowane były wyłącznie do obszaru pamięci SSD/NVME (flash) i /lub do pamięci cache elementów systemów przechowywania danych i/lub pamięci RAM serwerów/węzłów systemów przechowywania danych i/lub pamięci RAM serwerów testujących.

12. **Powtarzalność i wiarygodność wyników testów wydajnościowych musi zostać potwierdzona** poprzez **wielokrotne wykonanie danego testu** wydajnościowego: minimum 5 tur dla każdego typu testu. Wyniki wszystkich tur testu, wraz z wyliczoną średnią wydajnością uzyskanej we wszystkich turach danego typu testu muszą zostać podane w odpowiednich tabelach formularza technicznego.
13. *Wyniki testów wydajnościowych przeprowadzonych na poszczególnych poziomach muszą być spójne, tzn. różnice wyników testów na różnych poziomach systemu muszą odzwierciedlać typowe narzuty poszczególnych warstw systemów przechowywania. Dotyczy to zależności:*
 - a. *Poziom blokowy vs poziomy plikowy - dla systemu składowania dla obliczeń HPC (scratch) - wyniki uzyskane na poziomie plikowym muszą być niższe niż na poziomie blokowym.*
 - b. *Poziomy obiektowy vs poziomy plikowy - dla systemu składowania danych użytkowników i projektów (project data) - suma wyników dla danego typu testu wydajności uzyskanych na poziomie obiektowym i plikowym musi być nie mniejsza niż 120% wartości wyniku testu przeprowadzonego na poziomie plikowym.*
14. Wyniki wszystkich tur testów wszystkich typów testów muszą zostać udokumentowane w ofertach:
 - a. Wyniki poszczególnych tur wszystkich przeprowadzonych typów testów wydajnościowych muszą zostać umieszczone w odpowiednich tabelach formularza technicznego oferty. Należy wpisać do poszczególnych tabel prezentujących wyniki testów wyniki (wydajność) uzyskane dla poszczególnych tur przeprowadzonych testów wraz z wyliczoną średnią wyników wszystkich tur testu uzyskanych dla danego typu testu.
 - b. Ponadto wartość średnia wydajności dla danego typu testu uzyskanej we wszystkich turach testu musi być podana w formularzu technicznym - wartość ta będzie wykorzystywana w procesie oceny ofert do wyliczania dodatkowej punktacji za kryteria techniczne.
 - c. Ponadto, na żądanie Zamawiającego, Wykonawca zobowiązany jest zaprezentować pełny wydruk wyników testów wydajnościowych, wraz z informacją pochodzącą z monitoringu systemów przechowywania danych zebraną podczas wykonywania testu (monitoring systemu wskazujący obsługiwane liczby operacji I/O czy też uzyskaną przepustowość odczytu czy zapisu danych - uzyskiwane podczas uruchomienia testu wydajności). Pełne wyniki testów muszą zawierać – poza wartościami liczbowymi uzyskanymi dla testów wydajności - także opis środowiska testowego i wykonanych testów - zawierający wszystkie informacje pozwalające na ocenę wiarygodności i poprawności uzyskanych wyników testów, w tym m.in.: konfiguracji elementów infrastruktury przechowywania danych (producent/model/typ, liczba i typ kontrolerów, JBODów, macierzy, pojemność pamięci cache, SSD/NVMe, zastosowane poziomy redundancji danych (RAID, EC itp.), niestandardowe ustawienia konfiguracyjne, liczba serwerów testujących wydajność, pojemność pamięci RAM tych serwerów, nazwa narzędzia testującego i użyte parametry testu, w szczególności wskazujące rozmiar wolumenu testowego i/lub czas wykonania testu, wielkość bloku I/O skonfigurowanego dla poszczególnych testów i liczba wątków testu wydajnościowego itd.);

Procedury testów wydajnościowych

Procedury testów wydajnościowych dla systemów składowania danych dla obliczeń HPC:

Procedura testowa dla bloku danych systemu składowania dla HPC (scratch) - zapis i odczyt sekwencyjny danych na poziomie blokowym

1. Konfiguracja serwera wykorzystywanego w testach bloków danych systemu składowania dla HPC (scratch) musi spełniać wymagania zdefiniowane dla węzła dla danych w zakresie wydajności, niezawodności (zwiększenie ścieżek I/O, konfiguracja H/A);
2. Należy przeprowadzić test przepustowości systemu dyskowego dla operacji **zapisu** sekwencyjnego i dla operacji **odczytu sekwencyjnego** przy wykorzystaniu bloku zapisu o wielkości 1MB.
3. Testy (**zapisu i odczytu**) **sekwencyjnego** należy przeprowadzić wg następującej procedury:
 - a. Do przeprowadzenia testu należy wykorzystać jedno z narzędzi:
 - i. **fio** (<http://freecode.com/projects/fio>):
 1. należy wybrać workload write oraz read (parametr --rw);
 - a. przykładowe polecenie uruchamiające test na 1 z hostów:
 - i. dla testu **zapisu** sekwencyjnego blokiem **1MB**:

```
fio --name=BD_write_1M --filename=/dev/vol --rw=write --ioengine=libaio --iodepth=16 --bs=1M --direct=1 --size=2000g --numjobs=32 --group_reporting --output=output_write_1M.log
```
 - ii. dla testu **odczytu** sekwencyjnego blokiem **1MB**:

```
fio --name=BD_read_1M --filename=/dev/vol --rw=read --ioengine=libaio --iodepth=16 --bs=1M --direct=1 --size=2000g --numjobs=32 --group_reporting --output=output_read_1M.log
```
 - ii. **iometer** (<http://www.iometer.org>),
 1. w wypadku tego narzędzia należy skonfigurować test losowego zapisu i losowego odczytu oraz określić zawartość danych testowych jako losowy;
 - iii. UWAGA! nie jest dopuszczalne wykorzystanie narzędzia iozone, z uwagi na brak możliwości generowania odpowiednio losowej treści danych testujących przez to narzędzie;
 - iv. W celu wykonania testu wydajności należy wykorzystać oficjalną, publiczną wersję oprogramowania testującego - np. najnowsza wersja benchmarku dostępnego dla danego systemu operacyjnego w gałęzi stabilnej lub skompilować własną wersję ze źródeł dostępnych w publicznym repozytorium. Niedopuszczalne są jakiegokolwiek modyfikacje w kodzie źródłowym lub binarnym benchmarków.
 - b. Serwery testujące muszą pracować pod kontrolą systemu z rodziny Linux (kernel min. 5.0). Host, który wyłącznie nadzoruje wykonanie testu może działać pod systemem Windows.
 - c. Wyniki testu należy uzyskać w następujący sposób:
 - i. Należy wykonać 5 tur testu. Każda tura testu powinna trwać minimum 1000 sekund lub zapisywać lub odczytywać zbiór danych o wolumenie minimum 4-krotnie większym niż suma pamięci RAM i cache w serwerach testujących i kontrolerach i elementach systemu przechowywani danych.

- ii. W każdej turze należy uruchomić program testujący na 1 lub więcej hostów przy czym należy zapewnić synchronizację (równoczesność) wykonania testu na wielu hostach – rozsuniecie w czasie rozpoczęcia i zakończenia testu na poszczególnych hostach nie może być większe niż 5 sekund.
 1. W przypadku wykorzystania narzędzia fio należy wykonać synchronizację ręcznie lub za pomocą skryptu uruchamiającego test równocześnie na wielu hostach i zatrzymującego go po określonym czasie na wielu hostach z niewielkim przesunięciem w czasie, nie przekraczającym 1% czasu wykonania testu.
 2. W przypadku narzędzia iometer można wykorzystać mechanizmy uruchomienia klastrowego testu wbudowane w to narzędzie.
 - iii. wynik danego testu uzyskuje się w sposób właściwy dla narzędzia – tj. poprzez odczyt wartości IOPS dla całego testu bądź dla poszczególnych wątków i zsumowanie wartości dla wszystkich wątków.
 - iv. Ostateczne wyniki testu uzyskuje się poprzez uśrednienie wyników z 5 tur testu (średnia arytmetyczna) dla poszczególnych operacji (zapis i odczyt).
4. Wyniki testu wydajności muszą zostać zamieszczone w formularzu technicznym stanowiącym część oferty. Dla każdego typu testu (zapis i odczyt) należy zaprezentować wyniki każdej z 5 tur testu oraz wartość uśrednioną z 5 tur testu (oddzielna średnia wartość dla zapisów i dla odczytów). Dla prezentacji wyników można wykorzystać poniższą tabelę

Rodzaj testu systemu scratch - poziom blokowy	Zapis sekwencyjny blokiem 1MB [MB/s]	Odczyt sekwencyjny blokiem 1MB [MB/s]
Wynik tura 1		
Wynik tura 2		
Wynik tura 3		
Wynik tura 4		
Wynik tura 5		
Średni wynik dla testu		

Procedura testowa dla bloku meta-danych systemu składowania dla HPC (scratch) - zapis i odczyt losowy danych na poziomie blokowym

1. Konfiguracja serwera wykorzystywanego w testach bloków meta-danych systemu składowania dla HPC (scratch) musi spełniać wymagania zdefiniowane dla węzła dla meta-danych w zakresie wydajności, niezawodności (zwielokrotnienie ścieżek I/O, konfiguracja H/A);
2. Należy przeprowadzić test przepustowości podsystemu pamięci flash (NVMe, SSD) dla operacji **zapisu losowego** i dla operacji **odczytu losowego** przy wykorzystaniu bloku zapisu o wielkości 4kB.

3. Testy (**zapisu i odczytu**) należy przeprowadzić wg następującej procedury:

a. Do przeprowadzeniu testu należy wykorzystać jedno z narzędzi:

i. **fio** (<http://freecode.com/projects/fio>):

1. należy wybrać workload **randwrite** oraz **randread** (parametr **--rw**);
2. przykładowe polecenie uruchamiające testu blokiem I/O losowego na 1 z hostów:

a. dla testu **zapisu**: blokiem **4k**

```
fio --name=test_ randwrite --ioengine=libaio --iodepth=16 --rw=randwrite  
--bs=4k --direct=1 --size=20100g --numjobs=16 32  
--filename=/dev/vol --group_reporting --output=output_ randwrite_4k.log
```

b. dla testu **odczytu** blokiem : blokiem **4k**

```
fio --name=test_ randread --ioengine=libaio --iodepth=16 --rw=randread  
--bs=4k --direct=1 --size=2000g --numjobs=32  
--filename=/dev/vol --group_reporting --output=output_ randread_4k.log  
--filename=/dev/vol --group_reporting --output=output_ randread_1M.log
```

ii. **iometer** (<http://www.iometer.org>),

1. w wypadku tego narzędzia należy skonfigurować test losowego zapisu i losowego odczytu oraz określić zawartość danych testowych jako losowy;

iii. **UWAGA!** nie jest dopuszczalne wykorzystanie narzędzia **iozone**, z uwagi na brak możliwości generowania odpowiednio losowej treści danych testujących przez to narzędzie;

iv. W celu wykonania testu wydajności należy wykorzystać oficjalną, publiczną wersję oprogramowania testującego - np. najnowsza wersje benchmarku dostępnego dla danego systemu operacyjnego w gałęzi stabilnej lub skompilować własną wersję ze źródeł dostępnych w publicznym repozytorium. Niedopuszczalne są jakiegokolwiek modyfikacje w kodzie źródłowym lub binarnym benchmarków.

b. Serwery testujące muszą pracować pod kontrolą systemu z rodziny Linux (kernel min. 5.0). Host, który wyłącznie nadzoruje wykonanie testu może działać pod systemem Windows. .

c. Podczas testu mogą być włączone mechanizmy de-duplikacji danych działające po stronie systemu pamięci SSD/NVMe, jednakże należy zapewnić, że narzędzie testujące generuje odpowiednio losowe dane testowe.

d. Wyniki testu należy uzyskać w następujący sposób:

- i. Należy wykonać 5 tur testu. Każda tura testu powinna trwać minimum 1000 sekund lub zapisywać lub odczytywać zbiór danych o wolumenie minimum 4-krotnie większym niż suma pamięci RAM i cache w serwerach testujących i kontrolerach oraz macierzy SSD/NVMe.
- ii. W każdej turze należy uruchomić program testujący na 1 lub więcej hostów przy czym należy zapewnić synchronizację (równoczesność) wykonania testu na wielu hostach – rozsuniecie w

czasie rozpoczęcia i zakończenia testu na poszczególnych hostach nie może być większe niż 5 sekund.

1. W przypadku wykorzystania narzędzia fio należy wykonać synchronizację ręcznie lub za pomocą skryptu uruchamiającego test równocześnie na wielu hostach i zatrzymującego go po określonym czasie na wielu hostach z niewielkim przesunięciem w czasie, nie przekraczającym 1% czasu wykonania testu.
 2. W przypadku narzędzia iometer można wykorzystać mechanizmy uruchomienia klastrowego testu wbudowane w to narzędzie.
 - iii. wynik danego testu uzyskuje się w sposób właściwy dla narzędzia – tj. poprzez odczyt wartości IOPS dla całego testu bądź dla poszczególnych wątków i zsumowanie wartości dla wszystkich wątków.
 - iv. Ostateczne wyniki testu uzyskuje się poprzez uśrednienie wyników z 5 tur testu (średnia arytmetyczna) dla poszczególnych operacji (zapis i odczyt).
4. Wyniki testu wydajności muszą zostać zamieszczone w załączniku do oferty. Dla każdego typu testu (zapis i odczyt) należy zaprezentować wyniki każdej z 5 tur testu oraz wartość uśrednioną z 5 tur testu (oddzielna średnia wartość dla zapisów i dla odczytów). Dla prezentacji wyników można wykorzystać poniższą tabelę:

Rodzaj testu systemu scratch - poziom blokowy	Zapis blokiem 4kB [IOPS]	Odczyt blokiem 4kB [IOPS]
Wynik tura 1		
Wynik tura 2		
Wynik tura 3		
Wynik tura 4		
Wynik tura 5		
Średni wynik dla testu		

Procedura testowa dla bloku danych systemu składowania dla HPC (scratch) - na poziomie plikowym

1. Należy przeprowadzić test przepustowości na poziomie plikowym dla sekwencyjnego zapisu i odczytu danych dużym blokiem (1M), testy losowego zapisu i odczytu danych małym blokiem (4k) oraz operacji plikowych i na meta-danych:
2. Test przepustowości na poziomie plikowym dla sekwencyjnego zapisu i odczytu danych dużym blokiem (1M) - należy przeprowadzić wg następującej procedury:
 - a) **Testy przepustowości** - do pomiaru przepustowości GB/s „bloku pamięci” tj. OSS + OST. Liczba klientów testowych powinna być wystarczająco duża, aby prawidłowo wykorzystać przepustowość sekwencyjnego I/O systemów pamięci masowej i serwerów OSS
 - Sekwencyjny zapis blokowy 1 MB:
fio --name=BD_write_1M --filename=/dev/vol --rw=write --ioengine=libaio --iodepth=16 --bs=1M --direct=1 --size=2000g --numjobs=32 --group_reporting --output=output_write_1M.log
 - Sekwencyjny odczyt bloków 1 MB:
fio --name=BD_read_1M --filename=/dev/vol --rw=read --ioengine=libaio --iodepth=16 --bs=1M --direct=1 --size=2000g --numjobs=32 --group_reporting --output=output_read_1M.log
3. Test losowego zapisu i odczytu danych małym blokiem (4k) - należy przeprowadzić wg następującej procedury:
 - a) **Testy zapisu losowego** - do pomiaru zdolności IOPS (ang. IOPS capability) „bloku pamięci” tj. OSS + OST. Założeniem jest, że losowe I/O na danych jest przyspieszane przez pamięci flash/SSD SSD w ramach serwerów OST.
 - Losowy zapis blokowy 4kB:
fio --name=test_randwrite --ioengine=libaio --iodepth=16 --rw=randwrite --bs=4k --direct=1 --size=2000g --numjobs=32 --filename=/dev/vol --group_reporting --output=output_randwrite_4k.log
 - Losowy odczyt blokowy 4kB:
fio --name=test_randread --ioengine=libaio --iodepth=16 --rw=randread --bs=4k --direct=1 --size=2000g --numjobs=32 --filename=/dev/vol --group_reporting --output=output_randread_4k.log
3. Testy operacji plikowych i na meta-danych - należy przeprowadzić wg następującej procedury:
 - a) **Testy operacji plikowych** - należy przeprowadzić za pomocą aplikacji **MDtest** (<https://github.com/LLNL/mdtest>).
MDtest to aplikacja oparta na MPI do oceny wydajności metadanych systemu plików. Została zaprojektowana do testowania równoległych systemów plików. MDTest nie jest testem porównawczym specyficznym dla Lustre i można go uruchomić w dowolnym systemie plików zgodnym z POSIX. Wymaga on w pełni zainstalowanej i skonfigurowanej implementacji systemu plików. W przypadku Lustre oznacza to, że usługi MGS, MDS i OSS muszą być zainstalowane, skonfigurowane i uruchomione oraz że istnieje populacja klientów Lustre działających z zamontowanym systemem plików Lustre.
Aplikacja mdtest działa na klientach Lustre we w pełni skonfigurowanym systemie plików Lustre. Wiele procesów mdtest jest uruchamianych równoległe w kilku węzłach przy użyciu MPI w celu wysycenia / wykorzystania przepustowości I/O systemu plików. Program może tworzyć drzewa katalogów o

dowolnej głębokości i może być skierowany do tworzenia kombinacji obciążeń, w tym testów tylko na plikach.

W celu wykonania testu wydajności należy wykorzystać oficjalną, publiczną wersję oprogramowania testującego - np. najnowsza wersje benchmarku dostępnego dla danego systemu operacyjnego w gałęzi stabilnej lub skompilować własną wersję ze źródeł dostępnych w publicznym repozytorium.

Niedopuszczalne są jakiegokolwiek modyfikacje w kodzie źródłowym lub binarnym benchmarków.

Szczegółowa procedura uruchamiania tetow wydajności na poziomie plikowym z wykorzystaniem aplikacji MDTEST:

a) Przygotowanie środowiska

- instalacja pakietów

```
sudo yum -y install openmpi-devel git
```

- kompilacja pakietów

```
git clone https://github.com/IOR-LANL/ior.git
```

```
cd ior
```

```
module load mpi/openmpi-x86_64
```

```
make clean && make
```

- Środowisko uruchomieniowe

```
-- Utworzenie kont shellowych na maszynach testowych
```

```
-- Wymiana kluczy ssh między maszynami
```

```
ssh-keygen -t rsa -N "" -f $HOME/.ssh/id_rsa
```

```
-- uruchomienie środowiska openmpi na klientach Lustre
```

```
yum install openmpi
```

```
module load mpi/openmpi-x86_64
```

b) Uruchomienie testów

- tworzymy plik z hostami testowymi z poziomu jednej maszyny testującej:

```
for i in `seq -f "%03g" 1 32`; do
```

```
  echo "n"$i" slots=16"
```

```
done > $HOME/hfile
```

```
# Result:
```

```
n001 slots=16
```

```
n002 slots=16
```

```
n003 slots=16
```

```
n004 slots=16
```

- uruchomienie zrównoleżonego testu mdtest z w/w maszyn

```
mpirun --hostfile $HOME/hfile -np 48 ./mdtest -n 1000 -i 2 -F -C -T -r -u -e=4096 -w 4096 -E -d <ścieżka>
```

Uwaga!

W odróżnieniu od procedury testowej opisanej dla systemu scratch wykorzystującej narzędzie mdtest, benchmark narzędziem mdtest dla project data powinien testować tworzenie niepustych plików o wielkości 4kB każdy.

c) przykładowy output testu:

FS: 6453.6 TiB Used FS: 3.1% Inodes: 4975.7 Mi Used Inodes: 1.6%
24 tasks, 24000 files/directories

SUMMARY rate: (of 1 iterations)

Operation	Max	Min	Mean	Std Dev
Directory creation	4418.876	4418.876	4418.876	0.000
Directory stat	8669.878	8669.878	8669.878	0.000
Directory rename	495.019	495.019	495.019	0.000
Directory removal	3829.898	3829.898	3829.898	0.000
File creation	5737.289	5737.289	5737.289	0.000
File stat	18487.891	18487.891	18487.891	0.000
File read	26764.266	26764.266	26764.266	0.000
File removal	5468.135	5468.135	5468.135	0.000
Tree creation	1494.763	1494.763	1494.763	0.000
Tree removal	732.118	732.118	732.118	0.000

1. Wyniki poszczególnych typów testów wydajności muszą zostać zamieszczone w załączniku do oferty.
- a. Dla każdego typu testu operującego na danych (zapis i odczyt) strumieniowy/sekwencyjny vs losowy w ramach plików należy zaprezentować wyniki każdej z 5 tur testu oraz wartość uśrednioną z 5 tur testu (oddzielna średnia wartość dla zapisów i dla odczytów, oddzielnie należy uśrednić wyniki testów operacji sekwencyjnych/strumieniowych i losowych). Dla prezentacji wyników można wykorzystać poniższe tabele:

Rodzaj testu systemu scratch - poziom plikowy	Zapis blokiem 1MB [GB/s]	Odczyt blokiem 1MB [GB/s]
Wynik tura 1		
Wynik tura 2		
Wynik tura 3		
Wynik tura 4		
Wynik tura 5		
Średni wynik dla testu		

Rodzaj testu systemu scratch - poziom plikowy	Zapis blokiem 4kB [IOPS]	Odczyt blokiem 4kB [IOPS]
Wynik tura 1		
Wynik tura 2		
Wynik tura 3		
Wynik tura 4		
Wynik tura 5		
Średni wynik dla testu		

- b. Dla każdego typu testu operacji plikowych i na meta-danych - należy zaprezentować wyniki każdej z 5 tur testu oraz wartość uśrednioną z 5 tur testu (oddzielna średnia wartość dla każdego typu operacji, przy czym jako wynik końcowy testu należy wziąć wartość określoną jako „mean” na wyjściu narzędzia testowego). Dla prezentacji wyników można wykorzystać poniższe tabele:

Rodzaj testu systemu scratch - poziom plikowy	<i>File create</i> [OPS/s]	<i>File stat</i> [OPS/s]	<i>File read</i> [OPS/s]	<i>File removal</i> [OPS/s]
Wynik tura 1				
Wynik tura 2				
Wynik tura 3				
Wynik tura 4				
Wynik tura 5				
Średni wynik dla testu				

Procedura testowa dla bloku danych systemu składowania danych użytkowników i projektów (project data) - na poziomie plikowym

- Należy przeprowadzić test przepustowości na poziomie plikowym dla sekwencyjnego zapisu i odczytu danych dużym blokiem (1M), testy losowego zapisu i odczytu danych małym blokiem (4k) oraz operacji plikowych i meta-danych:
- Test przepustowości na poziomie plikowym dla sekwencyjnego zapisu i odczytu danych dużym blokiem (1M) - należy przeprowadzić wg następującej procedury:
 - Testy przepustowości** - do pomiaru przepustowości GB/s podsystemu dyskowego (z ew. akceleracją flash).
Liczba klientów testowych powinna być wystarczająco duża, aby prawidłowo wykorzystać przepustowość sekwencyjnego I/O podsystemu dyskowego (z ew. akceleracją flash).
- Sekwencyjny zapis blokowy 1 MB:

```
fio --name=BD_write_1M --filename=/dev/vol --rw=write --ioengine=libaio --iodepth=16 --bs=1M --direct=1 --size=2000g --numjobs=32 --group_reporting --output=output_write_1M.log
```

 - Sekwencyjny odczyt bloków 1 MB:

```
fio --name=BD_read_1M --filename=/dev/vol --rw=read --ioengine=libaio --iodepth=16 --bs=1M --direct=1 --size=2000g --numjobs=32 --group_reporting --output=output_read_1M.log
```
- Test losowego zapisu i odczytu danych małym blokiem (4k) - należy przeprowadzić wg następującej procedury:
 - Testy zapisu losowego** - do pomiaru zdolności IOPS (ang. IOPS capability) podsystemu dyskowego (z ew. akceleracją flash). Założeniem jest, że losowe I/O na danych jest przyspieszane przez pamięci flash/SSD.
- Losowy zapis blokowy 4kB:

```
fio --name=test_randwrite --ioengine=libaio --iodepth=16 --rw=randwrite --bs=4k --direct=1 --size=2000g --numjobs=32 --filename=/dev/vol --group_reporting --output=output_randwrite_4k.log
```


- Losowy odczyt blokowy 4kB:

```
fio --name=test_randread --ioengine=libaio --iodepth=16 --rw=randread --bs=4k --direct=1 --size=2000g -  
-numjobs=32 --filename=/dev/vol --group_reporting --output=output_randread_4k.log
```

3. Testy operacji plikowych i na meta-danych - należy przeprowadzić wg następującej procedury:

a) **Testy operacji plikowych** - należy przeprowadzić za pomocą aplikacji **MDtest**

(<https://github.com/LLNL/mdtest>).

MDtest to aplikacja oparta na MPI do oceny wydajności metadanych systemu plików. Została zaprojektowana do testowania równoległych systemów plików. MDTest nie jest testem porównawczym specyficznym dla Lustre i można go uruchomić w dowolnym systemie plików zgodnym z POSIX. Wymaga on w pełni zainstalowanej i skonfigurowanej implementacji systemu plików. W przypadku Lustre oznacza to, że usługi MGS, MDS i OSS muszą być zainstalowane, skonfigurowane i uruchomione oraz że istnieje populacja klientów Lustre działających z zamontowanym systemem plików Lustre.

Aplikacja mdtest działa na klientach Lustre we w pełni skonfigurowanym systemie plików Lustre. Wiele procesów mdtest jest uruchamianych równolegle w kilku węzłach przy użyciu MPI w celu wysycenia / wykorzystania przepływności I/O systemu plików. Program może tworzyć drzewa katalogów o dowolnej głębokości i może być skierowany do tworzenia kombinacji obciążeń, w tym testów tylko na plikach.

Szczegółowa procedura uruchamiania testów wydajności na poziomie plikowym z wykorzystaniem aplikacji MDTEST:

a) Przygotowanie środowiska

- instalacja pakietów

```
sudo yum -y install openmpi-devel git
```

- kompilacja pakietów

```
git clone https://github.com/IOE-LANL/ior.git
```

```
cd ior
```

```
module load mpi/openmpi-x86_64
```

```
make clean && make
```

- Środowisko uruchomieniowe

```
-- Utworzenie kont shellowych na maszynach testowych
```

```
-- Wymiana kluczy ssh między maszynami
```

```
ssh-keygen -t rsa -N "" -f $HOME/.ssh/id_rsa
```

```
-- uruchomienie środowiska openmpi na klientach testujących
```

```
yum install openmpi
```

```
module load mpi/openmpi-x86_64
```

b) Uruchomienie testów

- tworzymy plik z hostami testowymi z poziomu jednej maszyny testującej:

```
for i in `seq -f "%03g" 1 32`; do
```

```
  echo "n$i" slots=16"
```

```
done > $HOME/hfile
```

Result:

n001 slots=16

n002 slots=16

n003 slots=16

n004 slots=16

- uruchomienie zrównoleżonego testu mdtest z w/w maszyn

```
mpirun --hostfile $HOME/hfile -np 48 ./mdtest -n 20840 -i 10 -u -d /lustre/demo/mdtest-scratch
```

c) przykładowy output testu:

FS: 6453.6 TiB Used FS: 3.1% Inodes: 4975.7 Mi Used Inodes: 1.6%

24 tasks, 24000 files/directories

SUMMARY rate: (of 1 iterations)

Operation	Max	Min	Mean	Std De
Directory creation	4418.876	4418.876	4418.876	0.000
Directory stat	8669.878	8669.878	8669.878	0.000
Directory rename	495.019	495.019	495.019	0.000
Directory removal	3829.898	3829.898	3829.898	0.000
File creation	5737.289	5737.289	5737.289	0.000
File stat	18487.891	18487.891	18487.891	0.000
File read	26764.266	26764.266	26764.266	0.000
File removal	5468.135	5468.135	5468.135	0.000
Tree creation	1494.763	1494.763	1494.763	0.000
Tree removal	732.118	732.118	732.118	0.000

1. Wyniki poszczególnych typów testów wydajności muszą zostać zamieszczone w załączniku do oferty.
- a. Dla każdego typu testu operującego na danych (zapis i odczyt) strumieniowy/sekwencyjny vs losowy w ramach plików należy zaprezentować wyniki każdej z 5 tur testu oraz wartość uśrednioną z 5 tur testu (oddzielna średnia wartość dla zapisów i dla odczytów, oddzielnie należy uśrednić wyniki testów operacji sekwencyjnych/strumieniowych i losowych). Dla prezentacji wyników można wykorzystać poniższe tabele:

Rodzaj testu systemu project data - poziom plikowy	Zapis blokiem 1MB [GB/s]	Odczyt blokiem 1MB [GB/s]
Wynik tura 1		
Wynik tura 2		
Wynik tura 3		
Wynik tura 4		
Wynik tura 5		
Średni wynik dla testu		

Rodzaj testu systemu project data - poziom plikowy	Zapis blokiem 4kB [IOPS]	Odczyt blokiem 4kB [IOPS]
Wynik tura 1		
Wynik tura 2		
Wynik tura 3		
Wynik tura 4		
Wynik tura 5		
Średni wynik dla testu		

- b. Dla każdego typu testu operacji plikowych i na meta-danych - należy zaprezentować wyniki każdej z 5 tur testu oraz wartość uśrednioną z 5 tur testu (oddzielna średnia wartość dla każdego typu operacji, przy czym jako wynik końcowy testu należy wziąć wartość określoną jako „mean” na wyjściu narzędzia testowego). Dla prezentacji wyników można wykorzystać poniższe tabele:

Rodzaj testu systemu project data - poziom plikowy	<i>Directory create [OPS/s]</i>	<i>Directory stat [OPS/s]</i>	<i>Directory rename [OPS/s]</i>	<i>Directory removal [OPS/s]</i>
Wynik tura 1				
Wynik tura 2				
Wynik tura 3				
Wynik tura 4				
Wynik tura 5				
Średni wynik dla testu				

Rodzaj testu systemu project data - poziom plikowy	<i>File create [OPS/s]</i>	<i>File stat [OPS/s]</i>	<i>File read [OPS/s]</i>	<i>File removal [OPS/s]</i>
Wynik tura 1				
Wynik tura 2				
Wynik tura 3				
Wynik tura 4				
Wynik tura 5				
Średni wynik dla testu				