

---

# numatop

## Outil d'analyse pour l'accès mémoire

Les systèmes modernes utilisent un concept NUMA (Non-Uniform Memory Access) pour le multi-cpu. Dans les systèmes NUMA et les processeurs organisés de telle manière que certaines parties de la mémoire sont proches d'un processeur donné, alors que d'autres parties sont plus éloignées. Un processeur peut accéder à la mémoire proche de lui plus rapidement que la mémoire éloignée.

numatop est un outil d'observation pour la caractérisation de localité mémoire en temps-réel et d'analyse de processus et threads tournant dans un système NUMA. Il aide l'utilisateur à caractériser le fonctionnement NUMA des processus et threads et d'identifier où les goulots d'étranglement se produisent en terme de performances. L'outil utilise la technologie de compteur de performance Intel et associe les données de performances avec les informations temps réel de Linux pour fournir une analyse temps-réel dans les systèmes de production. L'outil peut être utilisé pour :

- caractériser la localité de tous les processus et threads pour identifier ceux ayant la localité la plus pauvre dans le système.
- Identifier les aires de mémoire "hot", indiquer la latence mémoire moyenne, et fournir l'emplacement où la mémoire accédée est allouée. Une aire de mémoire "hot" est où les accès process/thread sont les plus fréquents. numatop a une métrique appelée "ACCESS%" qui spécifie le pourcentage d'accès mémoire attribuable à chaque aire mémoire. numatop n'enregistre que les accès mémoire qui ont des latences supérieures à ceux prédéfinis par le seuil (128 cycles cpu)
- Fournir les call-chain dans le code process/thread qui accède à une aire mémoire hot.
- Fournir call-chain quand le process/thread génère certains compteurs d'évènement (RMA/LMA/IR/CYCLE). Les call-chain aident à localiser le code source qui génèrent les évènements.

**RMA** Remote Memory Access

**LMA** Local Memory Access

**IR** Instruction Retired

**CYCLE** cycles CPU

- Fournir des statistiques par nœud pour l'utilisation mémoire et CPU. Un nœud est une région de mémoire dans laquelle tout octet a la même distance de chaque CPU.
- Affiche, en utilisant une interface user-friendly, la liste des process/thread triés par certaines métriques (par défaut, trié par l'utilisation CPU). Les utilisateurs peuvent également utiliser les raccourcis pour retrier la sortie par ces métriques : RMA, LMA, RMA/LMA, CPI et CPU%.

**RMA/LMA** ratio de RMA/LMA

**CPI** cycle CPU par instruction

**CPU%** utilisation CPU

numatop est un outil GUI qui traque et analyse périodiquement l'activité NUMA des processus et threads et affiche des métriques utiles.

Ci-dessous est une description détaillée des diverses affichages et les éléments de données qu'elles affichent.

## WIN1 - Supervision des processus et threads

Affiche la caractérisation de localité de tous les processus. C'est la première fenêtre affichée à démarrage. Cette fenêtre affiche une liste de processus, trié par % de CPU. Généralement, le processus intensifs en mémoire est également intensif en CPU. les touches 1,2,3,4, et 5 permettent de trier la sortie par RMA, LMA, RMA/LMA, CPI, et CPU%, respectivement.

**RAM(K)** nombre de RMA (l'unité est 1000)

---

**LMA(K)** nombre de LMA (l'unité est 1000)

**RMA/LMA** ratio de RMA/LMA

**CPI** cycles CPU par instruction

**CPU%** utilisation CPU du système

**Q** quitter l'application

**H** Rafraîchir la fenêtre

**R** Rafraîchis les données

**I** Passer à WIN2

**N** Passer à WIN11

**1** Trier par RMA

**2** Trier par LMA

**3** Trier par ratio RMA/LMA

**4** Trier par CPI

**5** Trier par CPU%

## WIN2 - Supervision des processus et threads (normalisée)

Affiche la caractérisation de localité normalisée de tous les processus

**RPI(K)** RMA normalisé par 1000 instructions :  $RPI(K) = RMA / (IR / 1000)$

**LPI(K)** LMA normalisé par 1000 instructions :  $LPI(K) = LMA / (IR / 1000)$

**Les autres métrique sont les mêmes.**

**Q** quitter l'application

**H** Passer à WIN1

**B** Retour à la précédent fenêtre

**R** Rafraîchis les données

**N** Passer à WIN11

**1** Trier par RPI

**2** Trier par LPI

**3** Trier par ratio RMA/LMA

**4** Trier par CPI

**5** Trier par CPU%

## WIN3 - Supervision de processus

Affiche la caractérisation de localité avec l'affinité de nœud d'un processus spécifié

**NODE** L'id de nœud

**CPU%** Utilisation CPU par nœud

**Les autres métriques restent les mêmes**

**Q** quitter l'application

**H** Passer à WIN1

**B** Retour à la précédent fenêtre

**R** Rafraîchis les données

- 
- N** Passer à WIN11
  - L** Affiche les informations de latence
  - C** Affiche le call-chain

## WIN4 - Supervision de tous les threads

Affiche la caractérisation de localité de tous les threads dans un processus spécifié.

## WIN6 - Supervision des aires mémoire

Affiche l'utilisation d'aire mémoire avec la latence d'accès associée d'un process/thread

- ADDR** Adresse de début de l'aire mémoire
- SIZE** Taille de l'aire mémoire
- ACCESS%** Pourcentage d'accès mémoire fait dans cette aire mémoire
- LAT(ns)** La latence moyenne des accès mémoire
- DESC** Description de l'aire mémoire
- Q** quitter l'application
- H** Passer à WIN1
- B** Retour à la précédent fenêtre
- R** Rafraîchis les données
- A** Affiche la distribution de nœuds d'accès mémoire
- C** Affiche le call-chain quand les process/thread accèdent à la mémoire.

## WIN7 - vue générale de distribution de nœuds d'accès mémoire

Affiche le pourcentage d'accès mémoire venant du process/thread de chaque nœud

- Q** quitter l'application
- H** Passer à WIN1
- B** Retour à la précédent fenêtre
- R** Rafraîchis les données

## WIN8 - sépare l'aire mémoire dans la mémoire physique dans le nœud

Sépare l'aire mémoire dans le mappage physique dans le nœud avec la latence d'accès associée d'un processus/thread.

## WIN9 - Call-chain quand les process/threads génèrent l'évènement (RMA/LMA/CYCLE/IR)

---

Détermire les call-chains du code qui génère RMA/LMA/CYCLE/IR

- Q** quitter l'application
- H** Passer à WIN1
- B** Retour à la précédent fenêtre
- R** Rafraîchis les données
- 1** Localise le call-chain quand le processus/thread génère RMA
- 2** Localise le call-chain quand le processus/thread génère LMA
- 3** Localise le call-chain quand le processus/thread génère CYCLE
- 4** Localise le call-chain quand le processus/thread génère IR

## WIN10 - Call-chain quand le process/thread accède à l'aire mémoire

Détermine les call-chains du code qui référence cette aire mémoire. La latence doit être supérieure au seuil de latence prédéfinis (128 cycles CPU)

## WIN11 - Vue générale des nœuds

Affiche les statistiques de base par nœud pour ce système

- MEM.ALL** RAM utilisable totale (physique moins quelques octets réservés et le code du kernel)
- MEM.FREE** Somme LowFree + HighFree
- CPU%** Utilisation CPU par nœud

## WIN12 - information de nœud

Affiche l'utilisation et CPU pour le nœud sélectionné

- CPU** Tableau de cpu logiques appartenant à ce nœud
- CPU%** Utilisation de CPU par nœud
- MEM active** quantité de mémoire utilisé la plus récemment et n'est pas réclamé
- MEM inactive** Quantité de mémoire qui n'a pas été utilisé depuis un certain temps et est éligible au swap
- Dirty** Quantité de mémoire attendant d'être écrite sur disque
- Writeback** Quantité de mémoire activement écrite sur le disque
- Mapped** Toutes les pages mappée dans un processus

## OPTIONS

- s sampling\_precision** balance entre précision et charge(normal), précision(high), faible charge(low). défaut : normal.
- l log\_level** Spécifie le niveau de log dans le fichier de log. valeurs valides : 2

---

**-f log\_file** spécifie le fichier de log  
**-d dump\_file** Spécifie le fichier de dump

## Exemples

Lancer numatop à haute précision :

**numatop -s high**

Écrire tous les messages d'alerte dans /tmp/numatop.log :

**numatop -l 2 -o /tmp/numatop.log**

Dump les données de l'écran dans /tmp/dump.log :

**numatop -d /tmp/dump.log**

## Notes

Il est nécessaire d'avoir les privilèges root pour lancer numatop, ou de définir /proc/sys/kernel/perf\_event\_paranoid à 1. numatop supporte les processeurs Intel Xeon.