Détection d'intrusions en environnement haute performance

Clusters HPC

Fabrice Gadaud
(fabrice.gadaud@cea.fr)

Sommaire

- **Caractéristiques d'un Cluster HPC**

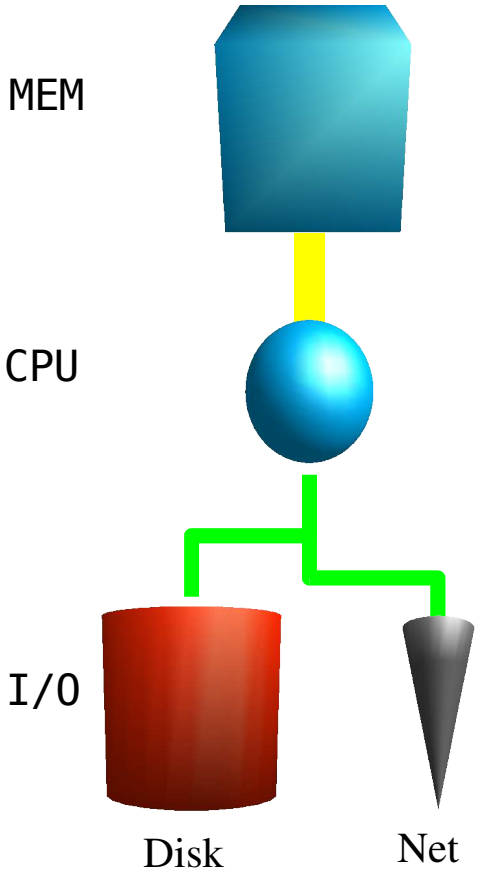
  “**H**igh **P**erformance **C**omputing”

- **Problèmes de sécurité**
- **Limitations**
- **Exemples de détections d'intrusions adaptées**
- **Avantages de l'environnement**
- **Orientations**

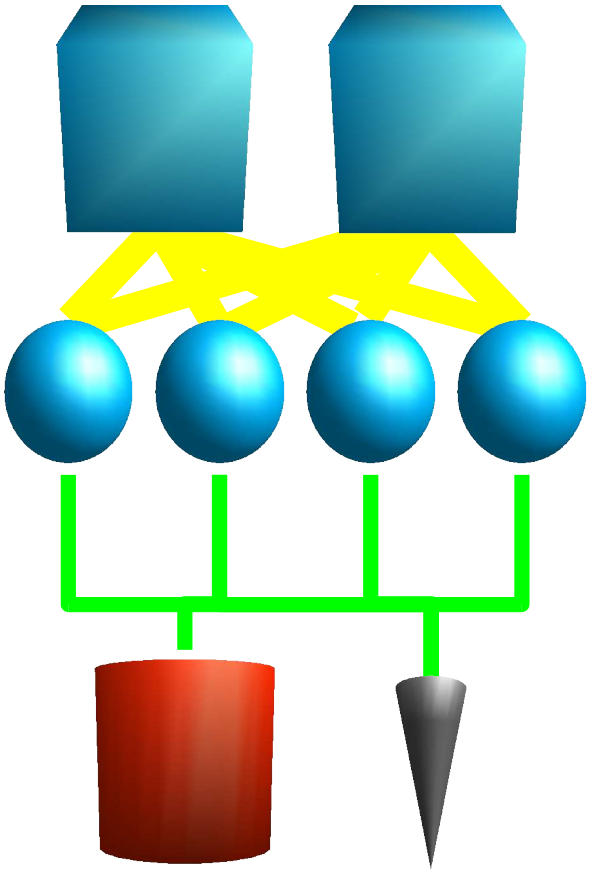
- **Centre de calcul Européen**
  - acteur majeur de la simulation numérique
  - promoteur des architectures HPC
    - ☞partenariats avec EDF, SNECMA, ONERA, ...
  - participations à des groupes de travail
    - ☞académiques
    - ☞industriels
  - compétences de renommée internationale
    - ☞codes de calcul
    - ☞maîtrise de la technologie
- **Tera 10:**
  - 50+ Tflops, plus gros calculateur Européen
  - parmi les 5 machines de calcul les plus puissantes au monde
  - participe à la force de dissuasion nucléaire française

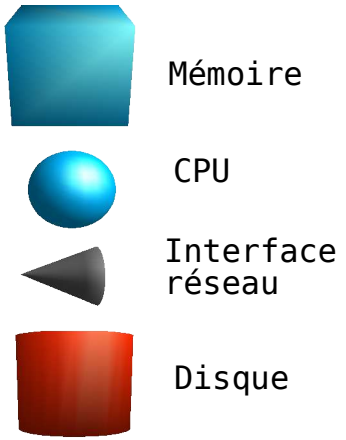
Architectures monolithiques

- **Classique uni-processeur**

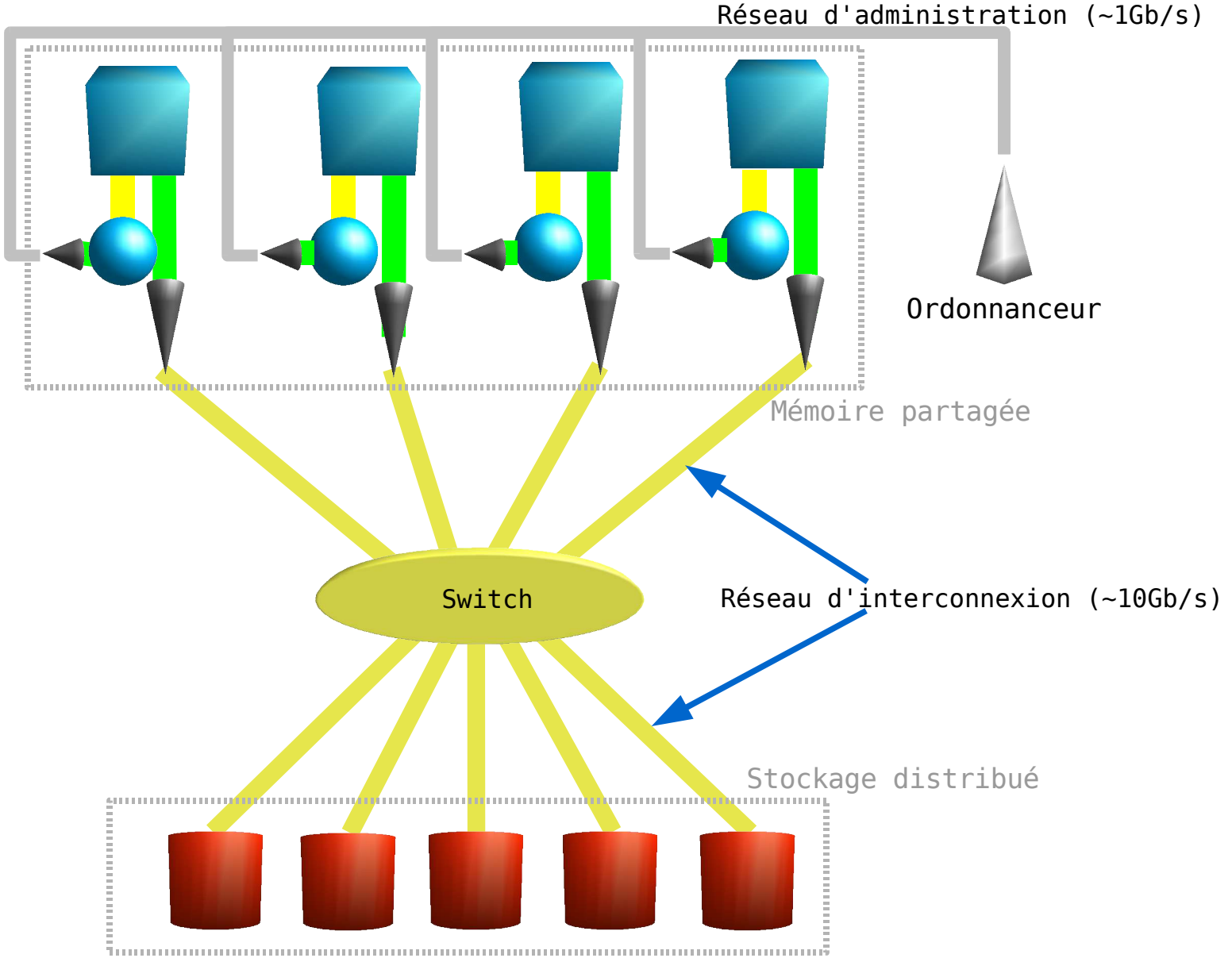


- **SMP/NUMA**

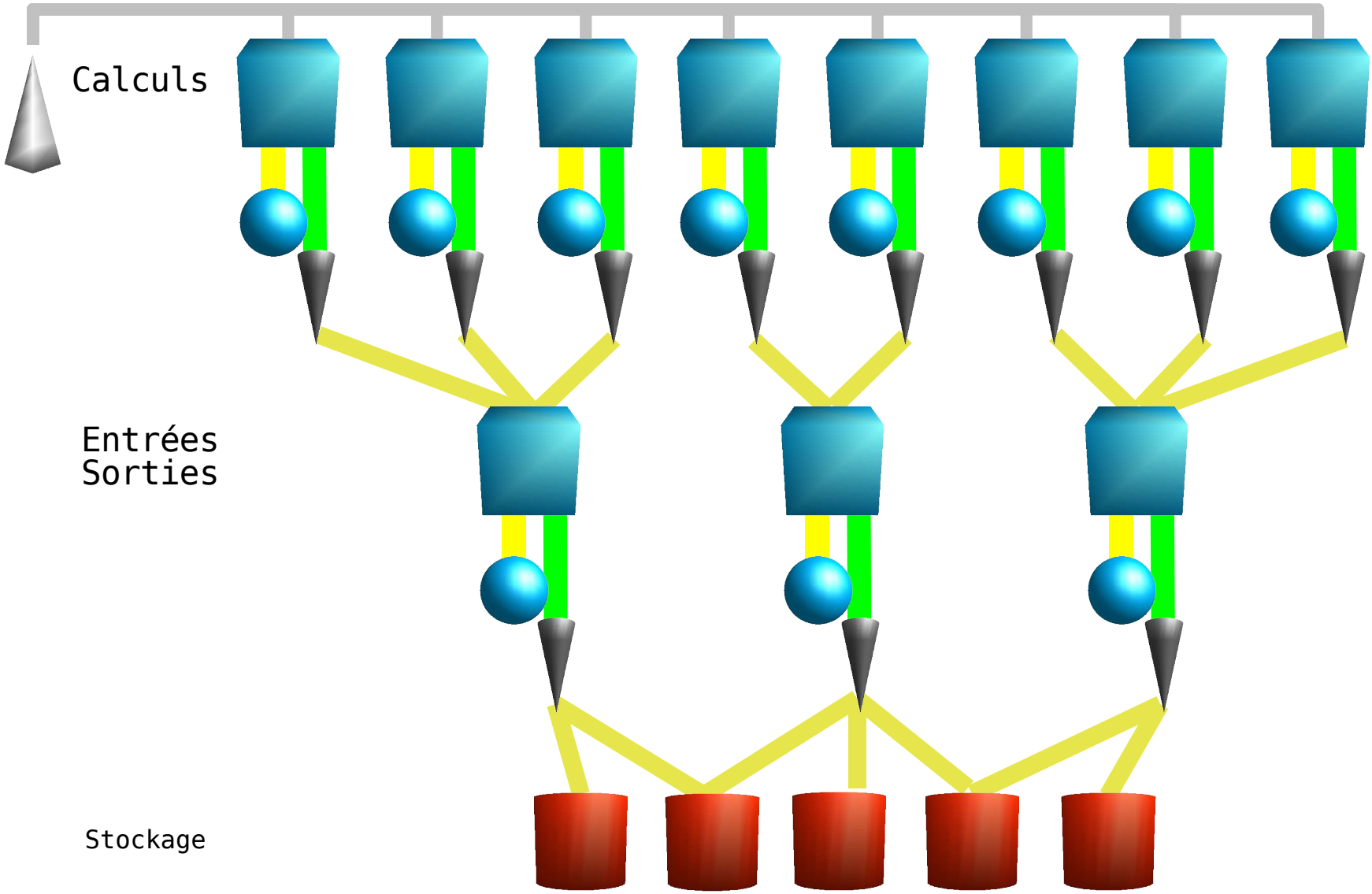




Cluster HPC

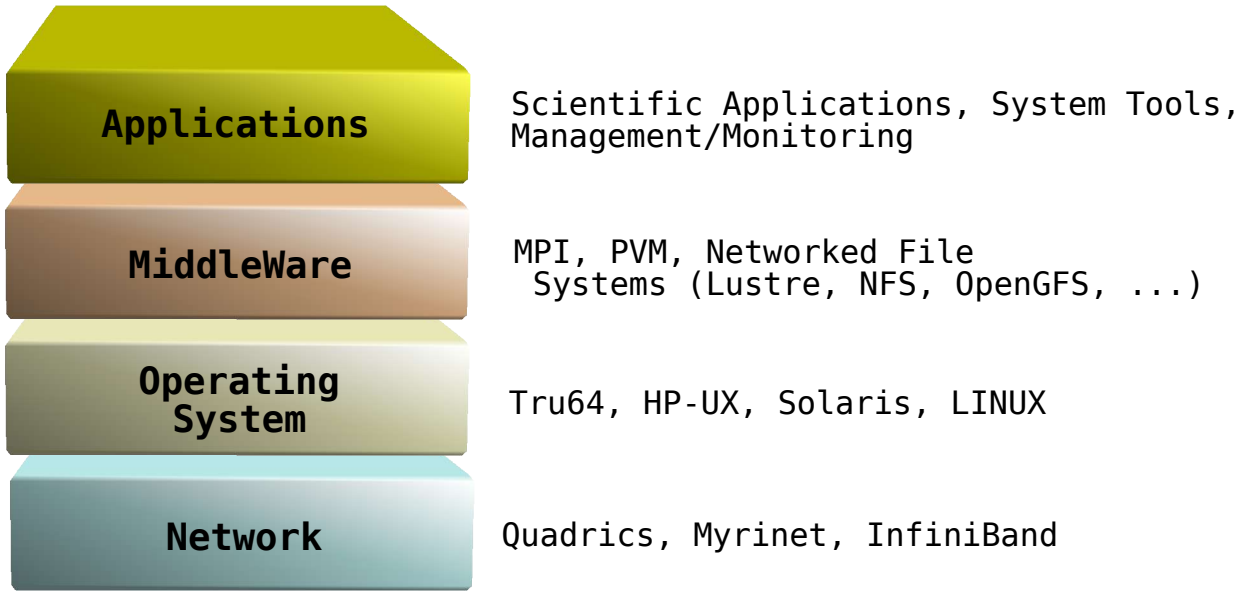

Réseau d'administration (~1Gb/s)
Ordonnanceur
Mémoire partagée
Switch
Réseau d'interconnexion (~10Gb/s)
Stockage distribué

Rôles



Couches

- **Spécificités**
  - couche applicative « middleware »
  - réseau de passage de messages



Problèmes de sécurité

- **système sensible**
  - concentration très importante de ressources (CPU + NET)
  - exécution de tâches critiques
  - homogénéité
  - cycles longs de mise à jour
- **problématique des systèmes partagés**
  - confidentialité
  - intégrité
  - disponibilité
- **concernant les IDS:**
  - intrusions externes menaçant le système
    - ☞attaques « classiques » à l'échelle du cluster, destruction de données, atteinte à la disponibilité, ...
  - intrusions internes, plus furtives
    - ☞collecte d'informations, exploitation de failles, falsification, ...

Limitations

- **Impacts sur les ressources à considérer**
  (caractéristiques principales conditionnant l'efficacité d'un cluster)
  - CPU (vol de cycles, changement de contextes, ...)
  - bande passante CPU <-> MEM
  - bande passante Noeud <-> Noeud
  - latence (temps moyen d'envoi d'un message)
  - disponibilité

- **Adéquation des solutions existantes**
  - Réseaux (d'une dizaine à quelques milliers de noeuds)
    - ☛WAN: débits et volumes
    - ☛LAN: latences (quelques micro secondes) et topologie
  - Systèmes de sécurité au niveau noyau
    - ☛perturbent le noeud (effets micro et macro-scopiques)
    - ☛ne passent pas à l'échelle

Limitations (architecture)

- **Modèle simple de description des performances: LogP**

Variables du systèmes:

P: nombre de processeurs

o: surcharge processeur, due à une communication

g: interval minimal entre 2 envois de messages

L: latence de communication

Variables définies sur un temps τ:

C: temps de calcul pur

a: nombre de messages “broadcast” et “reduce”

M: temps total d'accès à la mémoire

s: nombre de messages “point à point”

$$avec\ s < \frac{tau.L}{g}\ et\ a < \frac{L}{[(P-1).g]}$$

α : impact moyen

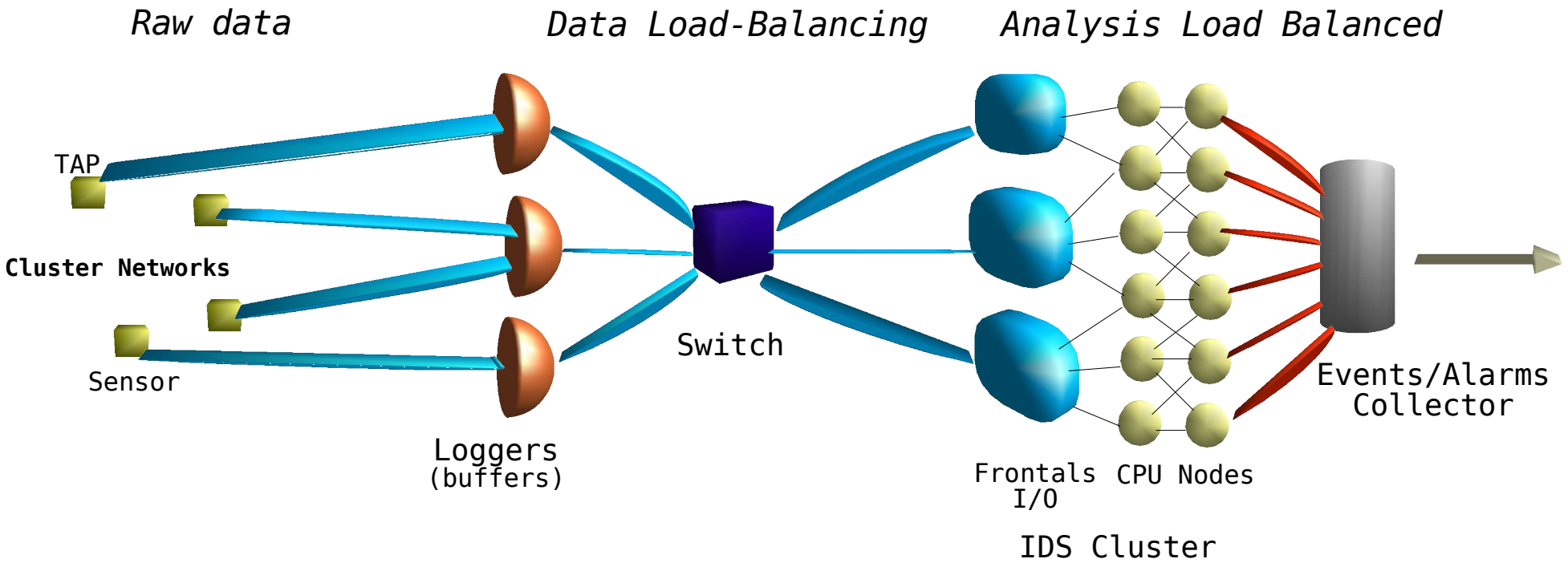
$$\tau' = \tau + surcharge$$

$$\tau' = C + (a+s).o + (1+\alpha).[M + 2L.(a.(P-1)+s)]$$

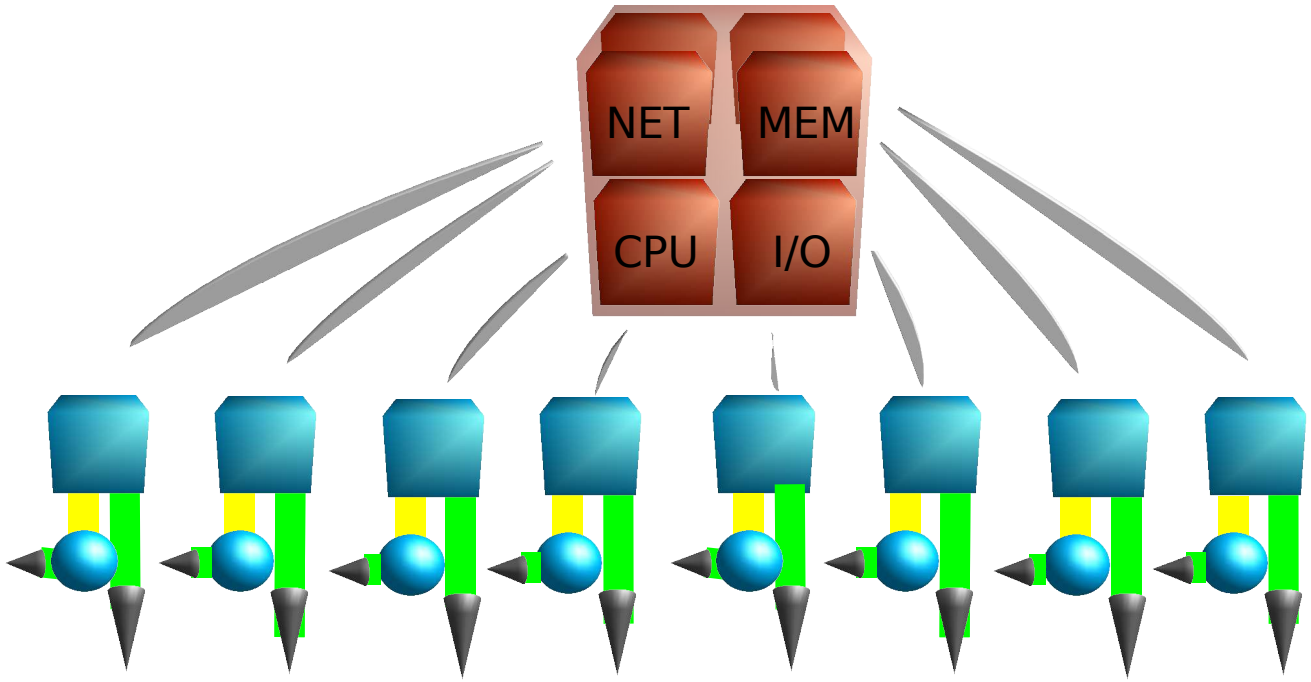
- **l'accès à la mémoire est dégradé**
- **l'impact augmente avec le nombre de processeurs**

Infrastructure adaptée aux réseaux

- **Architecture de traitement des données HPC**
  - capacités d'enregistrement / analyse élevées
  - «scalable» et utilisation d'équipements similaires à ceux du cluster
  - éclatement du processus de détection classique
  - ré-utilisation de solutions pour architecture mono-processeur



- **Vérification d'une politique de sécurité**
  - segmentation suivant de grands « pôles »
    - ☞répartition de la charge de sécurisation

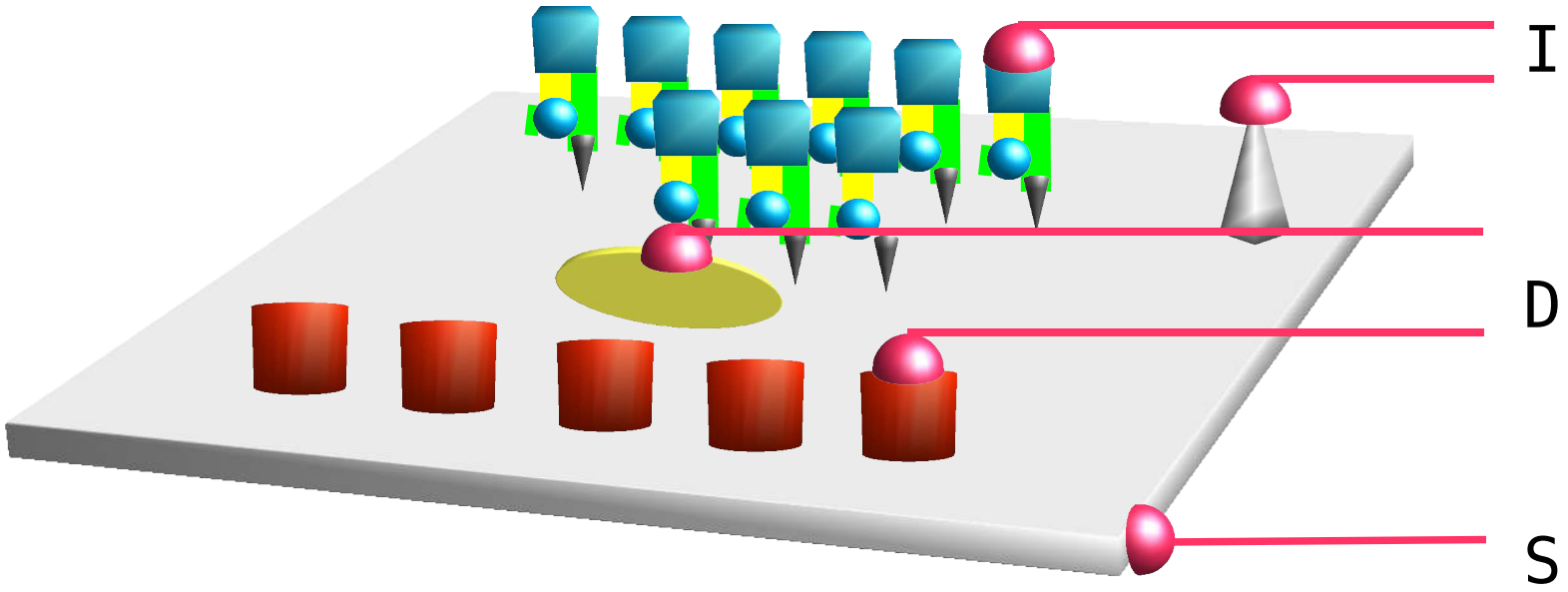


  - points particuliers
    - ☞éviter les vérifications sur des appels très fréquents et sans intérêt

Exploitation de l'architecture

- **Avantages:**
  - le système repose sur les réseaux, les programmes sont couplés
  - l'environnement est restreint (applications, protocoles, topologie, ...)
  - environnement statistiquement intéressant: entités identiques effectuant des opérations similaires
  - participation à l'effort de monitoring, statistiques, qualité de service (données relativement similaires)
- **Instrumentation adaptée à l'architecture**
  - rôles: récupérer l'information là où elle coûte le moins
  - ordonnanceur: coupler les informations provenant du manager de jobs avec les processus s'exécutant sur chaque noeud
  - système de fichiers distribué: requêtes sur les noeuds I/O et index
  - middleware: données échangées de « noeud à noeud » et exécution de sous-programmes par l'utilisateur (MPI-2)
  - OS: alarmes ponctuelles sur noeuds (accès à des fichiers locaux particuliers d'un noeud local, segmentation faults, ...)

Instrumentation générale d'un cluster



Conclusion

- **Les solutions actuelles ne passent pas à l'échelle**
- **Environnement restreint**
  - nombre d'applications limité
  - quelques protocoles réseau utilisés
  - caractéristiques statistiques intéressantes
- **Détection d'intrusions HPC (relativement à celle dite « classique »)**
  - plus de points de mesure
  - prise en compte de l'impact sur les performances
  - exploitation de l'environnement, régi par des règles implicites
  - recherche d'incohérences entre les informations issues des réseaux, de l'environnement d'exécution des noeuds, du système de fichiers, la couche applicative, et l'ordonnanceur
- **Réutilisation de certains logiciels disponibles, non dédiés**
- **Développement de systèmes de monitoring adaptés à l'architecture**
- **Développement de méthodes dédiées au système distribué**

Merci
Questions ?