



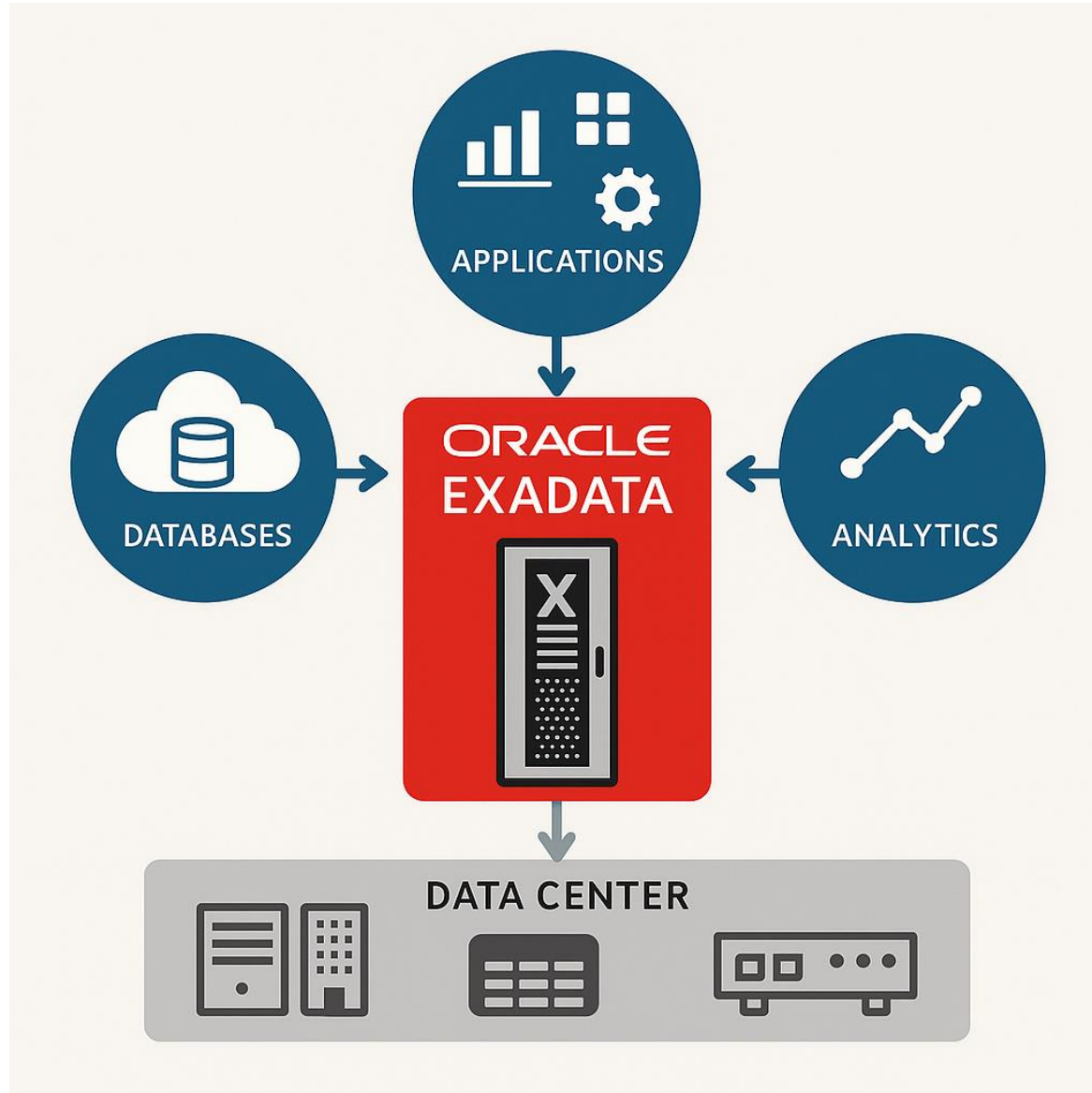
ORACLE EXADATA

IDEO-LAB

March 2025 – version 2.9

By Guillaume Oneill - Lockheed LLC - Raytheon





Oracle et ExaData un Peu d'Histoire

◆ Oracle Corporation : un leader mondial des bases de données

Fondée en 1977 sous le nom de *Software Development Laboratories (SDL)* par Larry Ellison, Bob Miner et Ed Oates, la société a été rebaptisée *Relational Software Inc.* puis finalement **Oracle Corporation** en 1982. Oracle s'est imposée comme l'un des leaders mondiaux en matière de **bases de données relationnelles** et de solutions logicielles d'entreprise.

Aujourd'hui, Oracle est également un acteur clé du **Cloud Computing** (Oracle Cloud Infrastructure – OCI) et des solutions SaaS (ERP, CRM, HCM).

◆ Naissance d'Oracle Exadata

- **Lancement en 2008** : Oracle présente **Exadata Database Machine**, conçu initialement en partenariat avec HP.
- Objectif : fournir une **appliance spécialisée** intégrant matériel et logiciel, optimisée pour les **bases de données hautes performances** (OLTP et Data Warehousing).
- Depuis 2009, Oracle développe Exadata **en interne** après le rachat de **Sun Microsystems**, ce qui lui permet de maîtriser toute la chaîne : serveurs, stockage, OS (Oracle Linux), réseau (Infiniband / RoCE) et moteur base de données.

Caractéristiques clés

- **Smart Scan** : exécution de traitements directement dans le stockage.
- **Storage Indexes** : filtrage intelligent réduisant l'I/O.
- **Hybrid Columnar Compression (HCC)** : compression massive des données.
- **Haute disponibilité et scalabilité extrême.**

◆ Les principaux clients mondiaux d'Oracle Exadata

Exadata est utilisé dans des environnements critiques où la performance et la fiabilité sont stratégiques :

Amérique du Nord

- AT&T (télécoms)
- JP Morgan Chase, Bank of America, Citigroup (banques)
- Lockheed Martin, Raytheon (défense et aéronautique)
- Facebook / Meta pour certaines plateformes analytiques

Europe

- Deutsche Bank (Allemagne)
- Lloyds Banking Group (Royaume-Uni)
- Airbus (France / Europe)
- EDF, BNP Paribas, Crédit Agricole, BPCE (France – secteur bancaire et énergie)

Asie-Pacifique

- Emirates NBD (Dubai)
- China Mobile, China Telecom (Chine – télécoms massifs)
- Telstra (Australie)
- Nippon Telegraph and Telephone (NTT) (Japon)

◆ Positionnement

Oracle Exadata est reconnu comme :

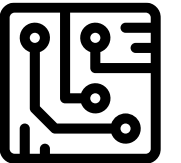
- une infrastructure haut de gamme pour bases de données transactionnelles et analytiques,
- un standard de facto dans de nombreux secteurs : banques, assurances, télécoms, défense, énergie, grande distribution.

INFRASTRUCTURE HARDWARE & SOFTWARE

- Smart Scan
- Storage Index
- HCC
- IORM
- Flash Cache
- Réseau (IB/RoCE)
- Outils d'admin (CellCLI, DBMCLI, DCLI, ExaCLI)
- Storage Server (cellsrv)
- Exachk & Patching



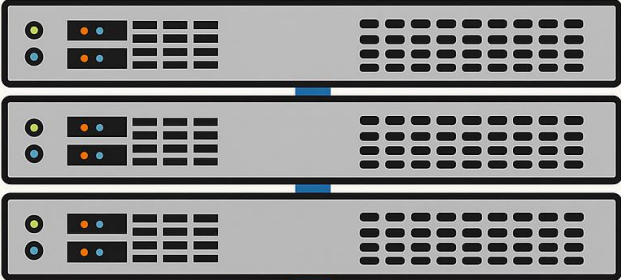
- Rack Préconfiguré X9M X10M
- DataBase Servers x86
- Storage Servers HDD SSD NVME
- Réseau ExaData
- Options Matérielles



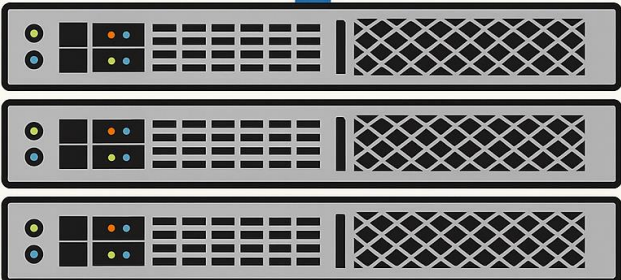
INFRASTRUCTURE HARDWARE

DATABASE SERVERS

(Compute Nodes)



**EXADATA
NETWORK**



STORAGE SERVERS

(Exadata Storage Cells)

1 Vue d'ensemble

Oracle Exadata = combinaison serveurs de calcul (compute nodes) + serveurs de stockage (storage cells) + réseau haut débit (InfiniBand / RoCE).

Tout est livré en rack préconfiguré (X9M, X10M).

👉 Trois composantes matérielles principales :

1. Database Servers (Compute Nodes)
 2. Storage Servers (Cells)
 3. Réseau Exadata (InfiniBand → RoCE 100 GbE)
-

2 Database Servers (Compute Nodes)

- Hébergent : Oracle RAC, ASM, bases Oracle.
- Serveurs x86 haut de gamme :
 - CPU : Intel Xeon ou AMD EPYC (jusqu'à 64–96 cœurs par serveur).
 - RAM : de centaines de Go à plusieurs To par nœud.
 - OS : Oracle Linux optimisé pour Exadata.
- Rôle : exécution du moteur Oracle Database.

3 Storage Servers (Exadata Storage Cells)

- Serveurs x86 spécialisés, chacun équipé de :
 - HDD haute capacité (data warehouse / archivage).
 - SSD/NVMe Flash (accélération Smart Flash Cache).
 - Carte réseau RoCE / InfiniBand.
- Logiciel embarqué : Exadata Storage Server Software (cellsrv).
- Rôle : stockage + intelligence (Smart Scan, Storage Index, IORM, HCC).

👉 Deux types de modèles :

- High Capacity (HC) → HDD + Flash cache (capacité max).
 - Extreme Flash (EF) → 100% Flash/NVMe (performance max).
-

4 Réseau Exadata

- Interconnexion RoCE (100 GbE RDMA) (anciennement InfiniBand).
- Latence ultra basse + débit très élevé.
- Sert à :
 - Communication DB server ↔ Storage server.
 - Cache Fusion entre nœuds RAC.
 - Gestion haute disponibilité.

5 Options matérielles

- Configurations standards : 1/8 rack, 1/4 rack, 1/2 rack, full rack.
 - Évolutivité horizontale : possibilité d'ajouter des DB servers et/ou des storage cells.
 - Sécurité matérielle : chiffrement au niveau disque.
 - Redondance : alimentation, réseau, disques (RAID + ASM).
-

6 Exemple – Exadata X10M (2024)

- Jusqu'à 896 CPU cores par rack (compute + storage).
 - Jusqu'à 22 To de RAM dans les compute nodes.
 - Jusqu'à 3,6 Po de stockage brut par rack.
 - NVMe Flash : plusieurs centaines de To.
 - Réseau interne : 100 GbE RDMA.
-

7 Points clés à retenir

- Oracle Exadata = matériel + logiciel intégré.
- Compute nodes = moteur DB (RAC, ASM).
- Storage cells = stockage intelligent (Smart Scan, Flash, HCC, IORM).
- Réseau RoCE = colonne vertébrale haute performance.
- Évolutivité : scale-out par ajout de nœuds.

EXADATA ET SES COMPETITEURS

- 1. Concurrents directs (appliances bases de données / Data Warehouse)
- 2. Concurrents Cloud (Database-as-a-Service hautes performances)
- 3. Concurrents orientés stockage / infrastructure optimisée

◆ 1. Concurrents directs (appliances bases de données / Data Warehouse)

Ces solutions intègrent matériel + logiciel optimisés, un peu comme Exadata :

- **IBM Netezza (PureData System for Analytics)** → spécialisé dans l'analytique massivement parallèle.
- **Teradata Vantage** → référence historique pour les gros entrepôts de données (banques, telcos, retail).
- **HPE Vertica (désormais sous Micro Focus / OpenText)** → orienté big data analytics, massivement scalable.
- **SAP HANA Appliance** (en mode appliance ou cloud) → surtout puissant pour l'in-memory computing.

◆ 2. Concurrents Cloud (Database-as-a-Service hautes performances)

De plus en plus, la concurrence se fait sur le cloud :

- **Amazon Redshift (AWS)** → orienté data warehouse cloud.
- **Amazon Aurora (compatible MySQL/Postgres)** → alternative transactionnelle mais pas aussi intégrée.
- **Google BigQuery** → cloud-native, serverless, très fort en analytique.
- **Snowflake** → concurrent majeur en mode SaaS, très populaire dans les grandes entreprises.
- **Microsoft Azure Synapse Analytics (ex-SQL Data Warehouse)** → concurrent direct pour l'analytique cloud.

◆ 3. Concurrents orientés stockage / infrastructure optimisée

Exadata est aussi un système matériel optimisé Oracle Database. Sur ce terrain, on retrouve :

- Dell EMC PowerMax + PowerFlex (infrastructure optimisée pour DBs lourdes).
- HPE Alletra / 3PAR (stockage haute perf pour bases de données).
- Hitachi Vantara (VSP 5000, etc.) → stockage ultra-critique pour bases transactionnelles.
- Fujitsu Primeflex / Primequest → serveurs dédiés haute perf pour Oracle et SAP.

✓ Résumé :

- Concurrents historiques : Teradata, IBM Netezza, Vertica, SAP HANA.
- Concurrents Cloud natifs : Snowflake, BigQuery, Redshift, Synapse.
- Concurrents infrastructure : Dell EMC, HPE, Hitachi.

👉 Exadata reste unique car il est co-développé matériel + software par Oracle, avec un focus total sur Oracle Database (ce que les autres n'ont pas).

Concurrents d'Oracle Exadata - Vue comparative

Fujitsu Primeflex / Primequest - Serveurs haute perf pour Oracle & SAP

Hitachi Vantara VSP - Stockage critique pour bases transactionnelles

HPE Alletra / 3PAR - Stockage haute performance DB

Dell EMC PowerMax/Flex - Infra optimisée bases lourdes

Azure Synapse Analytics - Analytique cloud Microsoft

Snowflake - SaaS Data Warehouse, populaire grandes entreprises

Google BigQuery - Cloud-native, serverless, analytique

Amazon Aurora - Transactionnel MySQL/Postgres mais limité

Amazon Redshift - Data warehouse cloud

SAP HANA Appliance - In-memory computing

HPE Vertica - Big Data Analytics scalable

Teradata Vantage - Entrepôt de données pour banques, telcos, retail

IBM Netezza (PureData) - Analytique massivement parallèle

- Appliances (historic)
- Cloud (DBaaS)
- Infrastructure optin



LES CLIENTS EXADATA EN FRANCE

1. **BNP Paribas** — annonce officielle (09 janv. 2025) d'adoption d'**Oracle Exadata Cloud@Customer**, hébergé dans les datacenters BNPP (France & Belgique). [BNP Paribas](#) [CIO Online](#)
2. **Crédit Agricole (CACIB)** — Crédit Agricole compte parmi les premiers clients européens d'**Autonomous Database sur Exadata Cloud@Customer** (annonce Oracle, 08 juil. 2020), avec des gains de performance cités par la banque. [oracle.com](#) [blogs.oracle.com](#)
3. **Société Générale (groupe)** — usage Exadata documenté au niveau du groupe (ex. BRD, filiale SG, en Exadata Cloud@Customer) et **offres d'emploi SG** requérant l'expertise Exadata (indice fort d'usage interne côté France). [oracle.com](#) [Careers Société Générale](#)
4. **Groupe BPCE (Banques Populaires, Caisses d'Épargne, Natixis)** — présence Exadata **rapportée par un traqueur technologique tiers** (TheirStack). NB : ce n'est pas une confirmation officielle Oracle/BPCE, mais un indicateur crédible. [TheirStack.com](#)
5. **HSBC (présent en France)** — le groupe HSBC a choisi Exadata Cloud@Customer pour moderniser une partie de ses bases Oracle ; même si l'annonce est globale, HSBC opère un important périmètre en France. [computerweekly.com](#)

EXADATA VS EMC/DELL POWERMAX

⚙️ 1. Oracle Exadata

- **Type** : Appliance (base de données + compute + stockage optimisés pour Oracle).
 - **Stockage** : inclus, basé sur des serveurs de stockage Exadata avec disques NVMe/Flash + software Oracle Smart Scan.
 - **Usage** : conçu uniquement pour Oracle Database (OLTP, Data Warehouse, consolidation).
 - **Forces** :
 - Performance extrême grâce au **traitement des requêtes dans le stockage** (Smart Scan).
 - Administration simplifiée (stack intégré Oracle).
 - Optimisations logicielles/hardware uniques (RAC, Infiniband/RoCE, compression Hybrid Columnar).
 - **Limite** : c'est une solution **fermée**, optimisée uniquement pour Oracle. Impossible de stocker des fichiers/applications non Oracle.
-

⚙️ 2. EMC Symmetrix VMAX (désormais PowerMax)

- **Type** : Baie de stockage haut de gamme, multi-protocoles (Fibre Channel, iSCSI, NVMe-oF...).
- **Stockage** : centralisé, virtualisé, avec de très gros volumes et réplication synchrone/asynchrone.
- **Usage** : stockage partagé pour **plusieurs serveurs/applications** (Oracle, SQL Server, SAP, VMware, fichiers, etc.).
- **Forces** :
 - Extrêmement fiable ("**mainframe-class**" **availability**).
 - Multi-tenant : un seul VMAX peut servir Oracle, SAP, Exchange, VMware, etc.
 - Fonctionnalités avancées de réplication (SRDF, SnapVX), tiering automatique (Flash/SSD/HDD).
- **Limite** :
 - Ne bénéficie pas des optimisations logicielles **spécifiques Oracle** (pas de Smart Scan).
 - Dépend des serveurs Oracle classiques → moins performant que l'appliance Exadata pour Oracle pur.

Comparatif direct

Critère	Oracle Exadata	Symmetrix VMAX
Nature	Appliance DB spécialisée	Baie de stockage générique haut de gamme
Usage	Exclusivement Oracle Database	Toutes bases & applications (Oracle, SAP, VMware...)
Performance Oracle	Optimisé (Smart Scan, offload SQL dans stockage)	Dépend des serveurs + réseau (moins optimisé)
Administration	Simplifiée (stack Oracle intégré)	Plus complexe (intégrer avec serveurs/applications)
Flexibilité	Fermée (Oracle only)	Ouverte (multi-usage, multi-OS, multi-DB)
Disponibilité	Haute (RAC + Data Guard)	Très haute (SRDF, active-active, mainframe-level)
Déploiement	Cloud, Cloud@Customer, appliance	Datacenter (on-premises), parfois cloud hybride

FICHE TECHNIQUE ORACLE EXADATA

AGENDA ARCHITECTURE

- 1. Présentation générale
- 2. Architecture matérielle
- 3. Fonctionnalités clés DBA
- 4. Outils d'administration
- 5. Sauvegarde & Restauration
- 6. Scénarios d'utilisation
- 7. Points DBA critiques en production
- 8. Questions fréquentes

1 Présentation générale

- Exadata = Database Machine : appliance optimisée pour exécuter Oracle Database.
 - Composants :
 - Compute Nodes (Database Servers) → exécutent Oracle RAC, ASM, Grid Infrastructure.
 - Storage Servers (Cells, "Exadata Storage Servers") → gèrent disques + Smart Scan + Flash Cache.
 - Réseau interne → RoCE (anciennement InfiniBand) pour faible latence.
 - Administration → Oracle Linux, Exadata System Software.
-

2 Architecture matérielle

- Database Servers (Compute Nodes) :
 - CPU : serveurs x86 (Intel Xeon/AMD EPYC, > 64 cœurs)
 - RAM : centaines de Go (jusqu'à plusieurs To)
 - Oracle RAC + ASM s'exécutent dessus.
- Exadata Storage Servers (Cells) :
 - Disques Flash (NVMe SSD) + HDD (selon config : Extreme Flash / High Capacity).
 - Software spécifique : Exadata Storage Server Software (cellsrv).
 - Gèrent le traitement SQL directement dans le stockage (Smart Scan).
 - Fonctionnalités : Hybrid Columnar Compression, Flash Cache, IORM.
- Réseau interne :
 - RoCE (RDMA over Converged Ethernet) 100 Gb/s (anciennement InfiniBand).
 - Séparation réseau : Client → DB servers / DB servers → Storage servers.
- Réseau externe :
 - Ethernet (1/10/25/40/100 Gb) pour clients et backups.

3 Fonctionnalités clés DBA

◆ Smart Scan

- Déplacement du traitement SQL vers les **storage servers**.
- Exécution des filtres WHERE, des projections (colonnes), des agrégats directement au niveau du stockage.
- Réduit trafic I/O → accélère fortement les requêtes.

◆ Storage Index

- Stocke min/max des blocs dans la mémoire du storage server.
- Permet d'éviter des lectures inutiles (skip scan).

◆ Hybrid Columnar Compression (HCC)

- Compression très efficace pour les tables historiques / entrepôts.
- Gain de place 10x à 50x.

◆ Flash Cache

- Cache SSD intelligent dans les Storage Cells.
- Priorisation via IORM (I/O Resource Manager).

◆ IORM (I/O Resource Manager)

- Permet de gérer la QoS (Quality of Service) des workloads.
- Exemple : OLTP prioritaire sur batch.

◆ Gestion patch & upgrades

- **opatchauto** (pour Grid Infrastructure & RDBMS).
- **patchmgr** (pour Exadata System Software, compute + storage).
- Rolling patch (pas d'interruption totale).

4 Outils d'administration

- CellCLI → admin local d'un storage server.
 - DBMCLI → admin de la DB Machine (global).
 - DCLI → exécuter une commande sur tous les nœuds en parallèle.
 - ExaCLI → client distant d'administration.
 - OEM (Oracle Enterprise Manager) + Cloud Control pour supervision centralisée.
 - Exachk → healthcheck automatisé de l'infrastructure.
 - Diag pack & TFA (Trace File Analyzer) → diagnostic avancé.
-

5 Sauvegarde & Restauration

- RMAN → sauvegardes classiques Oracle.
 - Intégration avec ZDLRA (Zero Data Loss Recovery Appliance).
 - Possibilité de sauvegarde vers Oracle Cloud Object Storage.
-

6 Scénarios d'utilisation

- OLTP à très haut débit (banques, télécoms).
- Data Warehouse (requêtes massives).
- Consolidation multi-databases Oracle.
- Cloud hybride (Exadata Cloud@Customer, Exadata dans OCI).

- Connaître Smart Scan (fonctionnement et cas où il s'applique ou pas).
- Différence Storage Index vs Index B-Tree.
- Comprendre IORM et gestion de la QoS.
- Savoir patcher un Exadata (Grid Infrastructure, RDBMS, Cell, Compute).
- Savoir expliquer compression HCC (mode Archive, Query High, Query Low, etc.).
- Savoir administrer avec CellCLI, DBMCLI, DCLI.
- Différence entre Exadata X9M (dernier modèle) et les versions précédentes.
- Rôle du RoCE / InfiniBand dans les perfs.

1. Explique le Smart Scan et dans quel cas il ne s'applique pas.
→ Exemple : tables small, index access path, LOBs non stockés en BasicFile.
2. Comment fonctionne la compression Hybrid Columnar Compression (HCC) ?
3. Comment patches-tu un Exadata sans downtime ?
4. Quelle est la différence entre ASM et Exadata Storage Software ?
5. Comment gérer une contention I/O entre OLTP et batch ?
→ Réponse : via IORM.
6. Quelles sont les différences entre Exadata on-prem, Exadata Cloud Service, Exadata Cloud@Customer ?
7. Quels outils utilises-tu pour diagnostiquer une panne dans un Storage Cell ?

ORACLE EXADATA EN PROFONDEUR

AGENDA DES BRIQUES ESSENTIELLES

1. FICHE TECHNIQUE – SMART SCAN

2. FICHE TECHNIQUE – STORAGE INDEX

3. FICHE TECHNIQUE – HYBRID COLUMNAR COMPRESSION (HCC)

4. FICHE TECHNIQUE – I/O RESOURCE MANAGER (IORM)

5. FICHE TECHNIQUE – SMART FLASH CACHE

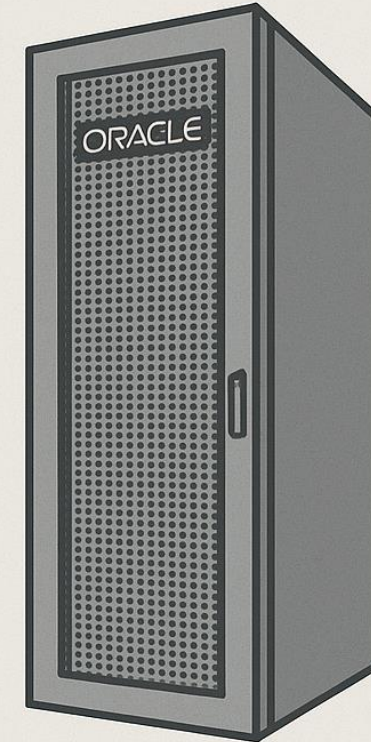
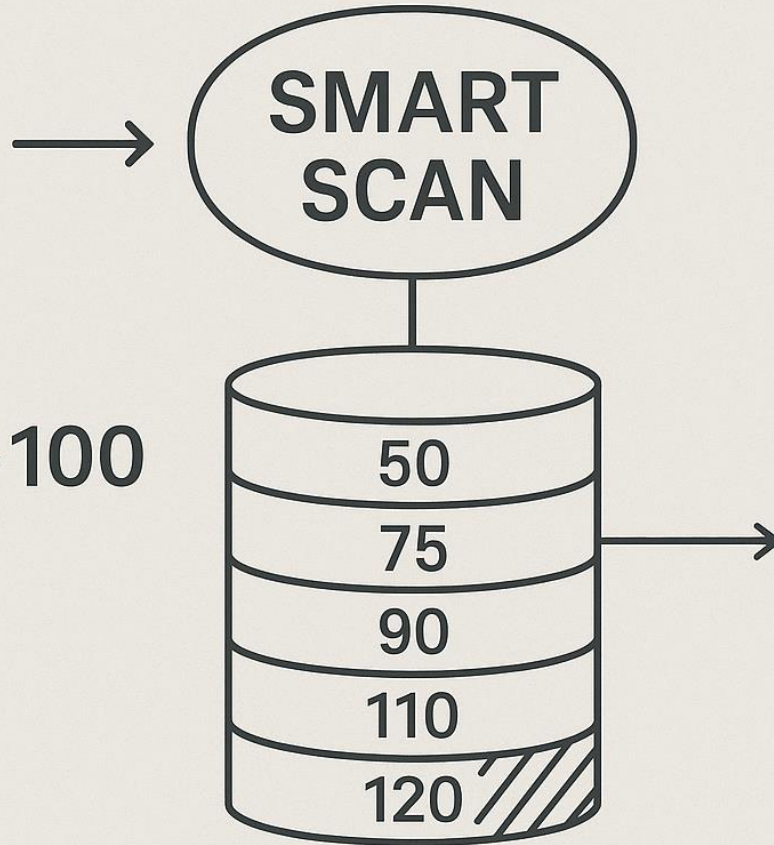
6. FICHE TECHNIQUE – RÉSEAU (INFINIBAND / ROCE)

7. FICHE TECHNIQUE – STORAGE SERVER CELLSRV

8. FICHE TECHNIQUE – EXACHK & PATCHING

SMART SCAN

**SELECT
colonne
FROM
table
WHERE
colonne > 100**



**ORACLE
EXADATA**

1 Définition

- **Smart Scan** = technologie qui permet d'exécuter une partie des traitements SQL directement dans les serveurs de stockage Exadata (Cells) au lieu de tout renvoyer vers les serveurs de base de données.
 - Objectif : réduire le volume de données transférées et accélérer l'exécution des requêtes.
-

2 Fonctionnement

1. Requête SQL lancée → serveur de DB envoie la requête au **storage server (Cell)**.
 2. Le **storage server exécute localement** certains traitements :
 - Filtrage WHERE (predicate filtering).
 - Projection → sélection uniquement des colonnes nécessaires.
 - Certaines fonctions SQL (aggregates, joins dans certains cas).
 3. Résultat filtré et réduit → renvoyé au serveur DB.
 4. Moins d'I/O + moins de trafic réseau = **gain massif de performance**.
-

3 Bénéfices

- Réduction drastique du volume d'I/O.
- Accélération des requêtes analytiques (Data Warehouse).
- Soulage le CPU des serveurs DB.
- Améliore la scalabilité pour OLTP + BI.

4 Exemple concret

sql

```
SELECT customer_id, SUM(amount)
FROM transactions
WHERE region = 'EUROPE'
AND date >= '01-JAN-2024';
```

DML SAMPLE

- **Serveur classique :**
 - lit toutes les lignes, envoie au DB server, puis filtre et agrège.
- **Exadata Smart Scan :**
 - le storage server lit uniquement les lignes **region = 'EUROPE'** et **date >= 2024**, calcule **SUM(amount)**, et renvoie seulement le résultat agrégé.

👉 Résultat : jusqu'à 100x plus rapide dans certains cas.

5 Limitations (quand Smart Scan ne s'applique PAS)

- Accès via **Index unique** ou **small lookup** (pas utile, car peu de données à lire).
- **LOBs** stockés en BasicFile (sauf SecureFiles dans certains cas).
- Données en mémoire (buffer cache) → pas de Smart Scan.
- Certaines opérations PL/SQL ou fonctions non prises en charge.

6 Outils DBA associés

- **V\$SQL, V\$CELL_OFFLOAD_PLAN_DISPLAY** → pour vérifier l'utilisation du Smart Scan.
- **Autotrace / SQL Monitor** → voir si la requête a été offloadée.

7

QUESTIONS FREQUENTES

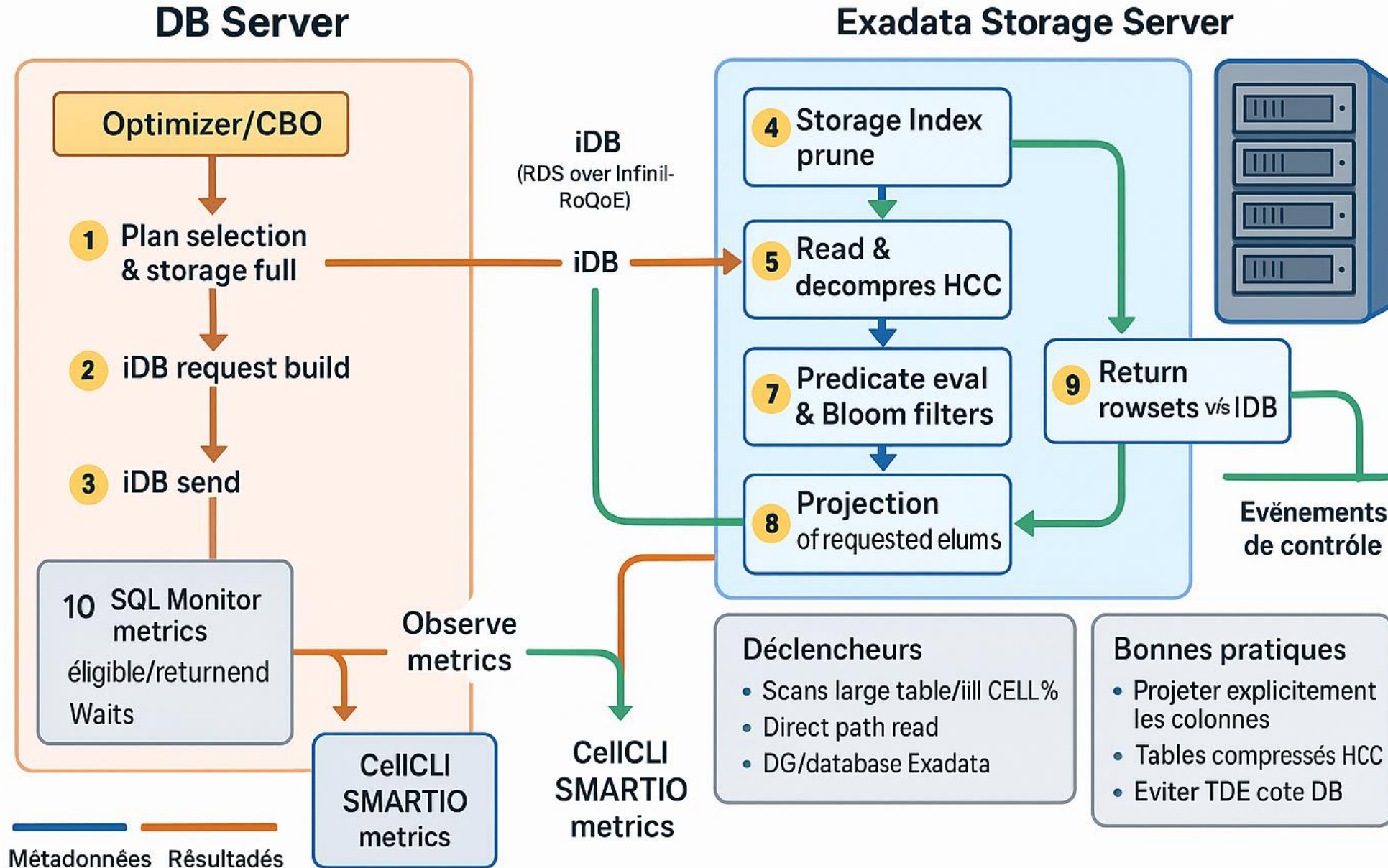
- Qu'est-ce que Smart Scan et comment ça marche ?
- Donne un cas où Smart Scan ne s'applique pas.
- Quels sont les bénéfices concrets pour un Data Warehouse ?
- Comment vérifier si une requête utilise Smart Scan ?



LES MECANISMES INTERNES

Smart Scan

IDEO-LAB



1) Rappel express

Smart Scan pousse une partie du traitement SQL (filtrage de lignes, projection de colonnes, filtrage de jointure via Bloom filters, (dé)compression, parfois déchiffrement) dans les Storage Servers (cellsrv). On ne renvoie au DB Server que les lignes/colonnes utiles au lieu d'expédier des blocs entiers. C'est transparent pour l'application. [oracle.com](#) [docs.oracle.com](#)

2) Quand Smart Scan se déclenche

- L'optimiseur choisit un **full table scan**, **fast full index scan** ou **full bitmap index scan** (souvent en **direct path**), et la base détecte que les segments résident sur des **grid disks Exadata**. [oracle.com](#) [docs.oracle.com](#)
 - La session reste avec `cell_offload_processing=TRUE` (par défaut sur Exadata). Si on le force à `FALSE`, on voit des événements *"cell smart ... : disabled by user"*. [docs.oracle.com](#)
-

3) Pipeline interne — pas à pas

Côté Database Server (instance Oracle)

1. **Plan** : le CBO choisit un scan "offloadable" (ex. `TABLE ACCESS STORAGE FULL`). [Oracle Diagnostician](#)
2. **Découpage par cellule** : l'instance construit, pour chaque cell, un paquet iDB (RDS/IB/RoCE) décrivant :
 - la liste de colonnes à projeter,
 - les prédicats (`WHERE`) à évaluer,
 - des Bloom filters pour les jointures (si présents),
 - la liste d'extents/blocs concernés. [oracle.com](#) [docs.oracle.com](#)
3. **Envoi** : ces métadonnées partent en parallèle vers toutes les cells concernées via iDB. [oracle.com](#)

Côté Storage Server (cellsrv)

4. **Pruning en amont** : avant tout, la cell tente d'éviter l'I/O inutile via **Storage Indexes** (min/max par zone de 1 Mo gardés en mémoire). Si les prédicats sont hors [min,max], **aucun bloc n'est lu**. [oracle.com](#)
5. **Lecture & (dé)compression** : les blocs utiles sont lus (disque/flash).
 - Si les données sont en HCC : la **décompression & la projection** se font dans la cell, colonne par colonne (utile pour ne décompresser que ce qu'on lit). [docs.oracle.com](#) [oracle.com](#)
6. **Déchiffrement (TDE) possible dans la cell** (tablespaces ou colonnes chiffrées) ; le filtrage peut alors s'appliquer sur **données déchiffrées côté storage**. [docs.oracle.com](#)
7. **Évaluation des prédicats** : comparaisons, BETWEEN/IN, LIKE (cas simples), etc. + **Bloom filters** pour éliminer tôt les lignes qui ne "matchent" pas la jointure. [Pythian](#)
8. **Projection** : la cell fabrique un jeu de résultats réduit (lignes/colonnes demandées) et n'envoie pas des blocs mais des rowsets filtrés au DB Server via iDB. [oracle.com](#)

4) Ce qui est offloadable

- Scans : Full table / Fast full index / Full bitmap index. [oracle.com](#)
- Filtrage : prédicats relationnels standards ; **Bloom filters** pour les jointures (star schemas/PDQ). [Pythian](#)
- Projection de colonnes (évite de remonter "SELECT *"). [oracle.com](#)
- (Dé)compression HCC au storage (épargne CPU DB). [docs.oracle.com](#)
- Chiffrement TDE (décryptage offloadé puis filtrage). [oracle.com](#) [docs.oracle.com](#)
- Fonctions spécifiques : ex. scoring `PREDICTION_PROBABILITY` (Data Mining), et certaines opérations LOB (12.2+ sur LOB inlined <4KB : `SUBSTR/LENGTH/INSTR/...` , y compris quelques REGEXP).

[docs.oracle.com +1](#)

5) Ce qui ne s'offloade généralement pas

- Accès row-by-row (nested loops avec lookups), **index range scans** classiques, **petites lectures** mono-bloc.
- Prédicats non déterministes/complexes, UDF/PLSQL arbitraires, motifs **LIKE** avec wildcard en tête, etc. *(Les règles exactes évoluent par version, mais l'idée est : tout ce qui empêche un grand scan séquentiel ou une évaluation "poussable" au storage).* [oracle.com](#)

6) Comment vérifier qu'un Smart Scan a eu lieu

a) Plan & SQL Monitoring

- Le plan doit montrer `STORAGE` (ex. `TABLE ACCESS STORAGE FULL`).
- Rapport SQL Monitor : colonnes `IO_CELL_OFFLOAD_ELIGIBLE_BYTES` et `IO_CELL_OFFLOAD_RETURNED_BYTES` par SQL_ID. Un ratio Returned/Eligible très bas = **offload efficace**.

6) Comment vérifier qu'un Smart Scan a eu lieu

a) Plan & SQL Monitoring

- Le plan doit montrer `STORAGE` (ex. `TABLE ACCESS STORAGE FULL`).
- Rapport `SQL Monitor` : colonnes `IO_CELL_OFFLOAD_ELIGIBLE_BYTES` et `IO_CELL_OFFLOAD_RETURNED_BYTES` par `SQL_ID`. Un ratio `Returned/Eligible` très bas = **offload efficace**.

docs.oracle.com

docs.aws.amazon.com

```
sql
```

```
-- Dernier SQL monitoré pour un SQL_ID
SELECT key, status, sql_id, sql_text, elapsed_time,
       io_cell_offload_eligible_bytes AS eligible,
       io_cell_offload_returned_bytes AS returned
FROM   v$sql_monitor
WHERE  sql_id = :sql_id
ORDER BY key DESC;
```

b) Statistiques & wait events (vue instance/session)

- Stats clés (`V$SYSSTAT/V$SESSTAT`) :

`IO_CELL_OFFLOAD_ELIGIBLE_BYTES`,

`IO_CELL_OFFLOAD_RETURNED_BYTES`,

`cell physical IO interconnect bytes`,

`cell physical IO bytes saved by storage index`.

docs.oracle.com +1

- Waits caractéristiques :

`cell smart table scan`, `cell smart index scan`, `cell external table smart scan`.

docs.oracle.com

[Frits Hoogland Weblog](#)

```
-- Efficacité offload au niveau système (depuis le démarrage)
SELECT
  name, value
FROM v$sysstat
WHERE name IN (
  'cell physical IO interconnect bytes',
  'cell physical IO bytes saved by storage index',
  'cell physical IO bytes eligible for predicate offload',
  'cell physical IO interconnect bytes returned by smart scan'
);

-- Waits Exadata visibles sur une session
SELECT event, total_waits, time_waited
FROM v$session_event
WHERE sid = :sid
AND event LIKE 'cell smart%';
```

c) Côté Storage (CellCLI)

- Smart I/O metrics dans `METRICCURRENT/METRICHISTORY` (`objectType=SMARTIO`). On peut suivre `...OFFLOAD...` et `...STORIDX...` pour voir l'I/O économisée. [docs.oracle.com +1](#)

bash

Cmd CellCLI

Exemples (sur une cell)

```
cellcli -e "list metriccurrent where objectType='SMARTIO'"
```

```
cellcli -e "list metrichistory where objectType='SMARTIO' and name like 'DB_IO_OFFLOAD_%'"
```

7) Interactions clés

- HCC + Smart Scan : la cell décompresse uniquement les colonnes lues, applique les prédicats, et n'envoie que le nécessaire → CPU DB et interconnect soulagés. [docs.oracle.com](#)
- TDE + Smart Scan : décryptage offloadé en cell puis filtrage/projection au storage. [oracle.com](#) [docs.oracle.com](#)
- Bloom filters : les filtres de jointure (ex. star) sont poussés aux cells pour éliminer en amont les lignes du gros fact table. [Pythian](#) [Tanel Poder Consulting](#)

8) Pourquoi un Smart Scan ne part pas (causes usuelles)

- Le segment accédé n'est pas 100% sur des grid disks Exadata (DG mixte, NFS, ou ASM non-Exadata). [docs.oracle.com](#)
- L'accès n'est pas un scan séquentiel (index range, nested loops, petits reads). [oracle.com](#)
- Session ou SQL désactive l'offload (`ALTER SESSION SET cell_offload_processing=FALSE`). [docs.oracle.com](#)

9) Bonnes pratiques pour maximiser Smart Scan

- Écrire des requêtes set-based (éviter les lookups row-by-row) et laisser l'optimiseur choisir des scans pertinents. [oracle.com](#)
- Projeter explicitement les colonnes (éviter `SELECT *`). [oracle.com](#)
- Sur gros entrepôts : HCC sur les segments froids et partitions lisibles en grand. [docs.oracle.com](#)
- Laisser TDE côté cell quand pertinent (gains CPU DB). [oracle.com](#)
- Vérifier les ratios Eligible vs Returned par SQL et côté cell (SMARTIO) ; un ratio faible Returned/Eligible = offload performant. [docs.aws.amazon.com](#)

10) Checklist "diagnostic rapide"

1. Plan contient `STORAGE` ? (oui/non) [Oracle Diagnostician](#)
2. `IO_CELL_OFFLOAD_ELIGIBLE_BYTES` \gg `IO_CELL_OFFLOAD_RETURNED_BYTES` ? (SQL Monitor) [docs.oracle.com](#)
3. Waits `cell smart table scan` visibles ? [Frits Hoogland Weblog](#)
4. `cell physical IO bytes saved by storage index` > 0 ? (session/sys) [docs.oracle.com](#)
5. Sur la cell : métriques SMARTIO actives ? [docs.oracle.com](#)

QUESTIONS FREQUENTES

Q1. Qu'est-ce qu'Oracle Exadata ?

- Appliance matérielle + logicielle optimisée pour Oracle Database
- Intègre compute nodes, storage servers, réseau RoCE/InfiniBand
- Objectif : haute performance + haute disponibilité + consolidation

Q2. Différence Exadata vs stockage classique (EMC, NetApp, etc.) ?

- Exadata → intelligence dans le stockage (Smart Scan, Storage Index)
- Stockage générique → simple I/O, pas d'optimisation SQL

Q3. Qu'est-ce que Smart Scan ?

- Exécution des filtres/projections SQL directement dans le storage server
- Réduit trafic I/O, accélère fortement les requêtes

Q4. Quand Smart Scan ne s'applique pas ?

- Accès via index unique
- LOBs en BasicFile
- Données déjà en cache mémoire

Q5. Qu'est-ce qu'un Storage Index ?

- Min/Max des blocs en mémoire du storage server
- Permet d'éviter la lecture de blocs inutiles

Q6. Comment gérer la QoS entre OLTP et batch ?

- Via IORM (I/O Resource Manager) → priorisation des workloads

Q7. Quels outils d'administration Exadata ?

- CellCLI, DBMCLI, DCLI, ExaCLI, Exachk
- OEM (Enterprise Manager) pour supervision globale

Q8. Comment patcher un Exadata sans downtime ?

- Rolling patching via `opatchauto` (DB/Grid) et `patchmgr` (Cells/Compute)

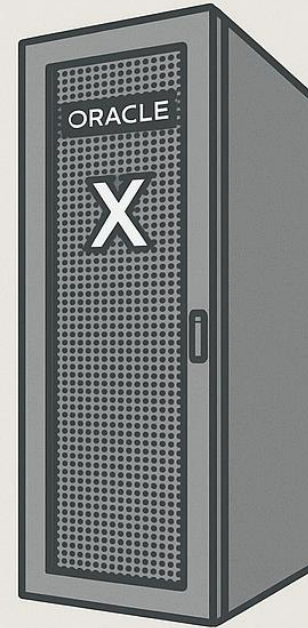
STORAGE INDEX

10	20	30
10	20	40
10	20	40
10	20	40

WHERE valeur = 50 ✗

WHERE valeur = 25 ✓

STORAGE INDEX
Min = 10 / Max = 40
Min = 10 / Max = 40
Min = 10 / Max = 40



ORACLE
EXADATA

Oracle Exadata – Storage Index

Vue technique détaillée



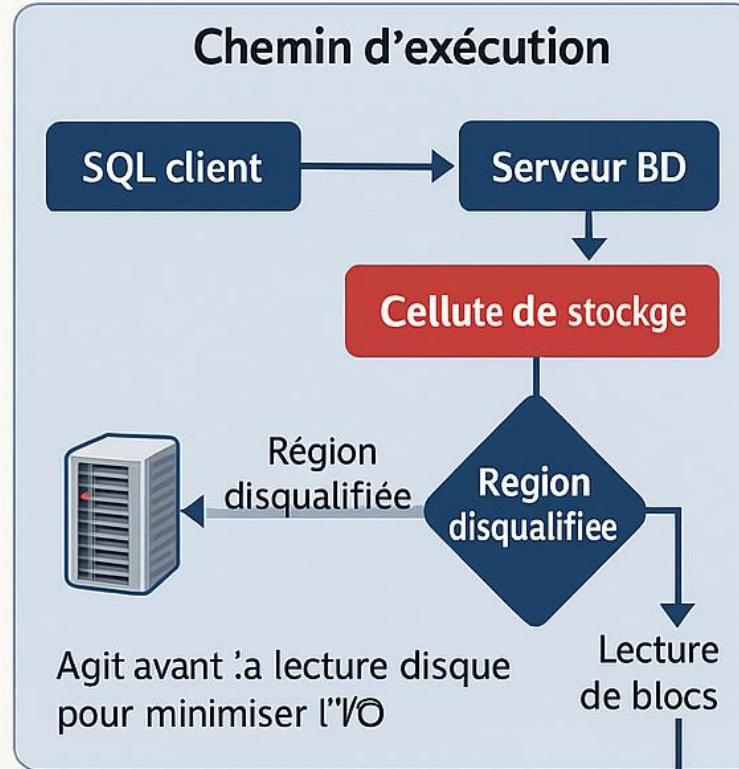
Principe

- Résident en mémoire sur les cellules de stockage (pas d'objet Oracle)
- Résume, par region de 1Mo, min/max par colonne + information NULL
- Cree/maintenu automatiquement à la lecture (Smart Scan)
- Utilisé lorsque des prédicats filtrants existent (WHERE, BETWEEN, IN, =, >=, <=)

Limitations

- Inefficace si prédicats peu sélectifs. non déterministes, LIKE %' : : colonnes à faible débit. débragent non possible

Chemin d'exécution



Bonnes pratiques

- Utiliser prédicats filtrables: éviter TO_CHAR/CAST sur colonne filtrée; p.ex.

Bonnes pratiques

- Utiliser prédicats filtrables: éviter TO_CHAR/CAST sur colonne

Taille & granularité

- 1 Mo typiquement

Cas d'usage

- Charge DWH/analytique, grandes tables, partitionnées
- Colonnés de filtres: date, id_client
- Synergie avec HCC et prédicats poussés

Limitations

- v\$sysstat / CELLCLI metricie «cell physical IO bytes saved by storage index»
- «ceil physical IO interconnect bytes saved by storage

1 Définition

- Le Storage Index (SI) est une métadonnée en mémoire maintenue par les serveurs de stockage Exadata.
- Objectif : éviter la lecture de blocs inutiles en se basant sur des valeurs min/max stockées pour chaque région de disque (1 Mo).

👉 Résultat : réduction massive d'I/O → performance accrue.

2 Fonctionnement

1. Lorsqu'une requête SQL contient une clause de filtre (WHERE), Exadata consulte le Storage Index.
2. Le SI contient, pour chaque colonne et chaque "region" (1 MB de stockage) :
 - Valeur MIN
 - Valeur MAX
3. Si la valeur recherchée ne peut pas être dans cette plage, le bloc entier est ignoré.
4. Les blocs "candidats" seulement sont lus depuis le disque → gain en I/O.

3 Exemple concret

sql

```
SELECT * FROM SALES WHERE SALE_DATE = '15-MAR-2025';
```

DML SAMPLE

- **Sans Storage Index :**
 - le storage lit tous les blocs de la table SALES.
- **Avec Storage Index :**
 - Exadata sait que dans certains blocs, `SALE_DATE` est uniquement entre '01-JAN-2025' et '28-FEB-2025'.
 - Ces blocs sont **ignorés** immédiatement.

👉 Réduction I/O = facteur 10x à 50x selon la sélectivité.

4 Caractéristiques clés

- Automatique : pas besoin de création ni de gestion DBA.
- Volatile : maintenu **en mémoire**, pas stocké sur disque (perdu si redémarrage, mais se reconstruit avec l'activité).
- Fonctionne main dans la main avec **Smart Scan**.
- Très efficace pour :
 - Colonnes de dates
 - Colonnes numériques à forte sélectivité

6 Outils DBA associés

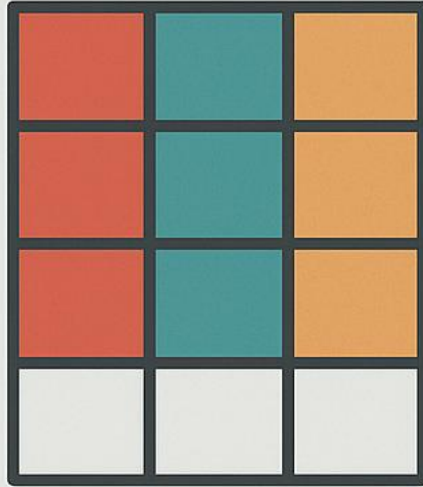
- Vues dynamiques :
 - V\$SYSSTAT → vérifier `cell physical IO bytes saved by storage index`
 - V\$MYSTAT (par session)
- SQL Monitor → voir si le SI a été utilisé.

7 Questions fréquentes

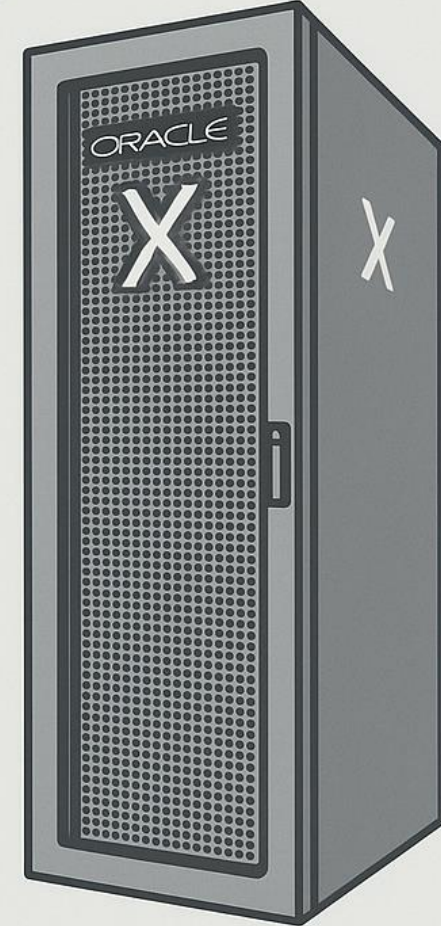
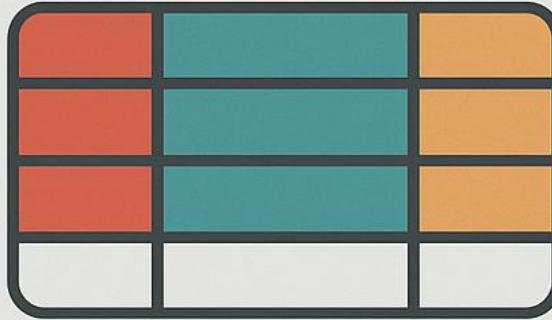
- Qu'est-ce qu'un Storage Index et à quoi ça sert ?
- Quelle est la différence avec un index Oracle classique (B-tree, bitmap) ?
- Pourquoi dit-on que le Storage Index est "in-memory" et "éphémère" ?
- Comment vérifier s'il a été utilisé dans une requête ?

HCC

HCC GRAPHIC



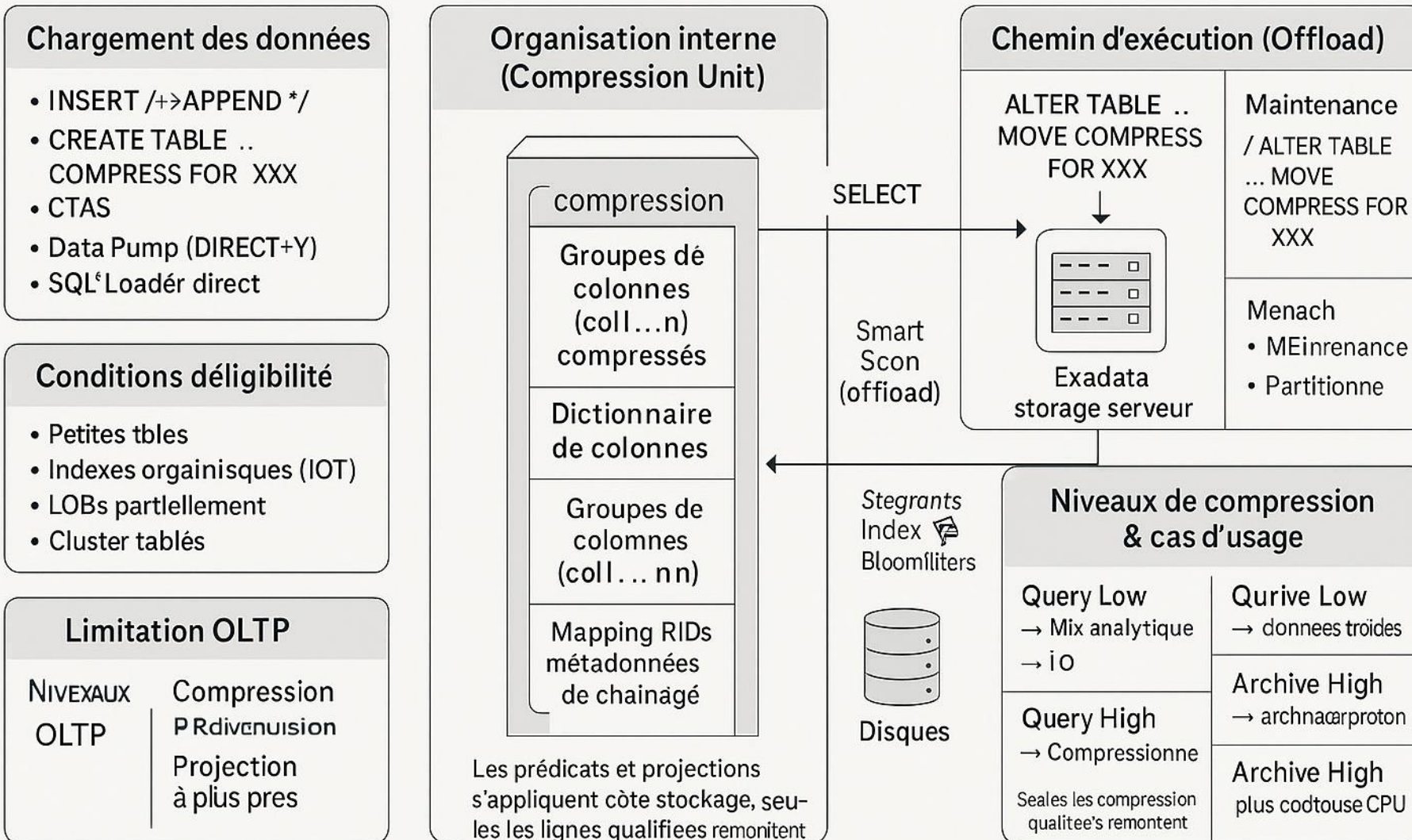
**HYBRID
COLUMNAR
COMPRESSION**



ORACLE

Oracle Exadata – Hybrid Columnar Compression (HCC)

IDEO
LAB



Oracle Exadata – Hybrid Columnar Compression (HCC)



IDEO•LAB

Vue technique détaillée

Chargement des données

- INSERT / + APPEND)
- CTAS
- Data Pump
- SQL/Loader

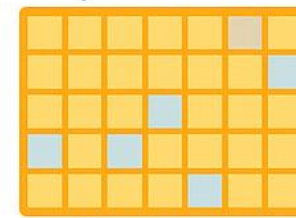
Conditions d'éligibilité

- Chemin direct requis
- Non pris en charge pour IOT
- DML fréquents déconseillés



Conditions d'éligibilité

Organisation interne



Smart Scan (offload)



Serveurs de stockage Exadata

Niveaux de compression & cas d'usage

- Query Low
Balance entre performances et compression
- Query High
Performances des scans
- Archive Low
Economie-de stockage
- Archive High
Plus forte compression

Limitations OLTP

HCC nécessite charges en chemin direct, inadapté aux DML fréquentes

1 Définition

- **Hybrid Columnar Compression (HCC)** = méthode de compression spécifique à Oracle Database, disponible uniquement sur Exadata, ZFS Storage Appliance et Pillar Axiom.
 - Mélange l'approche **row-based** (par ligne) et **column-based** (par colonne).
 - Objectif : **réduire la taille de stockage et améliorer la performance en lecture** sur de gros volumes.
-

2 Fonctionnement

1. Les données sont regroupées en **Compression Units (CU)** de plusieurs milliers de lignes.
 2. Chaque CU est organisée par colonne (columnar) → meilleure répétition des valeurs = meilleure compression.
 3. Les CU sont ensuite stockées dans le segment de table.
 4. Lors d'une requête, Exadata peut lire et décompresser uniquement les colonnes nécessaires.
-

3 Modes de compression

- **Query Low** → équilibre compression / performance (bon pour data warehouse).
- **Query High** → compression maximale, optimisé pour requêtes majoritairement en lecture.
- **Archive Low** → pour données peu accédées, compression plus forte.
- **Archive High** → compression maximale pour archivage longue durée.

4 Bénéfices

- Gain de stockage massif : réduction de 10x à 50x selon le type de données.
 - Réduction de la bande passante I/O (moins de blocs lus).
 - Amélioration de la performance des requêtes analytiques (moins de données à scanner).
 - Réduction du coût de stockage global (moins de disques nécessaires).
-

5 Exemple concret

Table de ventes : 1 To brut → avec HCC en mode Archive High → 100 Go seulement.

→ Lancer une requête sur les ventes 2024 → lecture très rapide car seulement quelques CU concernés.

6 Limitations

- Disponible uniquement sur **Exadata** (et certains autres systèmes Oracle).
- Moins adapté aux tables fortement mises à jour (compression optimisée pour lecture, pas pour OLTP).
- Chaque modification dans un CU peut entraîner un overhead (migration des blocs vers du stockage "non HCC").
- Pas applicable sur :
 - Tablespaces TEMP et UNDO
 - Index
 - Tables avec DML intensif

7 Outils DBA associés

- DBMS_COMPRESSION → pour tester le taux de compression potentiel.
- Vues : DBA_SEGMENTS (taille compressée), V\$SEGSTAT.
- Commandes SQL :

sql

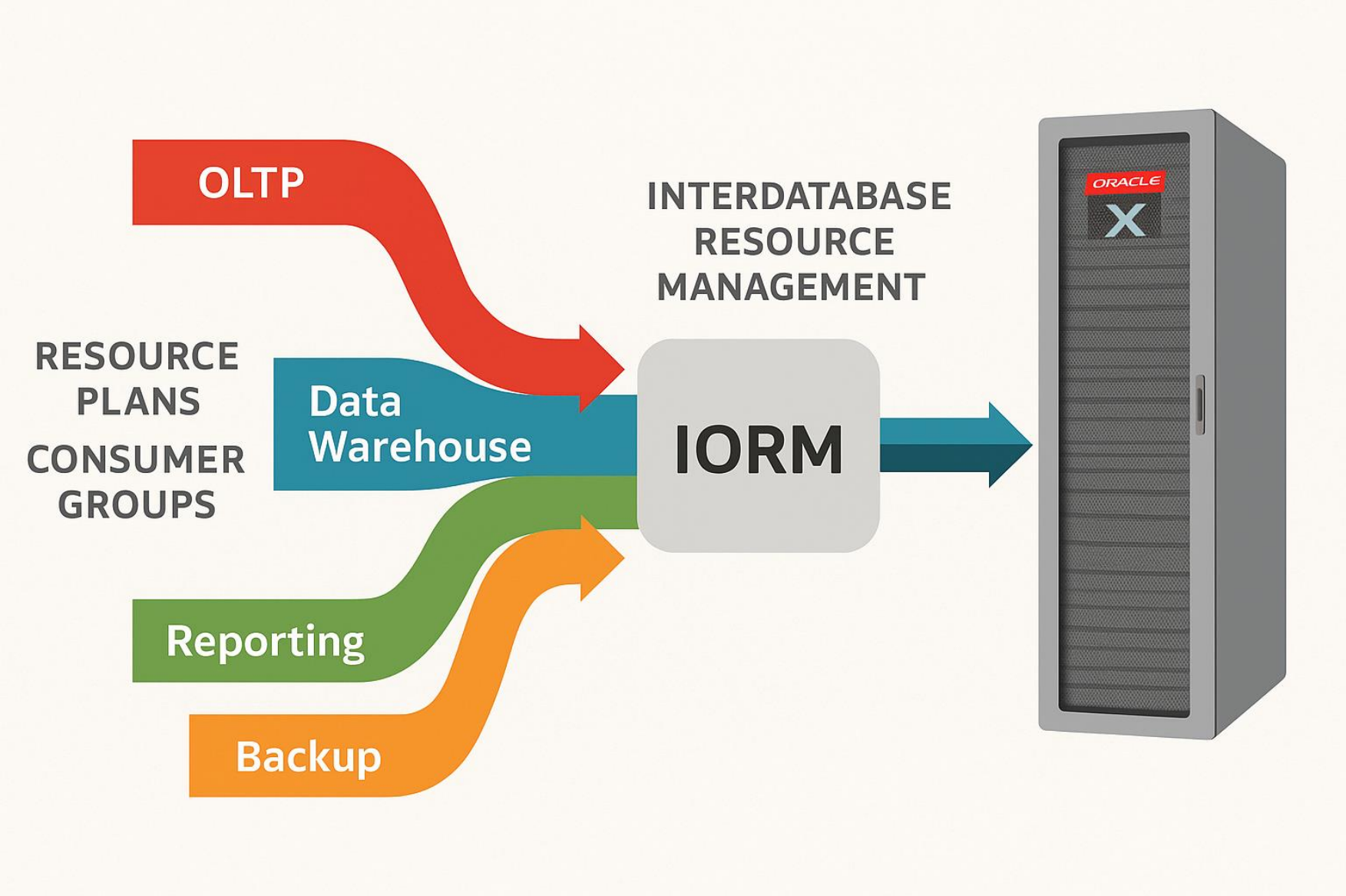
```
ALTER TABLE sales MOVE COMPRESS FOR QUERY HIGH;  
ALTER TABLE archive_data COMPRESS FOR ARCHIVE HIGH;
```

DML SAMPLE

8 Points clés à retenir

- Compression hybride = row + column store.
- Excellent pour data warehouse, reporting, archivage.
- Réduit stockage, accélère lecture.
- Pas recommandé pour OLTP intensif (beaucoup d'INSERT/UPDATE).

IORM



Oracle Exadata – I/O Resource Manager (IORM)



Vue technique détaillée

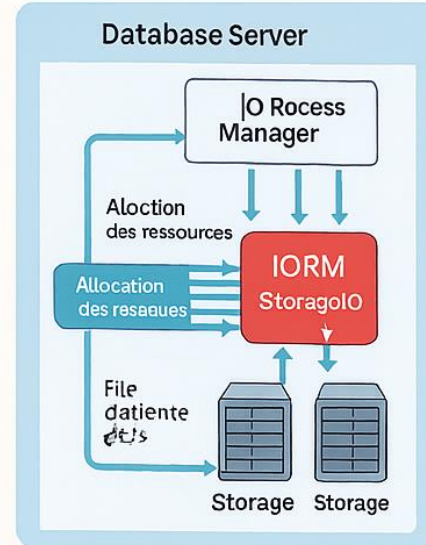
Objectifs

- Prioriser requêtes Pol par catégorie
- Définir politiques gdnise de distribution de io#s
- Limiter l'impact des requetes a laise prioritàs
- Optimiser les temps de reponse

Bonnes pratiques

- Différencier les requetes OLTP et Batch
- Prioritiser les requetes-prodminement interactifs
- Eviter excessivement les
- Spécifier la catégorie défaut pour le reste

Cas d'usage



Exemple de plan (CELLCLI)

```
CREATE IORMPLAN - category(sit0)  
-getarc:max:20)
```

Tailles & granularité

Catégories & directives IORM

- Consumer Group
- Utilisateur
- Base de données

Paramètres & directives

- Level sévice (chare,limit)
- min-max
- Intragroupe, intergroupe
- Administrative/pending

Limitations/ pièges

- Ne pas remplacer au brange demanagement

Tailles & granularité

- Mesures au céle storlage 5'1s seconde

Bonnes pratiques

- Dsi démoritor requêtes au Batch
- Prioriser des tableaux précerie minement interactes
- Eviter excessivement
- Spécifier catégorie défaut pour le reste de l'OI

Limitations/ pièges

- Ne pas remplacer DVIA du workload manag
- Tous lestype OLTP compatible avec un performances au taucr. dé risuré

Checklist mise en œuvre

- Mesurer le leade IIO
- Définier categories et services
- Tester et vallder
- Monitortou IORM

1 Définition

- IORM (I/O Resource Manager) = composant logiciel d'Exadata qui contrôle et répartit les ressources d'entrées/sorties (I/O) entre différentes bases de données, workloads ou utilisateurs.
 - Objectif : **garantir la QoS (Quality of Service)** et éviter qu'une charge gourmande (ex : batch) ne pénalise les autres (ex : OLTP).
-

2 Fonctionnement

- Les **storage servers Exadata** reçoivent toutes les requêtes I/O des serveurs de base.
 - IORM intercepte ces requêtes et applique des règles de **priorité / quotas / limites**.
 - Modes possibles :
 - **Inter-database plan** → répartit les I/O entre plusieurs bases consolidées sur Exadata.
 - **Intra-database plan** → gère la répartition entre *consumer groups* (OLTP, batch, reporting).
 - **Category plans** → regroupement de workloads par type (ex : PROD vs TEST).
-

3 Bénéfices

- **Isolation** des workloads : un batch massif ne bloque pas les transactions OLTP.
- **Priorisation** : les workloads critiques (paiements, transactions bancaires) passent avant le reste.
- **Équité** : chaque base obtient une part garantie des ressources I/O.
- **Flexibilité** : ajustement dynamique des ressources selon l'activité.

4 Exemple concret

Un Exadata héberge :

- Base A = OLTP bancaire (mission critique)
- Base B = Data warehouse analytique (batch de nuit)

👉 Sans IORM : si le batch lance un scan massif, les transactions OLTP sont ralenties.

👉 Avec IORM : OLTP garde 70% des ressources I/O, le batch est limité à 30%.

5 Outils DBA associés

- Commande `DBMCLI` → gestion des plans IORM.
- Vues dynamiques :
 - `V$CELL_IORM_PLAN` (plan actif)
 - `V$CELL_IORM_STATS` (statistiques d'utilisation)
- Exemple :

```
bash
```

```
DBMCLI> LIST DBPLAN
```

```
DBMCLI> ALTER IORMPLAN dbplan='prod=70 dw=30'
```

DDL SAMPLE

6 Limitations

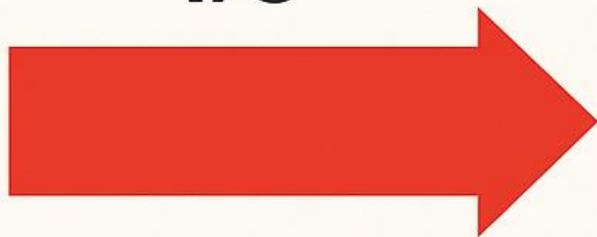
- Ne gère que les I/O Exadata (pas CPU ni mémoire).
- Nécessite un **bon dimensionnement initial** → mauvaise config = perte de performance.
- Pas toujours utile si Exadata héberge une seule base homogène.

7 Points clés à retenir

- IORM = garant de la QoS I/O sur Exadata.
- Permet d'isoler et prioriser les workloads.
- Gestion fine : inter-database, intra-database, categories.
- Outil principal : `DBMCLI` .

SMART FLASH CACHE

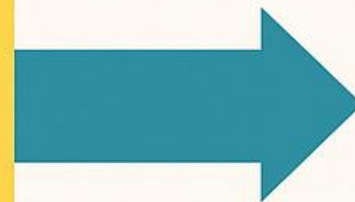
**DATABASE
I/O**



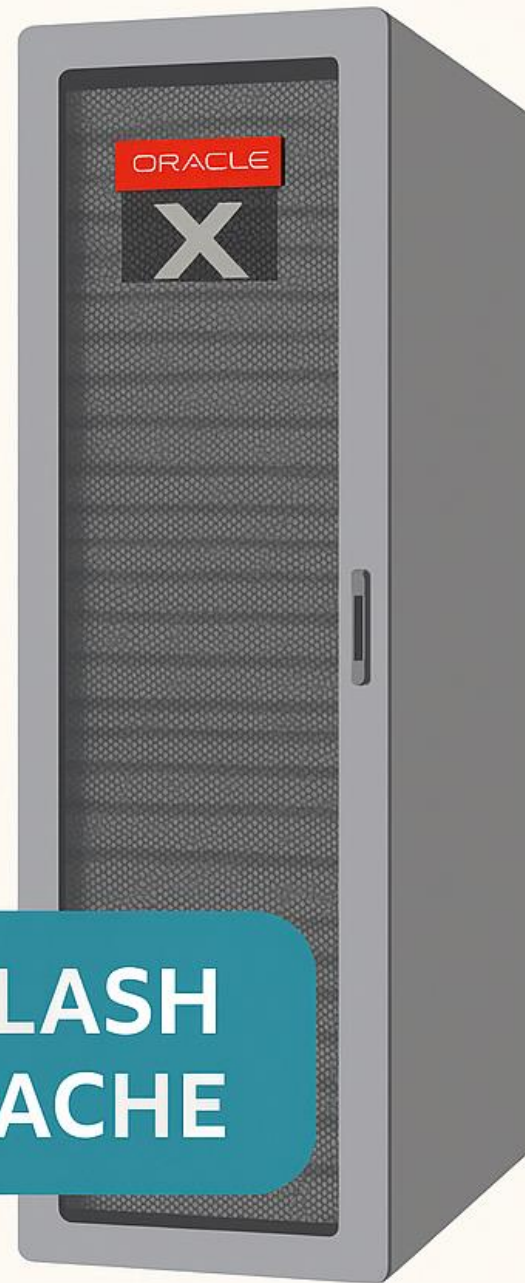
**SMART
FLASH
CACHE**



**DATA
ACCESS**



**FLASH
CACHE**



Oracle Exadata — Cache Flash intelligent

Objectifs

- Réduire la latence de lecture
- Absorber les pics d'écriture
- Protéger la latence de commit (Smart Flash Logging)
- Maximiser l'OPS
priorisation via IORM

Modes du peuplement

- Write-Back pré-procédé flash
- CELL_FLASH_CACHE KEEP/CACHE/ON
- Heuristiques de température, fréquence/recensement, obsolescence, priorité
- Flash cache compression HCC

Smart Flash Logging

- Commits en commits typiques
- Latence typique améliorée
- Écriture dans le flash log et offload complète

Cas d'usage

- Marqueurs intensifs de données
- Mixte de workloads
- Bypass des buffers
- À long terme : grands backups
- SGA, les buffers de mémoire

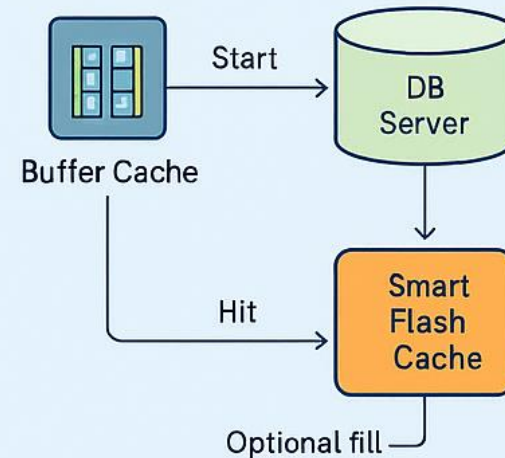
Modes du cache

- Write-Back typique de Flash
- Write-Through typique de Flash

Supervision & métriques

- :flashCacheHits
- :flashCacheReadHits
- :flashCacheWriteHits
- :flashCacheBytes
- :flashLogWriteLatency
- cell physical bytes saved by flash cache

Architecture



Bonnes pratiques

- Maintenez les tables indexées en mode KEEP
- Choisissez Write-Back
- Ayez un grand cold segment
- Surveillez les métriques en région

Limitations & pièges

- Persistance avec précaution
- Full table scans mais pas de cache
- HCC compression après décompression
- Vérifier avant

Taille & granularité

Database block granularité
Idem région et par région

1 Définition

- **Smart Flash Cache** = espace de cache basé sur **disques Flash / NVMe SSD** présents dans les **Exadata Storage Servers**.
 - Objectif : **accélérer les accès disque** en utilisant la rapidité du Flash tout en gardant le HDD pour la capacité.
 - Contrairement à un simple cache OS ou SAN, il est **intégré au moteur Exadata** et "intelligent".
-

2 Fonctionnement

- Les **blocs de données** fréquemment lus ou critiques sont stockés dans le Flash Cache.
 - La décision de mise en cache est **pilotée par Exadata** (pas seulement "hot data").
 - Comportement "intelligent" :
 - Exclut certains I/O (ex : table scan séquentiel inutile à mettre en cache).
 - Optimisé pour les workloads mixtes (OLTP + reporting).
 - Modes possibles :
 - **Write-back** → écrit d'abord dans le cache puis sur disque.
 - **Write-through** → écrit directement sur disque (cache utilisé seulement pour lecture).
-

3 Bénéfices

- **Réduction de la latence** (Flash μ s vs HDD ms).
- **Accélération massive OLTP** : accès aléatoires beaucoup plus rapides.
- **Optimisation des scans** : Smart Scan + Flash Cache = double effet boost.
- **Meilleure consolidation** : plusieurs bases peuvent partager la même infrastructure sans se ralentir.

4 Exemple concret

Une application OLTP effectue 100 000 requêtes aléatoires / seconde :

- Sur HDD : goulot d'étranglement → temps de réponse en millisecondes.
 - Avec Flash Cache : la majorité des blocs réside dans le cache → temps de réponse divisé par **10 à 100**.
-

5 Interaction avec IORM

- IORM (I/O Resource Manager) peut contrôler l'accès au Flash Cache par workload.
 - Exemple : OLTP prioritaire sur Flash, batch relégué sur HDD.
-

6 Outils DBA associés

- Commandes CellCLI :

```
bash
```

```
LIST FLASHCACHECONTENT DETAIL
```

```
LIST FLASHCACHE
```

CMD CELLCLI

- Vues dynamiques :
 - V\$CELL_FLASH_CACHE
 - V\$CELL_STATE
 - V\$SYSSTAT (statistiques cache hits/misses)

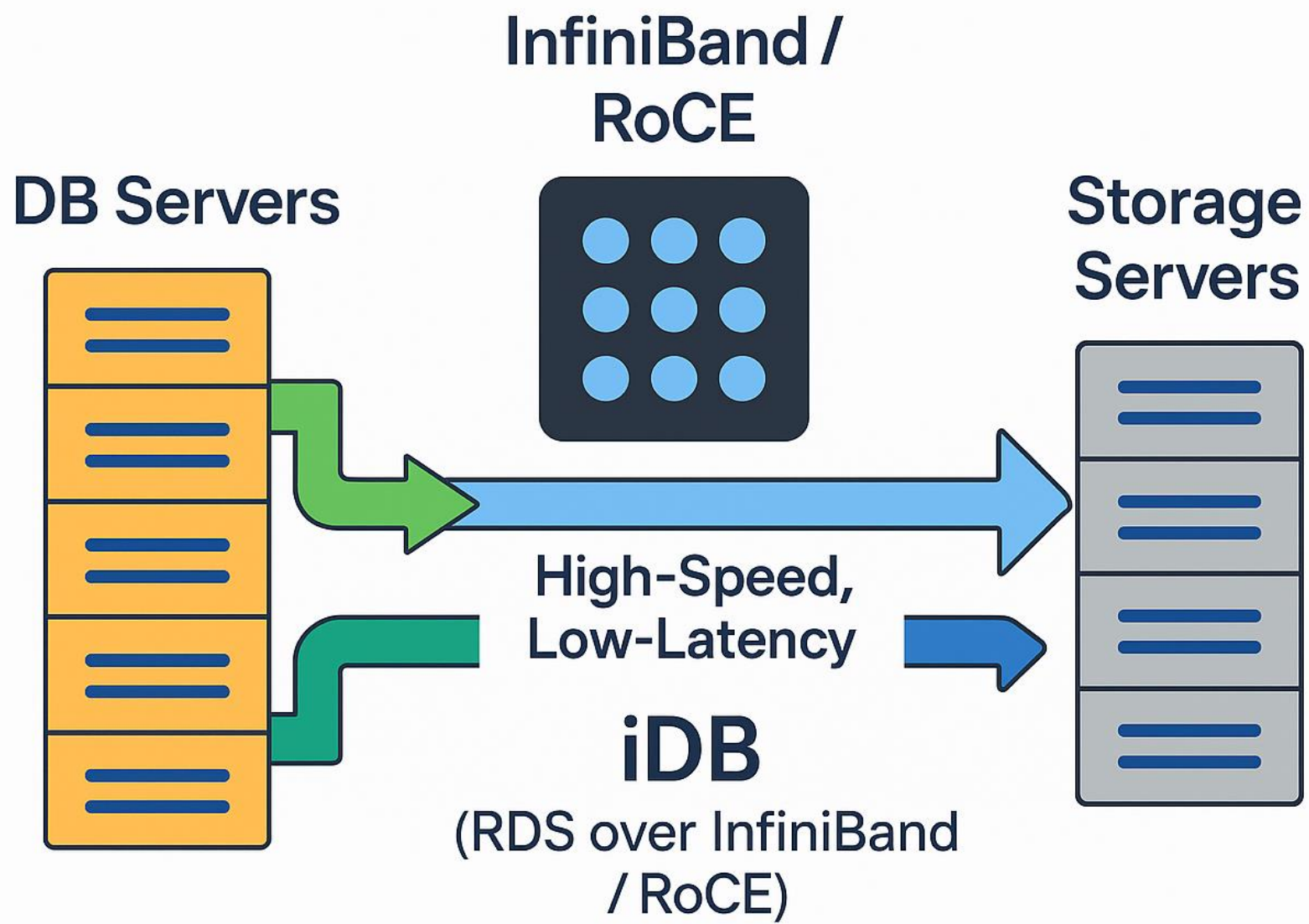
7 Limitations

- Taille limitée par le matériel (SSD/NVMe du modèle).
 - Pas un remplacement du buffer cache SGA (complémentaire).
 - Efficacité dépend du **workload** :
 - Très utile pour OLTP (accès aléatoires).
 - Moins utile pour scans séquentiels massifs (exclus automatiquement).
-

8 Points clés à retenir

- Flash Cache = **accélérateur intelligent** basé sur SSD/NVMe dans Exadata.
- Pilote automatiquement le **placement des données**.
- Combine vitesse Flash + capacité HDD.
- Fonctionne en synergie avec **Smart Scan** et **IORM**.

INFINIBAND / ROCE



1 Définition

- Le réseau Exadata relie les serveurs de base de données (compute nodes) aux serveurs de stockage (cells).
 - Ce réseau interne est **haut débit et très faible latence**, historiquement basé sur InfiniBand, et depuis les modèles récents (X8M, X9M, X10M) sur RoCE (RDMA over Converged Ethernet).
-

2 InfiniBand (jusqu'à X7)

- **Technologie** : réseau dédié HPC (High Performance Computing).
 - **Débit** : 40 Gb/s → 100 Gb/s par lien.
 - **Latence** : < 2 μ s.
 - **Protocole** : RDMA (Remote Direct Memory Access) → accès direct mémoire entre nœuds.
 - **Avantage** : communications ultra rapides pour Smart Scan et RAC inter-nœuds.
-

3 RoCE (X8M, X9M, X10M)

- **Technologie** : Ethernet standard + RDMA.
- **Débit** : 100 GbE (évolutif au-delà).
- **Latence** : similaire à InfiniBand grâce au RDMA.
- **Avantage** :
 - Utilise **Ethernet standard** (interopérabilité accrue).
 - Moins coûteux et plus standard qu'InfiniBand.
 - Maintient les performances (Smart Scan, RAC, stockage).

4 Rôle clé dans Exadata

- Transport des données filtrées par Smart Scan depuis les storage cells vers les DB servers.
- Support des communications RAC inter-nœuds (cache fusion rapide).
- Support des mécanismes de haute disponibilité et de load balancing.
- Permet de maintenir des performances linéaires en scale-out (ajout de nœuds).

5 Bénéfices

- Très faible latence → idéal pour OLTP haute fréquence.
- Très haut débit → nécessaire pour les scans massifs de Data Warehouse.
- Optimisé Oracle → intégration profonde avec RAC, ASM, Smart Scan.
- Évolutif → plus de nœuds = plus de bande passante.

6 Outils DBA & Admin

- Vérification des interfaces :

```
bash

ibstat      # InfiniBand (anciens modèles)
ibhosts     # Liste des hôtes IB
iblinkinfo  # état des Liens IB
```

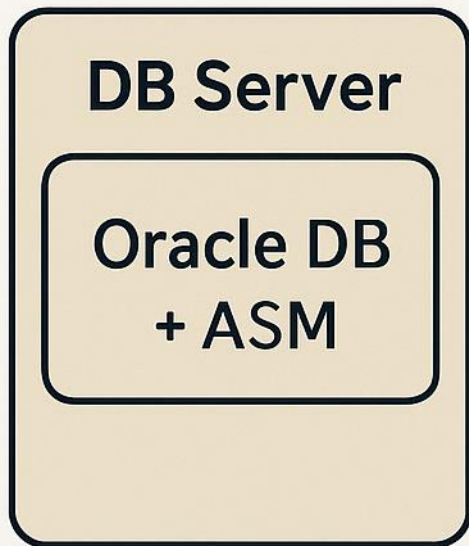
CMD CELLCLI

- Sur RoCE (nouveaux modèles) :
 - Outils standards Ethernet (et Oracle diag tools).
- Logs Exadata et OEM pour monitoring réseau.

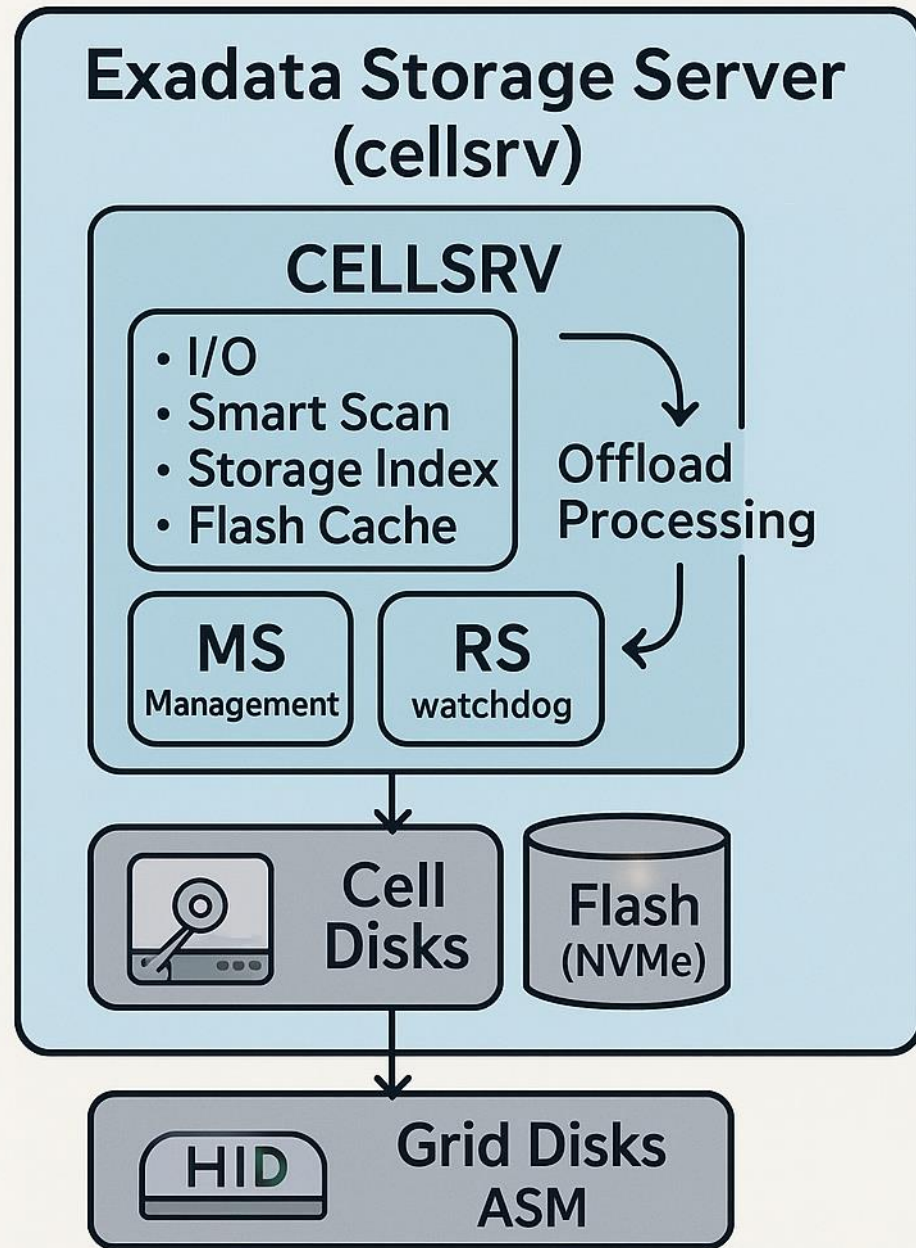
7 Points clés à retenir

- InfiniBand = historique, HPC, très performant.
- RoCE = modernisation → Ethernet + RDMA, plus standard, même niveau de perf.
- Réseau Exadata = **ossature invisible mais critique** : sans lui, pas de Smart Scan ni RAC efficace.
- Exadata ≠ simple cluster Oracle RAC : c'est la **synergie compute + storage + réseau très rapide**.

CELLSRV



iDB (RDS
over InfiniBand/
RoCE



1 Définition

- Exadata Storage Server Software (cellsrv) = logiciel propriétaire Oracle installé sur chaque Storage Cell (serveur de stockage).
 - C'est lui qui transforme un "simple serveur de disques" en un **moteur de traitement intelligent pour Oracle Database**.
 - Il exécute les optimisations Exadata : **Smart Scan, Storage Index, Flash Cache, IORM, HCC**.
-

2 Architecture d'un Storage Server

Chaque cell contient :

- Disques : HDD haute capacité + SSD/NVMe Flash.
- Cell Server Software (cellsrv) → cœur Exadata.
- MS (Management Server) → monitoring et gestion (via CellCLI/ExaCLI).
- RS (Restart Server) → superviseur, redémarre automatiquement les services cellsrv/MS en cas de crash.

👉 Trois processus essentiels : **cellsrv, ms, rs**.

3 Fonctionnalités clés (gérées par cellsrv)

1. **Smart Scan** → exécution des prédicats et projections SQL au niveau du stockage.
2. **Storage Index** → élimination de blocs inutiles grâce aux min/max en mémoire.
3. **Hybrid Columnar Compression (HCC)** → compression avancée au niveau du stockage.
4. **Smart Flash Cache** → gestion intelligente du cache SSD/NVMe.
5. **IORM (I/O Resource Manager)** → gestion de la QoS des workloads.
6. **Statistiques détaillées** → vues dynamiques Oracle alimentées par cellsrv.

4 Bénéfices

- Intelligence dans le stockage : pas un simple "block storage".
- Réduction massive d'I/O → gain de performance 10x-100x.
- Consolidation : plusieurs bases peuvent partager la même infra grâce à IORM.
- Résilience : MS + RS assurent haute disponibilité du service cellsrv.

5 Administration

- CellCLI (local) / ExaCLI (distant) pour gérer un Storage Server.
- Monitoring via OEM (Enterprise Manager).
- Vérification de l'état des processus :

```
bash
```

```
ps -ef | grep cellsrv
```

```
ps -ef | grep ms
```

```
ps -ef | grep rs
```

CMD CELLCLI

- Commandes utiles :

```
bash
```

```
CellCLI> LIST METRICCURRENT
```

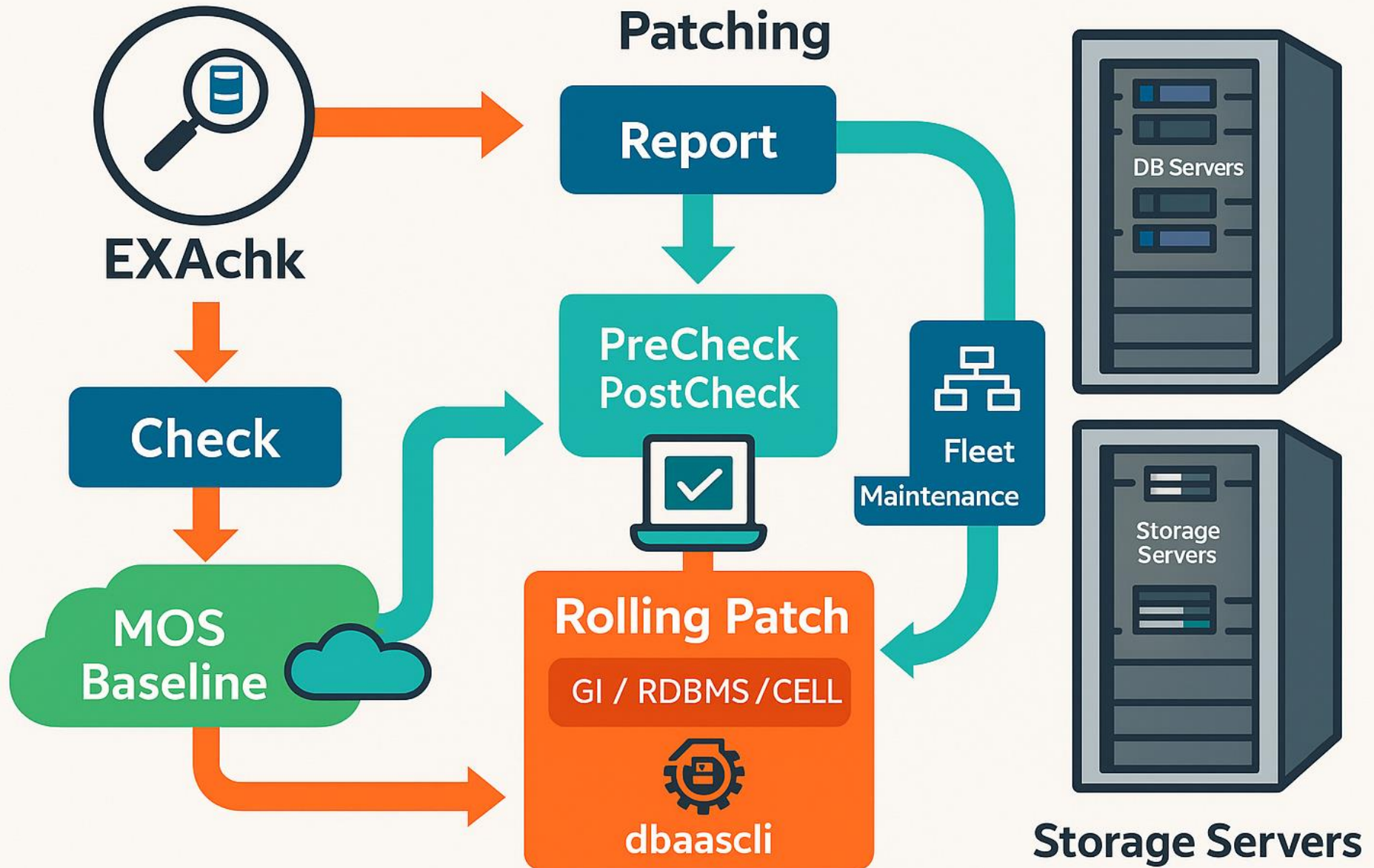
```
CellCLI> LIST CELL DETAIL
```

CMD CELLCLI

6 Points clés à retenir

- `cellsrv` = moteur intelligent de chaque Storage Server.
- Fait tourner toutes les optimisations Exadata (Smart Scan, SI, HCC, Flash Cache, IORM).
- Supervision et résilience assurées par **MS** et **RS**.
- Administration via **CellCLI/ExaCLI** et monitoring OEM.

EXACHK & PATCHING



1 Exachk (Exadata Configuration & Health Check)

Définition

- Exachk = outil Oracle automatisé pour vérifier la conformité, la configuration et la santé d'un système Exadata.
- Inclus dans le package ORAchk (outil générique Oracle), avec des checks spécifiques à Exadata.

Fonctionnalités

- Vérifie la conformité aux **best practices Oracle**.
- Analyse **compute nodes, storage cells, réseau, OS, Grid, DB**.
- Détecte les **écarts de configuration** et problèmes de sécurité.
- Produit un **rapport HTML/PDF** avec un score global et recommandations.

Commandes

bash

```
./exachk -a          # scan complet  
./exachk -profile exadata # checks spécifiques Exadata  
./exachk -html      # sortie en HTML
```

CMD ExaChk

Bénéfices

- Assurance proactive de stabilité.
- Requis par Oracle Support avant un patch ou un upgrade.
- Réduit les incidents liés aux mauvaises configurations.

2 Gestion des patches Exadata

Définition

- Exadata = **stack complet** (OS, Grid, DB, cellsrv, drivers, firmware).
- Patches distribués par Oracle sous forme de **Bundle Patches (BP)** ou **Quarterly Patch**.

Composants patchés

1. Grid Infrastructure & RDBMS → via `opatch` / `opatchauto`.
2. Exadata Storage Server Software (cellsrv) → via `patchmgr`.
3. Compute nodes (OS, drivers) → via `patchmgr`.
4. Firmware (disques, cartes, IB/RoCE) → intégré dans `patchmgr`.

Méthodologie

- **Rolling patching** : applique le patch nœud par nœud → disponibilité continue.
- **Non-rolling** (si nécessaire) : downtime global.
- Oracle recommande toujours :
 - Lancer **exachk** avant patch.
 - Appliquer d'abord **firmware/storage patch** puis **GI/DB patch**.

Commandes principales

```
bash

# Pour DB/Grid
$ opatchauto apply /u01/patch_dir

# Pour cells & compute
$ patchmgr -cells -patch /u01/cell_patch.zip
$ patchmgr -dbnodes -patch /u01/db_patch.zip
```

CMD Patching

3 Outils complémentaires

- **opatch** → utilitaire Oracle classique pour DB/Grid.
 - **opatchauto** → version automatisée pour RAC/ASM.
 - **patchmgr** → spécifique Exadata (cells/compute).
 - **dcli** → exécution distribuée pour vérifier l'état de patches.
-

4 Bonnes pratiques

- Toujours exécuter **Exachk** avant/après patch.
 - Vérifier la compatibilité patch via **MOS (My Oracle Support)**.
 - Prévoir **tests applicatifs** après mise à jour.
 - Documenter la **matrice des patches** (DB, GI, cells, firmware).
-

5 Points clés à retenir

- **Exachk** = "médecin" de l'Exadata → état de santé, conformité, recommandations.
- **Patching** = cycle régulier (trimestriel Oracle) → inclut OS + DB + Storage + firmware.
- Processus automatisés (opatchauto, patchmgr) + rolling patch = haute disponibilité.
- Indispensable pour **sécurité, stabilité, supportabilité**.

ORACLE EXADATA ADMINISTRATION

AGENDA OUTILS ADMINISTRATION

1. FICHE TECHNIQUE – CellCLI

2. FICHE TECHNIQUE – DBMCLI

3. FICHE TECHNIQUE – DCLI

4. FICHE TECHNIQUE ExaCLI

1 Vue d'ensemble

Exadata fournit une suite d'outils en ligne de commande (CLI) pour administrer :

- Les storage servers (cells)
- La Database Machine entière
- L'exécution distribuée de commandes
- L'administration distante

Ces outils permettent de gérer le matériel, le logiciel Exadata, le stockage et les performances.

2 CellCLI (Cell Command Line Interface)

- Portée : un seul Exadata Storage Server.
- Usage : configuration et monitoring local d'un *cell*.
- Fonctions :
 - Gestion du Flash Cache
 - Paramétrage IORM local
 - Surveillance des disques, grilles, métriques
- Exemple :

```
bash
```

```
CMD CELLCLI
```

```
CellCLI> LIST FLASHCACHEDETAILED
```

```
CellCLI> LIST METRICCURRENT WHERE objectType = 'CDisk'
```

3 DBMCLI (Database Machine Command Line Interface)

- Portée : toute la Database Machine (ensemble compute + storage).
- Usage : administration globale (IORM global, DB plans, etc.).
- Fonctions :
 - Création et gestion des IORM plans
 - Suivi de la consommation par base de données
- Exemple :

```
bash
```

CMD DBMCLI

```
DBMCLI> LIST DBPLAN
```

```
DBMCLI> ALTER IORMPLAN dbplan = 'oltp=70 dw=30'
```

4 DCLI (Distributed Command Line Interface)

- Portée : tous les nœuds Exadata (compute & storage) en parallèle.
- Usage : exécuter une commande sur tous les serveurs à la fois.
- Fonctions :
 - Automatisation (ex : patching, mise à jour configs)
 - Vérification globale (état des services, logs)
- Exemple :

```
bash
```

```
CMD DCLI
```

```
dcli -g cell_group -l root "cellcli -e 'list cell attributes name, status'"
```

```
dcli -g dbs_group -l oracle "ps -ef | grep pmon"
```

5 ExaCLI (Exadata Command Line Interface – distant)

- Portée : accès à distance aux Exadata Storage Servers.
- Usage : permet à un DBA d'administrer un cell sans se connecter en SSH.
- Fonctions :
 - Monitoring Flash Cache, IORM, disques
 - Commandes similaires à CellCLI mais exécutées à distance
- Exemple :

```
bash
```

CMD ExaCLI

```
exaccli -d storage01 -u admin -p **** "list metriccurrent"
```

6 Autres outils utiles

- Exachk → vérification automatique de la conformité et de la santé Exadata.
 - OEM (Oracle Enterprise Manager) → interface graphique globale.
 - TFA (Trace File Analyzer) → diagnostic et support Oracle.
-

7 Points clés à retenir

- CellCLI = local à un storage server.
- DBMCLI = global à la Database Machine.
- DCLI = commandes distribuées sur plusieurs nœuds.
- ExaCLI = accès distant sécurisé aux cells.
- Outils complémentaires : Exachk, OEM, TFA.



EXADATA

ENGINEERED FOR THE FUTURE

ORACLE

PRESENTATION DES FLUX DE DONNEES

ORACLE
CloudWorld

Oracle Exadata: Architecture and Internals Technical Deep Dive

October 7-9, 2024
Ideo-Lab Forum
Speaker Guillaume Oneill
Boston, MA



Exadata Vision

Ideal Database Hardware

- intelligent storage optimized for database

Automated Management

- optimized end-to-end

Database Aware System Software

- unique algorithms improve all modern db workloads

Available Everywhere

- on-prem, multi-cloud, hybrid

AUTONOMOUS DATABASE

X



Leading Organizations in Every Industry
rely on Exadata

10/10

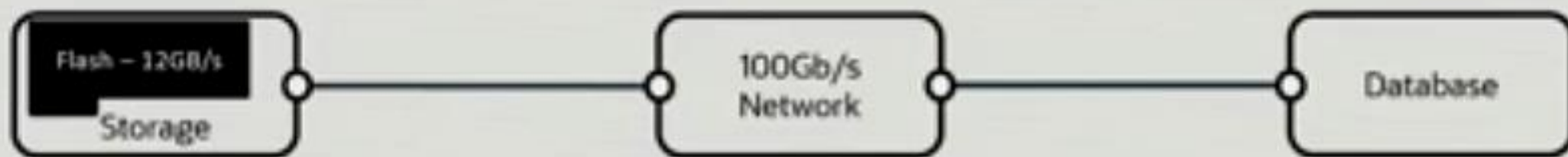
automotive, telecommunications

9/10

energy, banking, healthcare, technology,
food&drug, retail

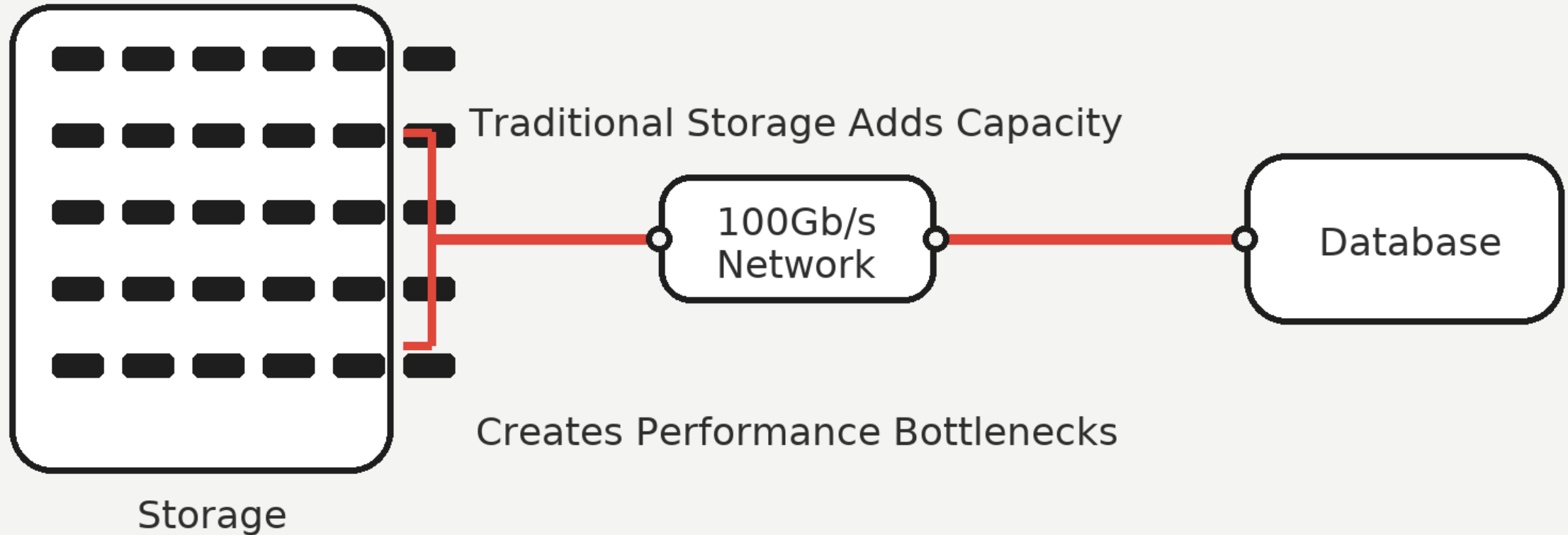


Storage Networking Background

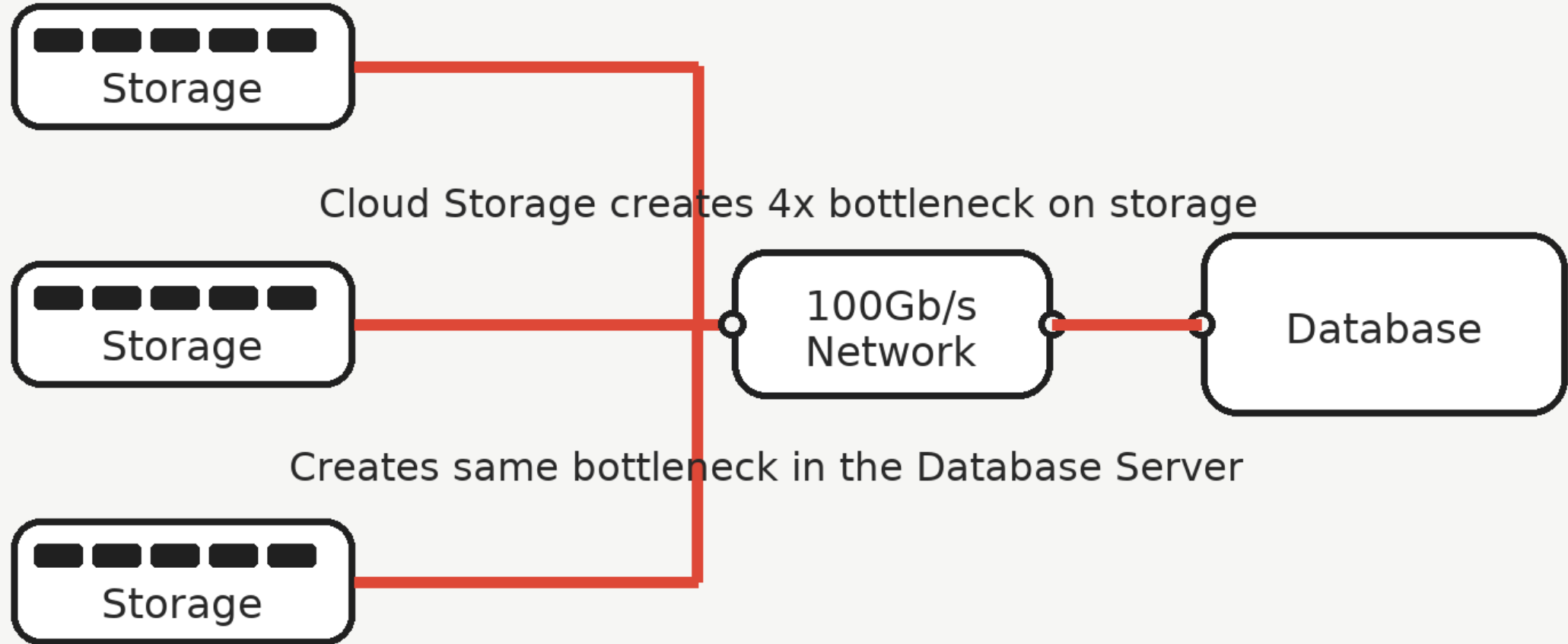


One flash card can saturate a 100Gb Network

Traditional Monolithic Storage Bottleneck



Cloud Storage Bottleneck



Traditional Scan

Find all Customers who ordered an item over \$1000

Customer	Price	Order ID
Alice	2000	101
Bob	800	102
John	200	103
Jane	125	104
Frank	200	105

Storage

all rows moved to database from storage

80% of rows not used by the Database

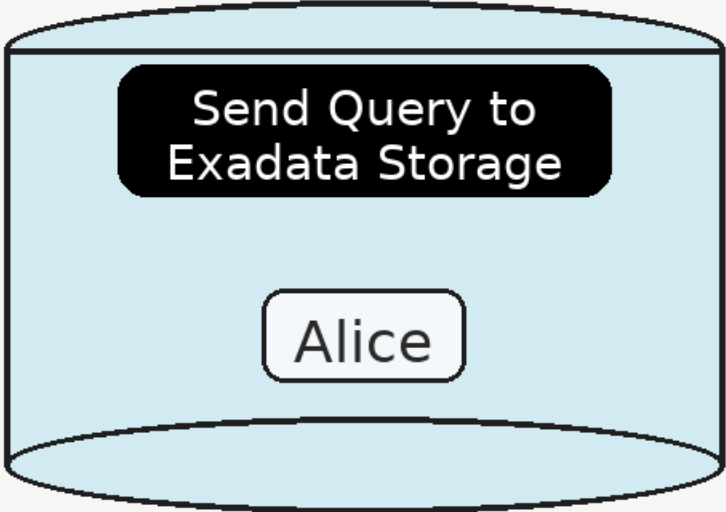
Customer	Price	Order ID	
Alice	2000	101	✓
Bob	800	102	✗
John	200	103	✗
Jane	125	104	✗
Frank	200	105	✗

Database

Smart Scan

Find all Customers who ordered an item over \$1000

Customer	Price	Order ID
Alice	2000	101
Bob	800	102
John	200	103
Jane	125	104
Frank	200	105



Database

Queries exploit full flash throughput

Exadata Storage

Smart Scan Filtering and Projection in AWR Report

- Instance Activity Stats
 - Ordered by statistics name

Statistic	Total	per Second	per Trans
cell physical IO bytes eligible for predicate offload	874,185,441,280	6,022,283.91	145,697,573.46
cell physical IO bytes eligible for smart IOs	1,194,825,646,080	96,525,647.79	199,137,607,680
cell physical IO bytes processed for IM capacity	867,698,982,912	70,098,866.37	144,616,497,152
cell physical IO bytes saved by columnar cache	447,248,203,776	36,131,566.75	74,541,367,296
cell physical IO interconnect bytes returned by smart scan	7,388,109,480	596,858.68	1,231,351,580.00

Filtering and Projection in SQL* Monitor

SQL Monitor Active Report

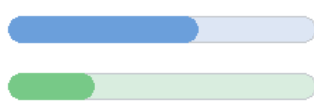
Real-time SQL Monitoring: 5r375t0h2rqw6 Status

Overview

General

```
select * from ssb.lineorder where lo_orderkey = 1174020208
```

Time & Wait



Plan Statistics Parallel SQL Text Activity Metrics

SELECT STATEMENT
RESULT CACHE
PX COORDINATOR
PX SEND QC (RANDOM)




TABLE ACCESS STORAGE FULL

Filter Predicates

'LO_ORDERKEY' = 1174020208

Other Information

Metadata bytes	4,896
Eligible bytes	173.352 G
Filtered bytes	49.254 M
SI saved bytes	4,661.609 M
Flash cache bytes	168.871 G

Close