

Université **IBM i**

19 et 20 novembre 2024

IBM Innovation Studio Paris

S19 – Performances IBM i : « éplucher l'oignon » avec iDoctor

19 novembre 14:45 - 15:45

Ludovic Ménard

IBM France

ludovic_menard@fr.ibm.com



uui2024

#ibmi

#uui2024

IBM

common
FRANCE

Agenda

- ✓ **Vue d'ensemble**
 - ✓ – IBM i Performance Data Collectors
 - ✓ – IBM iDoctor for IBM i (iDoctor)
- ✓ **Analyse avec iDoctor**
 - ✓ – Collection Services
 - ✓ – Job Watcher
 - ✓ – Performance Explorer (PEX)
 - ✓ – Catalogue Db2 et/ou IBM i Services
 - ✓ – Réduire les risques
- ✓ **Ressources supplémentaires**

Outil de performance IBM i

IBM i est un leader de l'industrie en matière de gestion des performances et d'outils.

- Des mesures de performance inégalées
- Services de collecte de données constamment actifs
- Visualisation graphique des données de performance

Instruments de mesure de performance

L'avantage d'IBM i

- IBM développe le système d'exploitation, du début à la fin.
 - ✓ Possibilité d'instrumenter le logiciel pour maintenir les mesures de performance
 - ✓ Les mesures de performance sont spécifiques aux composants
- IBM développe les collecteurs de données de performance qui récoltent ces mesures de performance.
- IBM i dispose d'une base de données intégrée - Db2
 - ✓ Les données de performance sont stockées automatiquement dans la base de données
- Les outils d'analyse exploitent les données de performance dans les fichiers Db2 à l'aide de SQL et présentent les données de manière significative.
 - ✓ PDI, iDoctor
- Liens étroits entre les équipes d'assistance, de service et de développement
 - ✓ Les clients aussi !

Aperçu du « Wait Accounting »

Etat des Thread/tasks

**Utilisation
de CPU**

Dispatched CPU

Le thread ou la tâche a été assigné à un processeur virtuel afin qu'il puisse commencer l'exécution des instructions.

**Attente
de CPU**

CPU Queuing

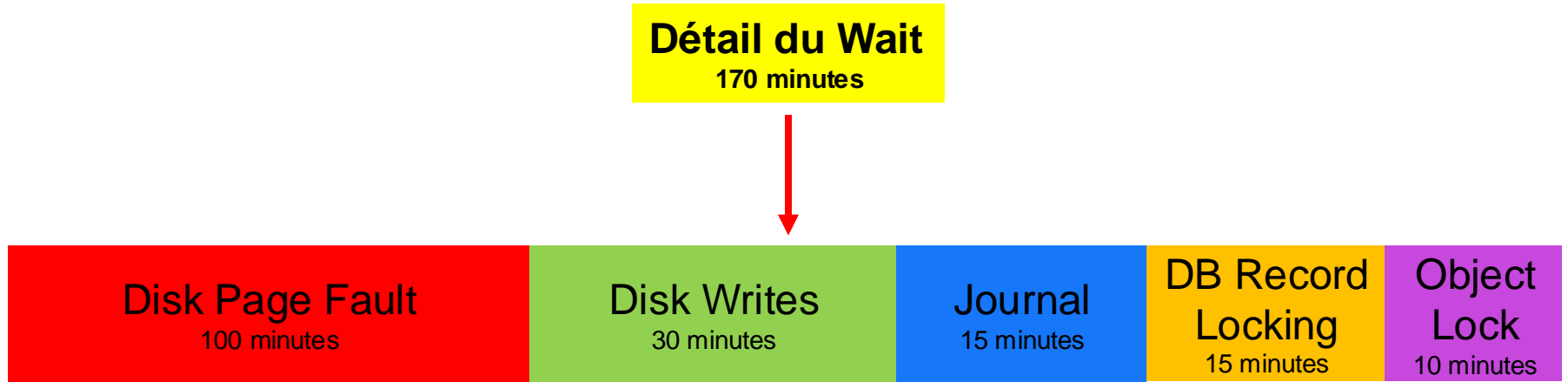
Processeur prêt à l'emploi, mais en attendant qu'il soit disponible

**En attente
d'autre chose**

Waits

Idle
Blocked

Wait Accounting



- Vous pouvez maintenant commencer à poser des questions telles que :
 - ✓ La taille de mes pools est-elle appropriée ?
 - ✓ Les E/S peuvent-elles être effectuées de manière asynchrone ?
 - ✓ Tous les journaux sont-ils nécessaires et configurés de manière optimale ?
 - ✓ Est-ce que je verrouille inutilement des enregistrements ou des objets ?

Catégories des « Wait Accounting »

Collection Services et Job Watcher

1. Time dispatched on a CPU
2. CPU queuing
3. Reserved
4. Other waits
5. Disk page faults
6. Disk non-fault reads
7. Disk space usage contention
8. Disk operation start contention
9. Disk writes
10. Disk other
11. Journaling
12. Semaphore contention
13. Mutex contention
14. Machine level gate serialization
15. Seize contention
16. Database record lock contention
17. Object lock contention
18. Ineligible waits
19. Main storage pool contention
20. Journal save while active (7.2+)
21. unused
22. unused
23. Reserved
24. Socket transmits
25. Socket receives
26. Socket other
27. IFS
28. PASE
29. Data queue receives
30. Idle/waiting for work
31. Synchronization Token contention
32. Abnormal contention

Outils d'analyse des performances - Types

Suivi en temps réel

- Essentiel pour une gestion proactive des performances
- Les données sont basées sur l'échantillonnage
- Rafraîchissement, redémarrage
- Surcharge du système généralement faible
- Pas de données historiques persistantes
- Pas aussi riche en métriques que les outils de collecte
 - WRKACTJOB
 - WRKDSKSTS
 - WRKSYSSTS
 - WRKSYSACT

Collecteur de données de performances

- Outils faisant partie du système d'exploitation de base
- De nombreuses mesures de performance sont disponibles (et continuent d'être ajoutées !)
- Données stockées dans des fichiers Db2
- Contrôle par l'utilisateur du démarrage/arrêt et d'autres paramètres de données
 - Collection Services
 - Job Watcher
 - Disk Watcher
 - Performance Explorer
 - SQL Performance Monitor

Capacity Planning

Capacity planning: [IBM Systems Workload Estimator \(WLE\)](#)

[IBM Navigator Graph History](#)

Collecteurs de données primaires

Chaque collecteur diffère dans le niveau de détail collecté, ainsi que dans le type de données collectées

Caractéristiques du collecteur

Collection Services

- Basé sur l'intervalle
- **Conçu pour collecter les données 24/7**
- Beaucoup de données, mais des données de plus haut niveau
- Système entier
- **Informations sur les temps d'attente**

Disk Watcher

- Données statistiques ou de suivi
- **Informations détaillées sur les opérations d'E/S vers les unités de disque**

Job Watcher

- Basé sur l'intervalle
- Prise en charge de très petits intervalles
- Focus sur les données du travail
 - ✓ **Call stacks**
 - ✓ **Instructions SQL**
- **Informations sur les temps d'attente**
 - ✓ **Wait objects**
 - ✓ **Holder info**

Performance Explorer

- Données statistiques ou de suivi
- **Peut collecter des données très détaillées et spécifiques**
- L'analyse peut être complexe

Tous les collecteurs sont inclus dans le système d'exploitation de base !

Collection Services

- Collection Services est une fonction d'IBM i qui collecte les données de performance au niveau des partitions et des tâches.
- Collecte des données sur de nombreuses ressources au niveau de la partition
 - ✓ CPU, Disque, Pools de mémoire, Jobs, Communications, Java, etc.
- Collecte des données à des intervalles allant de 15 secondes à 1 heure (la valeur par défaut est de 15 minutes, la valeur 5min est recommandée).
 - ✓ Les données sont initialement stockées dans un objet de collecte de gestion (*MGTCOL).
 - ✓ Peut contenir de grandes quantités de données de performance avec une surcharge minimale
 - ✓ Facile à déplacer, à sauvegarder
- Les données de performance sont copiées dans des fichiers Db2 pour analyse.
 - ✓ Les fichiers de base de données et les champs (métriques) sont décrits dans la documentation IBM i

<https://www.ibm.com/docs/en/i/7.5?topic=performance-collection-services-data-files>

Job Watcher

- Job Watcher collecte des données de performance plus détaillées que Collection Services et à des intervalles plus fréquents.
 - ✓ CPU, I/O et Wait Buckets (comme Collection Services)
 - ✓ Piles d'appels (Call Stack)
 - ✓ Instructions SQL
 - ✓ Informations détaillées sur l'attente :
 - Objets en attente, même le numéro d'enregistrement des tables de la base de données
 - Détenteur de l'objet/enregistrement
- Job Watcher ne collecte pas tout ce que Collection Services collecte
- Il ne collecte pas toujours des informations sur chaque thread
 - ✓ Le thread doit utiliser l'unité centrale pendant l'intervalle
 - ✓ Le thread doit exister pendant toute la durée de l'intervalle
- Les données sont écrites dans des fichiers Db2

https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_75/rzahx/rzahxperdatafiles1.htm

Job Watcher versus Collection Services

Comparaison des données de collecte

Collection Services

Recueille des informations sur les ressources au niveau du système que l'on ne trouve pas dans Job Watcher. Les exemples de domaines incluent :

- ✓ Les pools de mémoire
- ✓ Communications
- ✓ Groupes de charge de travail
- ✓ Système physique (cross-partition)
- ✓ Base de données / métriques SQL
- ✓ Création de groupes d'activation
- ✓ Métriques de stockage externe
- ✓ Détails de la configuration matérielle

Job Watcher

Collecte des informations plus détaillées au niveau du travail que les services de collecte, notamment :

- ✓ Les objets en attente
- ✓ Les informations sur les détenteurs (Holder)
- ✓ Les piles d'appels (Call Stack)
- ✓ Les instructions SQL
- ✓ Détails supplémentaires sur l'attente en cours
- ✓ Collecte des données à un intervalle de temps beaucoup plus granulaire, par exemple en secondes plutôt qu'en minutes.

Performance Explorer (PEX)

Le PEX permet d'identifier les causes des problèmes de performance qui ne peuvent être résolus à l'aide de l'un des autres collecteurs de données de performance. Il est généralement utilisé après une première analyse.

Les services de collecte peuvent vous indiquer quel **travail** consomme le plus de ressources CPU, **PEX** peut identifier les **programmes** qui consomment le plus de ressources CPU.

Les services de collecte peuvent vous indiquer qu'un grand nombre de fichiers sont ouverts chaque seconde par certains travaux ; vous pouvez alors utiliser **PEX** pour identifier quels **programmes** ouvrent quels fichiers.

PEX :

- ✓ Collecte des informations détaillées sur une application, un programme ou une ressource spécifique.
- ✓ Permet de collecter des données de traçage ou d'échantillonnage
- ✓ Peut collecter des informations sur chaque événement d'E/S
- ✓ L'analyse des données peut être assez complexe
- ✓ La charge des ressources varie en fonction de la définition utilisée

Performance Explorer (PEX) Définitions et Filtres

A utiliser : (ou utiliser l'assistant iDoctor PEX Analyzer)

Add Performance Explorer Definition (ADDPEXDFN)

- Créez une définition de session qui indique au système les données de performance que vous souhaitez collecter. Vous indiquez ici le type de collecte et un nom pour la définition.
 - ✓ Utiliser le nom de la définition pour commande Start Performance Explorer (STRPEX).

(Optionnel) Ajout d'un filtre PEX (ADDPEXFTR)

- Un filtre de Performance Explorer identifie les données de performance qui doivent être collectées au cours d'une session. Son but est de limiter la quantité de données collectées en spécifiant une valeur de comparaison pour des événements spécifiques.
 - ✓ Programme, objet, pool de mémoire, unité de disque, ASP, etc.

Collecting Performance Explorer (PEX) Data

Démarrer Performance Explorer (STRPEX)



Exécutez la commande, le programme ou la charge de travail que vous souhaitez utiliser pour l'analyse des données



Arrêter Performance Explorer (ENDPEX)



Recommencer si besoin

Outils d'analyse graphique

IBM fournit deux outils puissants pour vous aider à rendre votre analyse plus efficace et plus productive :

- Performance Data Investigator (PDI)
- IBM iDoctor pour IBM i

Les deux solutions prennent en charge l'analyse des données (à des degrés divers) pour les 4 collecteurs :

- Collection Services
- Job Watcher
- Disk Watcher
- Performance Explorer (PEX)

Qu'est-ce que iDoctor?

iDoctor est une suite d'outils dynamiques utilisés pour collecter, étudier et analyser les données de performance sur IBM i.

Surveillez l'état général du système à un niveau élevé « vue d'ensemble » ou descendez dans les détails des performances des tâches, des unités de disque et/ou des programmes

Utilisé par IBM, ainsi que par nos clients et nos consultants, pour aider à visualiser et à résoudre les problèmes de performances.

iDoctor est un outil d'analyse

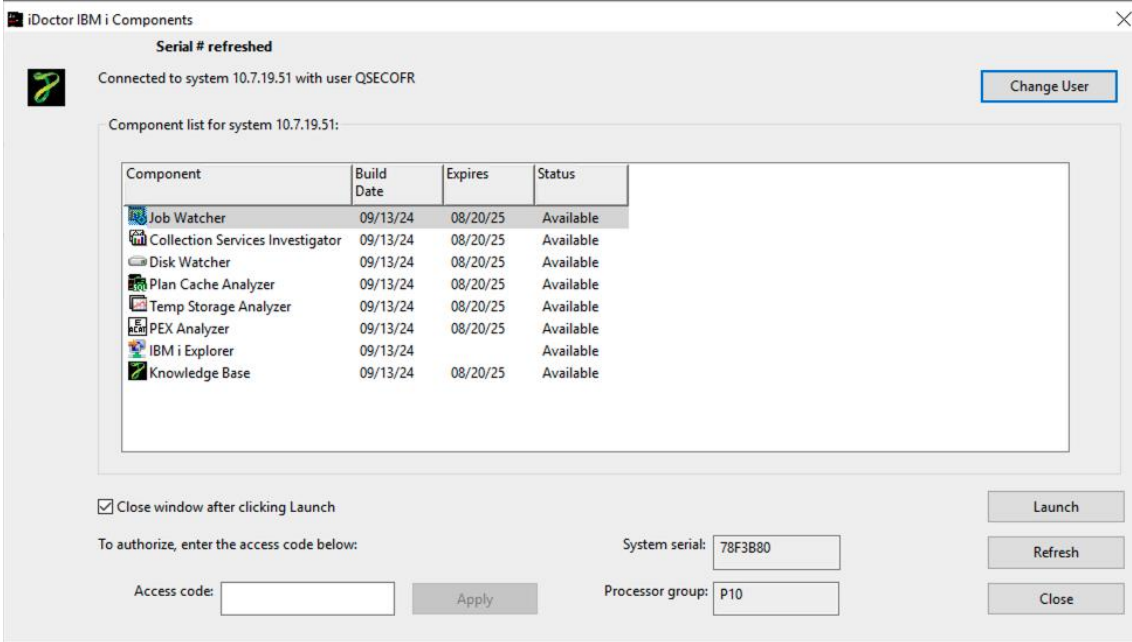
L'interface graphique iDoctor est utilisée pour étudier les données collectées par le système d'exploitation.

Collection Services Investigator	Job Watcher	Plan Cache Analyzer	PEX Analyzer	Disk Watcher
Les services de collecte crée les données dans les les fichiers QAPM*	QSYS/STRJW QSYS/ENDJW	La capture du Plan Cache est collectée par ACS	QSYS/ADDPEXDFN, QSYS/STRPEX, QSYS/ENDPEX	QSYS/STRDW QSYS/ENDDW

Portabilité : Les données peuvent être collectées sur un système ne disposant pas d'une licence iDoctor, puis restaurées et analysées sur un système disposant d'une licence iDoctor.

Composants de iDoctor

iDoctor est une boîte à outils, composée de nombreux outils différents appelés « composants ».

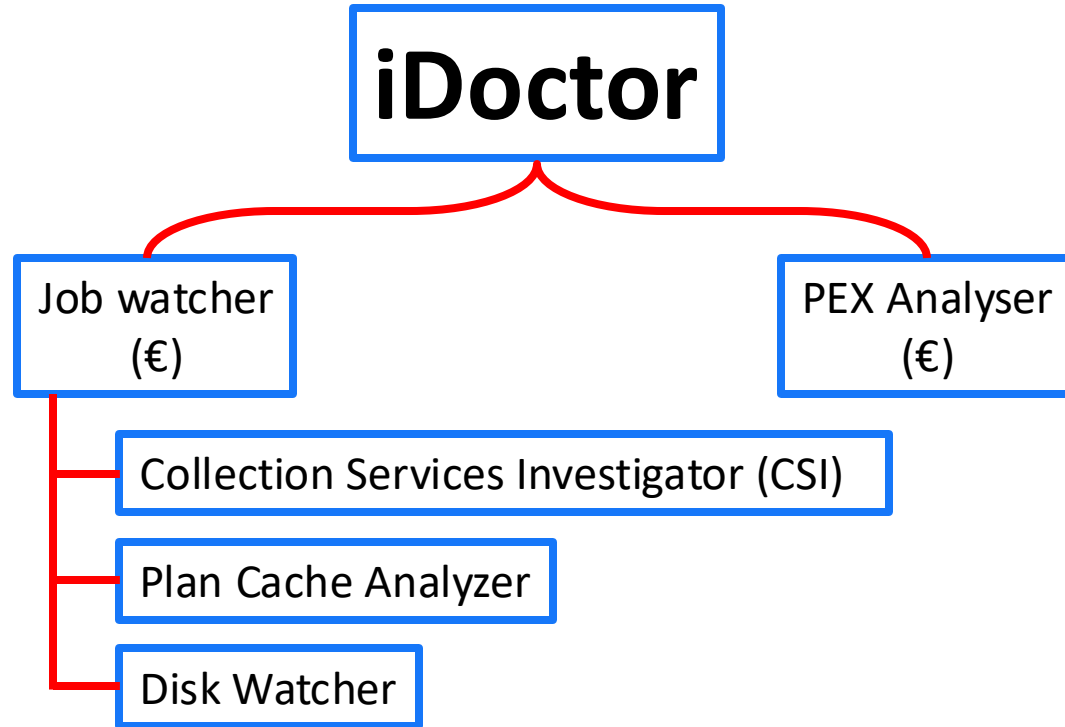


The screenshot shows the 'iDoctor IBM i Components' window. At the top, it indicates 'Serial # refreshed' and 'Connected to system 10.7.19.51 with user QSECOFR'. A 'Change User' button is visible. Below this, a table lists the components for system 10.7.19.51:

Component	Build Date	Expires	Status
Job Watcher	09/13/24	08/20/25	Available
Collection Services Investigator	09/13/24	08/20/25	Available
Disk Watcher	09/13/24	08/20/25	Available
Plan Cache Analyzer	09/13/24	08/20/25	Available
Temp Storage Analyzer	09/13/24	08/20/25	Available
PEX Analyzer	09/13/24	08/20/25	Available
IBM i Explorer	09/13/24	08/20/25	Available
Knowledge Base	09/13/24	08/20/25	Available

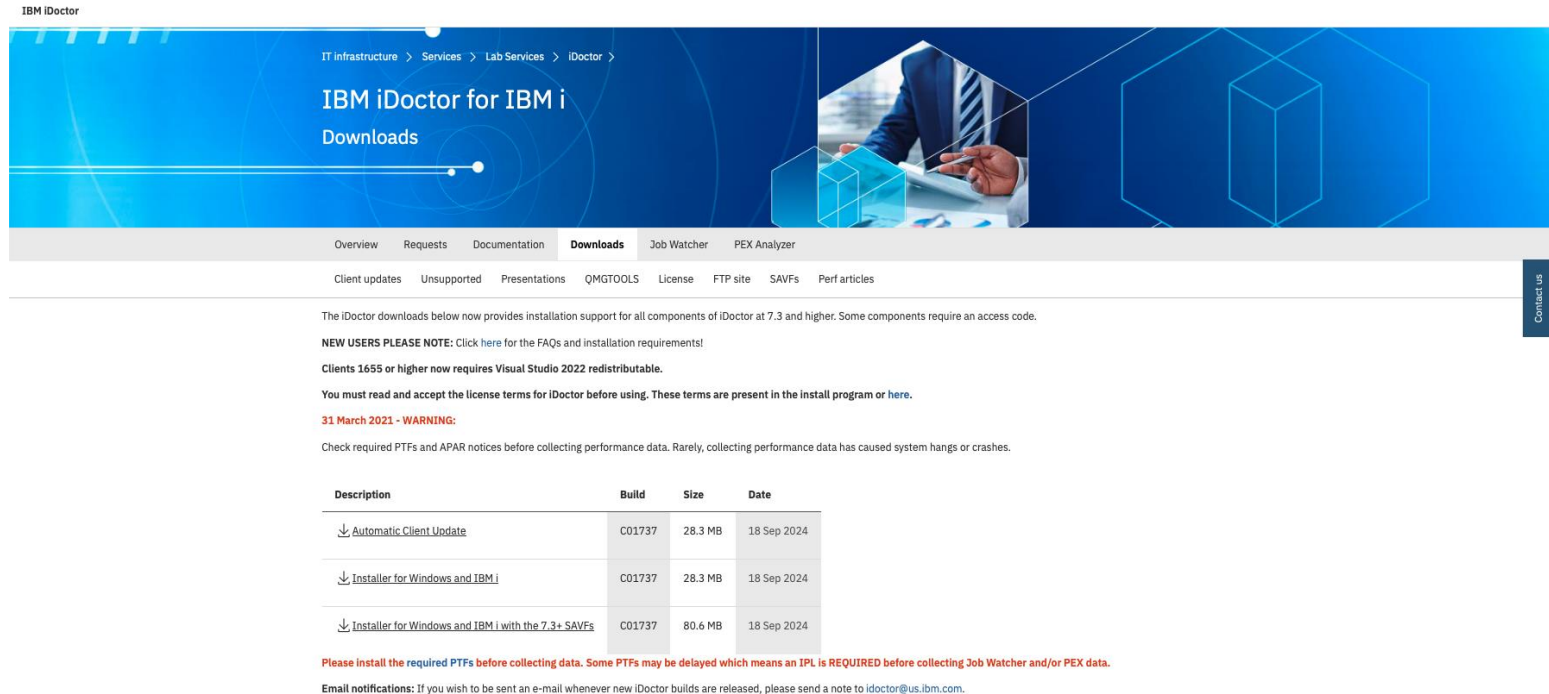
At the bottom of the window, there is a checkbox for 'Close window after clicking Launch' which is checked. Below this, there are fields for 'Access code:', 'System serial:' (78F3B80), and 'Processor group:' (P10). Buttons for 'Launch', 'Refresh', and 'Close' are also present.

Licence iDoctor



Installation du Serveur et du Client

Depuis le site principal IBM iDoctor for IBM i



IBM iDoctor

IT Infrastructure > Services > Lab Services > iDoctor >

IBM iDoctor for IBM i Downloads

Overview Requests Documentation **Downloads** Job Watcher PEX Analyzer

Client updates Unsupported Presentations QMGTOOLS License FTP site SAVFs Perf articles

Contact us

The iDoctor downloads below now provides installation support for all components of iDoctor at 7.3 and higher. Some components require an access code.

NEW USERS PLEASE NOTE: Click [here](#) for the FAQs and installation requirements!

Clients 1655 or higher now requires Visual Studio 2022 redistributable.

You must read and accept the license terms for iDoctor before using. These terms are present in the install program or [here](#).

31 March 2021 - WARNING:

Check required PTFs and APAR notices before collecting performance data. Rarely, collecting performance data has caused system hangs or crashes.

Description	Build	Size	Date
Automatic Client Update	C01737	28.3 MB	18 Sep 2024
Installer for Windows and IBM i	C01737	28.3 MB	18 Sep 2024
Installer for Windows and IBM i with the 7.3+ SAVFs	C01737	80.6 MB	18 Sep 2024

Please install the required PTFs before collecting data. Some PTFs may be delayed which means an IPL is REQUIRED before collecting Job Watcher and/or PEX data.

Email notifications: If you wish to be sent an e-mail whenever new iDoctor builds are released, please send a note to idoctor@us.ibm.com.



























<https://public.dhe.ibm.com/services/us/igsc/idoctor/html/downloadOptions.html>

Collection Services

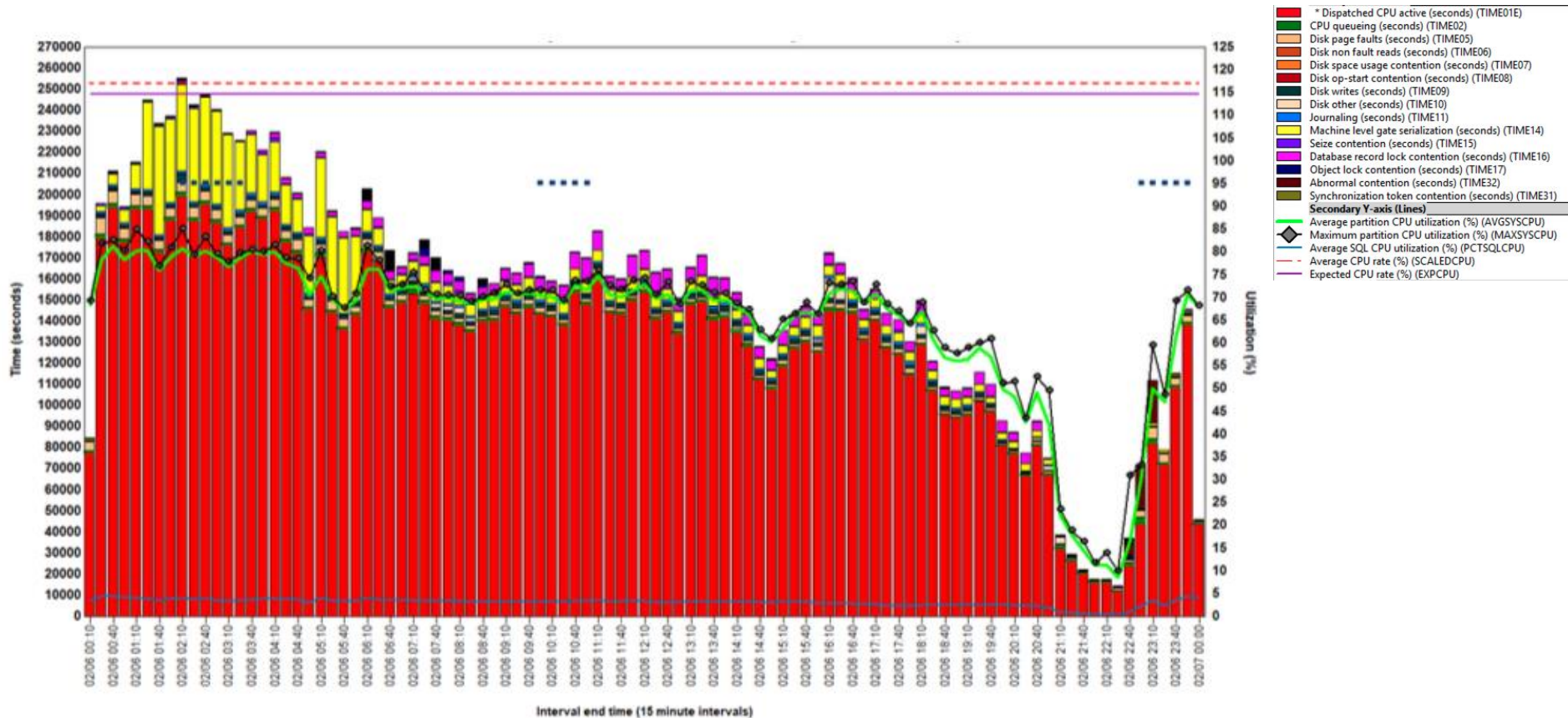
Commencez l'analyse avec les données des services de collecte (CSI)

- Qu'il s'agisse d'enquêter sur un problème signalé, de surveiller le fonctionnement de votre système ou de recherche d'amélioration, il peut s'avérer utile.
- Utilisez-le pour comprendre l'utilisation des ressources, ce qui est ou n'est pas utilisé, comment cela a changé, quand cela s'est produit, etc... En fait, il s'agit de cerner le problème !
 - ✓ Comment puis-je obtenir une vue système du fonctionnement de ma partition ?
 - ✓ Quels sont les travaux qui utilisent le plus de CPU ?
 - ✓ Quels sont les threads ou les tâches qui font le plus de fautes de page ?
 - ✓ Quelles tâches effectuent le plus grand nombre d'E/S sur disque physique ?
 - ✓ Quels types d'E/S ces tâches effectuent-elles ?
 - ✓ Quelles sont les caractéristiques d'exécution et d'attente de la partition dans le temps ?
 - ✓ Quelles sont les caractéristiques d'exécution/d'attente d'un travail dans le temps ?

Analyse de la collection et des waits

Report folder	Description	Tree table
 Collection overview time signature		
 Virtual CPU thread wait ready and dispatch latency		
 Dispatched CPU wait		
 Collection overview time signature with max waits in-progress		
 Seizes and locks time signature		
 Disk time signature		
 Journaling time signature		
 DB record lock time signature		
 Data queue receives time signature		
 Mutex contention time signature		
 Communications time signature		
 Wait counts		
 Data queue activity		
 Average wait times		
 Average CPU times		
 Average disk wait times		
 Dispatched CPU rankings	Ranks jobs by dispatched CPU usage	
 Disk page faults rankings	Ranks jobs by disk page faults	
 Disk time rankings	Ranks jobs by disk times	
 Seizes and locks		
 Wait counts rankings	Ranks jobs by wait bucket counts	
 Data queue activity rankings	Ranks jobs by data queue receive count	
 Average wait times rankings	Ranks jobs by average wait bucket time (all interesting waits in seconds)	
 Average CPU times rankings	Ranks jobs by average CPU/CPU queuing bucket times (in microseconds)	
 Average disk wait times rankings	Ranks jobs by average disk wait bucket time (in milliseconds)	
 Collection totals	Pie charts summarizing metrics in the collection	

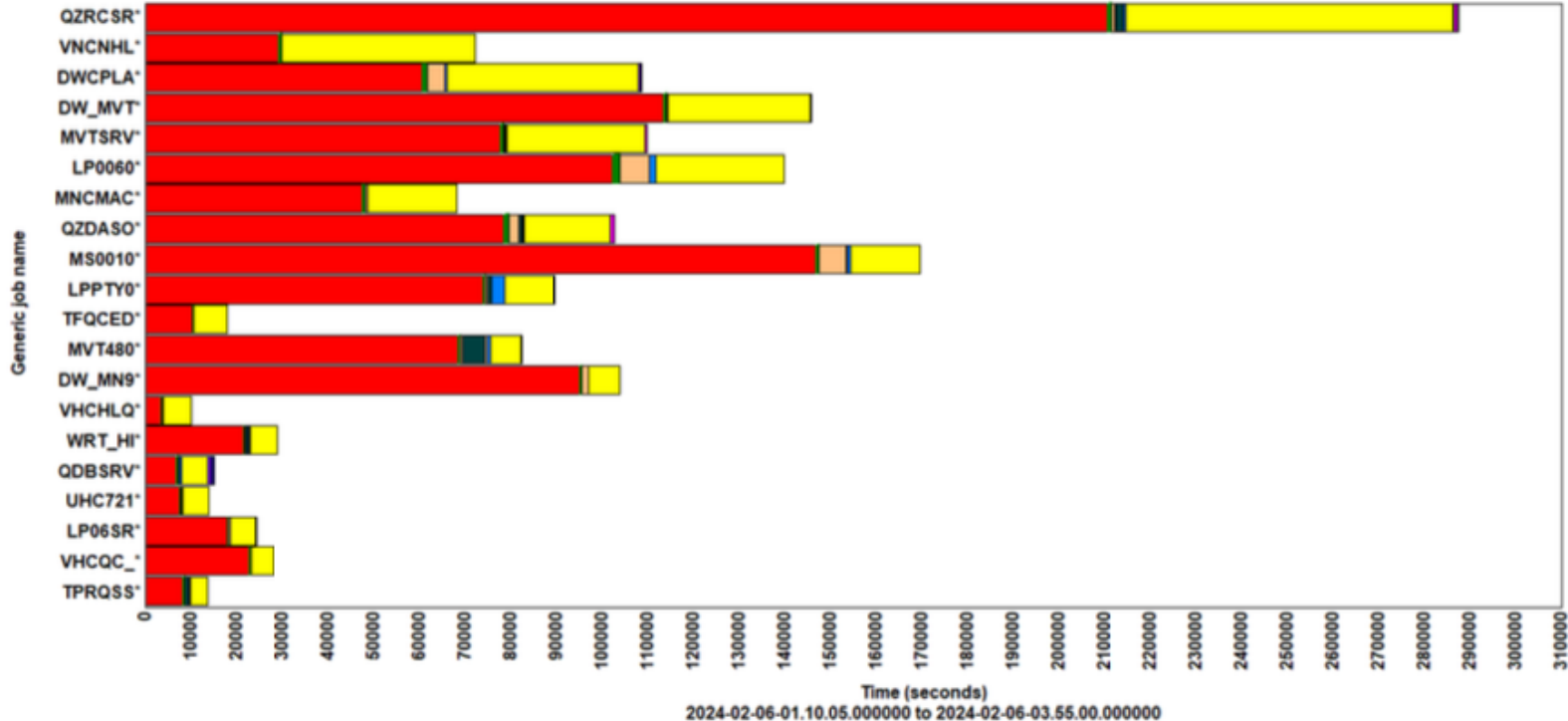
Vue d'ensemble de l'UC et des temps d'attente



Fonctionnalités d'exploration détaillée des services de collecte dans iDoctor

- Fonctions d'analyse approfondie pour analyser les temps d'attente ou l'utilisation de la CPU par
 - ✓ Sous-système
 - ✓ Jobs (génériques ou individuels)
 - ✓ Utilisateur actuel
 - ✓ Pool de mémoire
 - ✓ Priorité
 - ✓ Auteur du travail
 - ✓ Défaut de page
 - ✓ et de nombreux autres composants

Travaux impactés par les attentes des « gate serialization » entre 1h et 6h



Job Watcher – Détails de l'intervalle

Quick View **Call stack** Object waited on Wait buckets Physical I/Os Logical I/Os Transactions IFS **SQL** Other statistics Query

General:

Primary thread: QZDASOINIT / QUSER / 128968: 00000003 Interval: 19
Job subsystem: QUSRWRK Job status: LCKW Job function: Pool: 2
Current user profile: LISAW Current state: WAIT Priority (XPF/LIC): 20/160 Original LIC: 176
Current or last wait: (111/RDu) Db record lock: update Wait duration: 839.099 milliseconds
Object waited on: INVENTORY/INVENTORY (Record #37) Interval duration: 1.030 seconds
Holding job or task: QZDASOINIT / QUSER / 128890 Interval end: 2014-01-03-14.33.48.477000
SQL client job: None detected this interval

Call stack contents: **Advanced** Stack frames: 39 **Show Holder**

Call level	Program model	Program	Module	Procedure
001	LIC			qutde_block_tra
002	LIC			longWaitReceive_9QuCounterFR12RmprReceiverPvQ2_8TDQSEnum4EnumUIC
003	LIC			DBLockConflict_15RmsIDBHashClassFR11RmsIPImpLKIIP12RmsLKIEntryP12RmsIDBAnchorP16Rr
004	LIC			rmsIDBHLock_FR11RmsIPImpLad
005	LIC			getLockWithWait_18DbpmUpdateResourcecode
006	LIC			getLock_18DbpmUpdateResourcead
007	LIC			getRowLock_18DbpmUpdateResourceFCUIRC9DbpmDseldP
008	LIC			execute_18DbpmUpdateLockNodeFR13DbpmQueryIn
009	LIC			vPositionNextAndExecute_18DbpmUpdateLockNodeFR13DbpmQueryIn
010	LIC			positionNextEntryAndFetchOutline_17DbpmReadOnlyQueryFRQ2_17DbpmReadOnlyQuery10De
011	LIC			positionNextEntryAndFetch_17DbpmReadOnlyQueryFRQ2_17DbpmReadOnlyQuery10Descript
012	LIC			DbpmExecQEOWrapperCursorRequest_FPvP35DbpmQEOWrapperCursorRequestTemplaag
013	LIC			dbmai_
014	LIC			#cfm
015	LIC			syscall_A_port
016	ILE	QDBGGETSQQO	QDBGGETSQQO	QDBGGETSQgm



Exemple d'analyse: CPU élevée en raison de Full-Opens excessifs

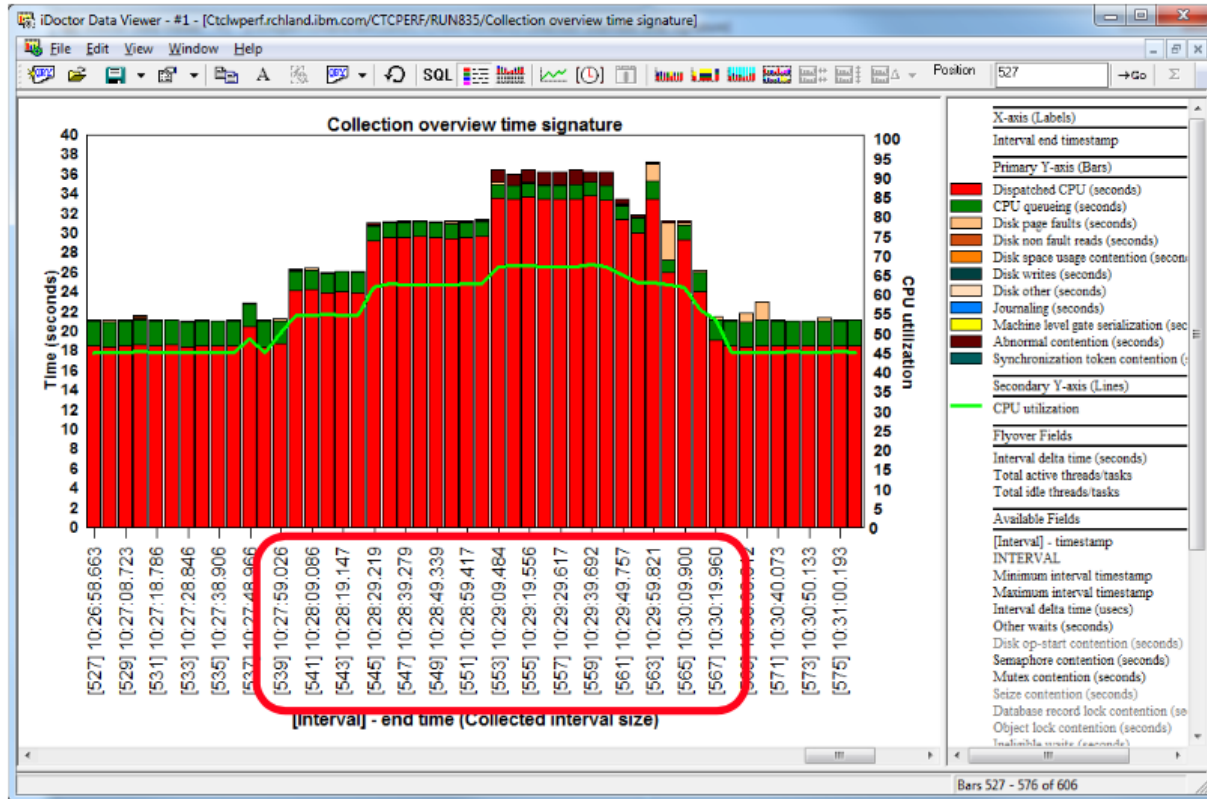
Ouverture et fermeture d'un fichier de base de données complet

- Les ouvertures complètes de bases de données sont gourmandes en ressources
 - ✓ Généralement le principal consommateur d'adresses temporaires
- Les Full Opens sont souvent effectuées des milliers de fois par seconde
 - ✓ Il pouvait y en avoir des centaines lorsque l'application a été déployée pour la première fois.
- Il est souhaitable d'avoir le moins d'ouvertures possible. Sur les grandes partitions, il ne faut pas dépasser quelques milliers par seconde.
- Il s'agit le plus souvent d'un problème lié à l'accès natif aux fichiers, plus rarement à SQL.
- Il n'est pas difficile d'identifier la source, ni d'apporter les modifications nécessaires à l'application (en général).

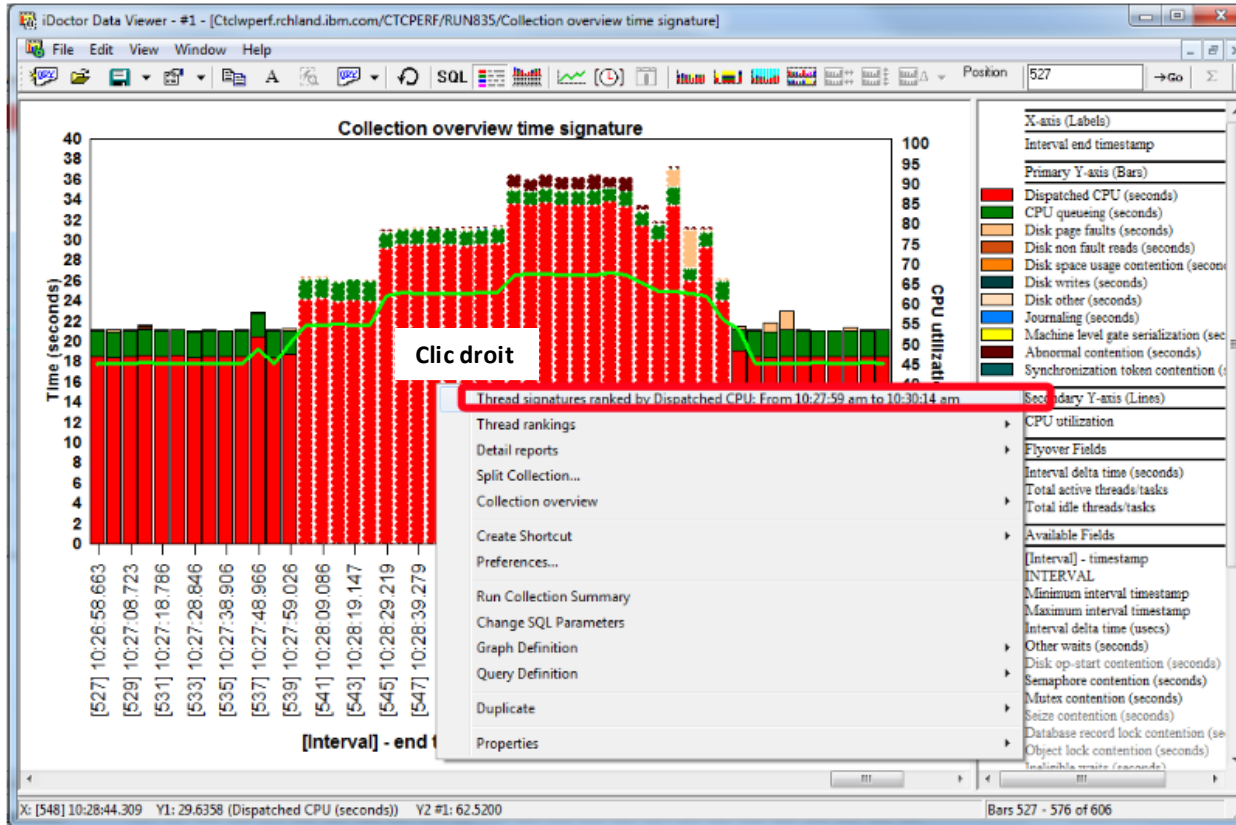
Les ouvertures de fichiers de base de données sont gourmandes en ressources

- Pour chaque fichier de base de données natif entièrement ouvert, les opérations suivantes sont effectuées :
 - ✓ Recherche dans la liste des bibliothèques pour le fichier en cours d'ouverture
 - ✓ Vérification de l'autorité sur le fichier en cours d'ouverture
 - ✓ Mise à jour du tableau d'utilisation de la base de données du système
 - ✓ Insertion d'entrées dans la table de saisie/verrouillage du système
 - ✓ Création d'objets temporaires (utilisation d'adresses temporaires) pour le curseur du fichier (chemin de données ouvert ODP)
 - ✓ Duplication des objets dans SLIC pour chaque fichier ouvert
- Lorsqu'un fichier est fermé, bon nombre de ces événements doivent être répétés.

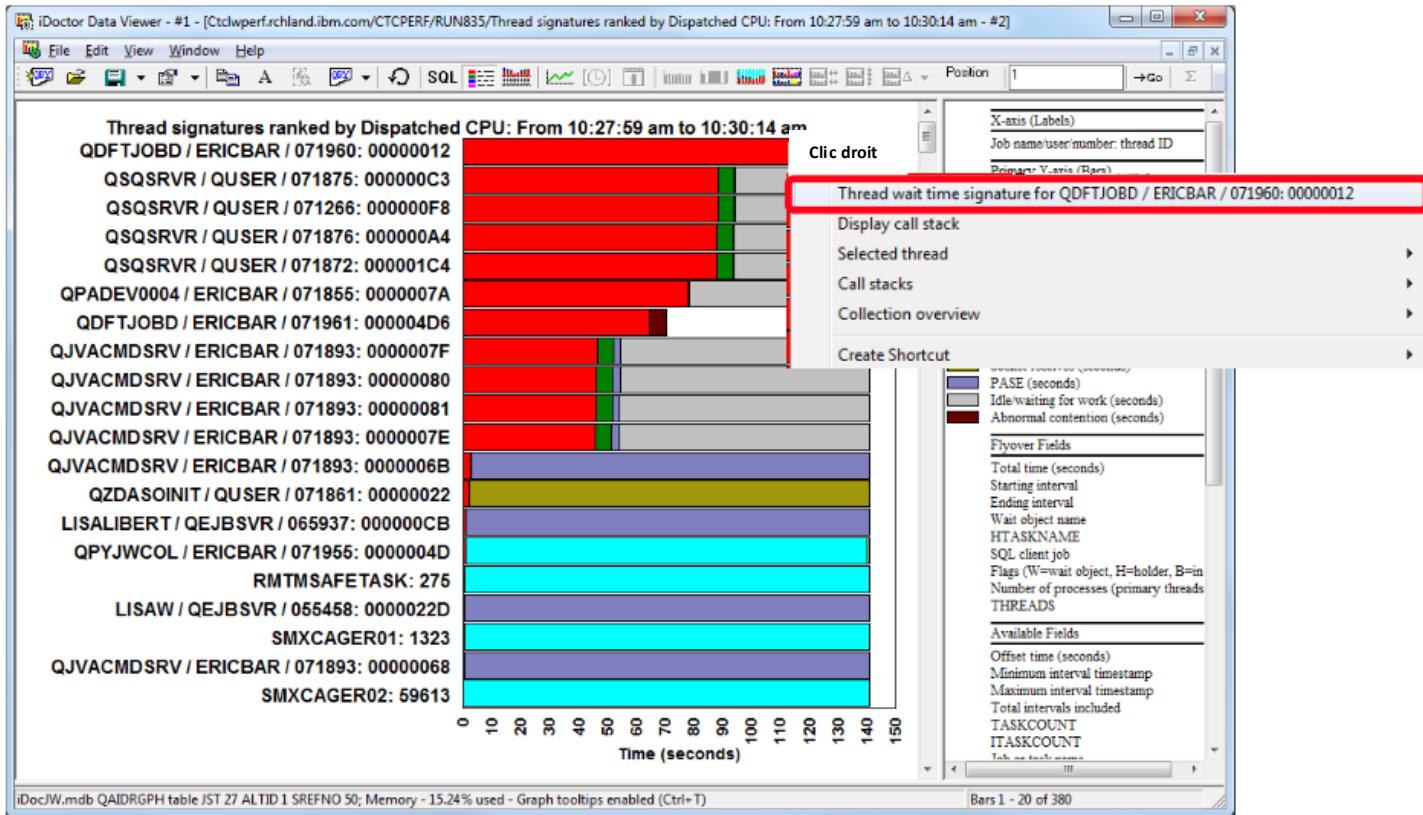
CS/JW Overview Time Signature



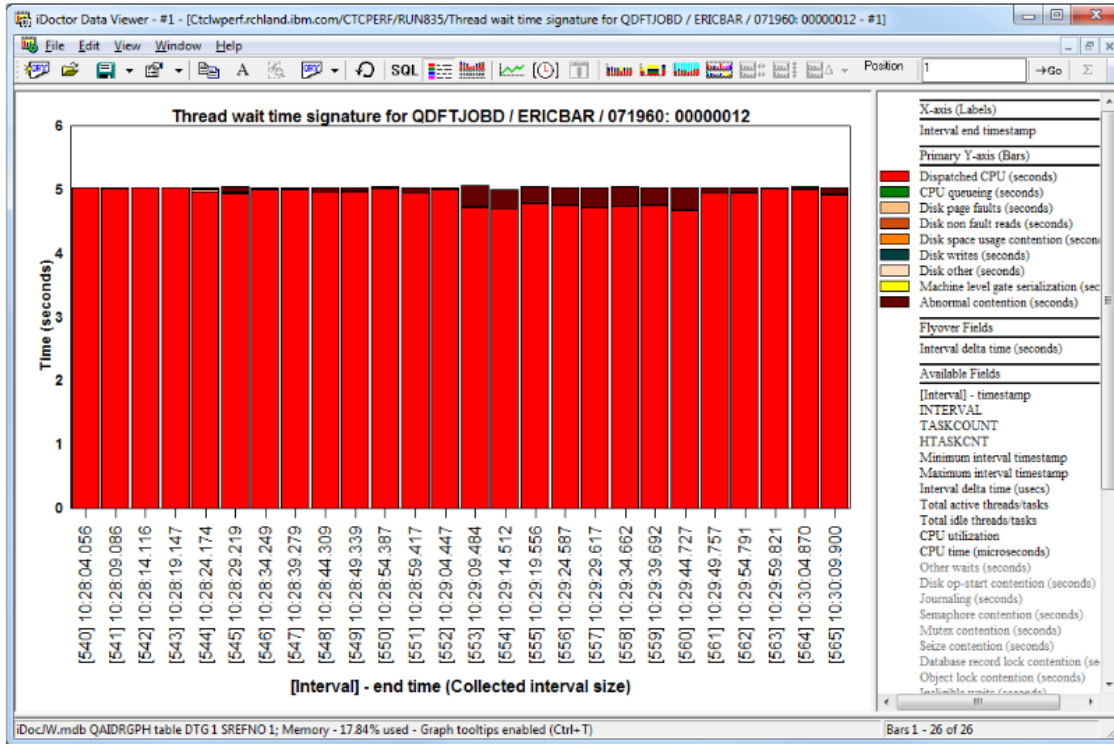
CS/JW Exploration détaillée



CS/JW Classement usage CPU



CS/JW Single Thread Run/Wait Signature



Il s'agit de la durée totale du Thread (pas de barres de défilement). Correspond à la période qui nous intéresse.

Niveau d'analyse suivant

Jusqu'à présent – Collection Services ou Job Watcher

- ✓ Durée du problème
- ✓ Type de problème (CPU)
- ✓ Jobs concernés par le problème

Prochaine étape - Job Watcher

- ✓ Que faisaient les tâches ?
- ✓ Quels programmes les tâches exécutaient-elles ?
- ✓ Les tâches attendaient-elles quelque chose ?

Vous utilisiez un moniteur Job Watcher, n'est-ce pas ?

Job Watcher : Assistant de définition

Sélection du job

Basic Options

< Back Next > Cancel

Specify the definition name and other optional parameters below.

Definition name: Actions

Description:

Interval duration: 0.1 - 3,600.0 seconds

Collect as fast as possible

Show advanced options

Data collection options:

Call stacks: Every interval
Collect for jobs in conflict for at least 1 microseconds
Collect for jobs in bad waits for at least 1 microseconds

Add Job Watcher Definition Wizard - Data Collection Options

Call stack SQL Java Activation groups Communications Conditions file

OK Cancel

Collection frequency

Every interval

Never

Every Nth interval

Include call stacks for jobs in conflict > N microseconds (DOES NOT WORK)
N:

Include call stacks for jobs in "bad waits" > N microseconds (DOES NOT WORK)
N:

Job Watcher : Définitions fournies en standard

The screenshot shows the Job Watcher application window. The left pane displays a tree view with the following items: Job Watcher, Libraries, Definitions (selected), Data repository, JVM analysis, SQL tables, Monitors, FTP Definitions, and IBM i Explorer. The main pane displays a table with the following columns: Definition, Description, and Command.

Definition	Description	Command
QIBMJW	IBM JW Definition	QSYS/ADDJWDFN DFN(QIBMJW) TEXT('IBM
Q1SEC	1 second intervals, Call stacks	QSYS/ADDJWDFN DFN(Q1SEC) TEXT('1 secc
Q1SECJ	1 second intervals, Call stacks, J9	QSYS/ADDJWDFN DFN(Q1SECJ) TEXT('1 sec
Q1SECSQL	1 second intervals, Call stacks, Sql	QSYS/ADDJWDFN DFN(Q1SECSQL) TEXT('1 :
Q1SECSQLJ	1 second intervals, Call stacks, Sql, J9	QSYS/ADDJWDFN DFN(Q1SECSQLJ) TEXT('1
Q10SEC	10 second intervals, Call stacks	QSYS/ADDJWDFN DFN(Q10SEC) TEXT('10 se
Q10SECJ	10 second intervals, Call stacks, J9	QSYS/ADDJWDFN DFN(Q10SECJ) TEXT('10 s
Q10SECSQL	10 second intervals, Call stacks, Sql	QSYS/ADDJWDFN DFN(Q10SECSQL) TEXT('1
Q10SECSQLJ	10 second intervals, Call stacks, Sql, J9	QSYS/ADDJWDFN DFN(Q10SECSQLJ) TEXT('
Q3MINQZDAS	QZDASOINIT jobs, 3 min intervals	QSYS/ADDJWDFN DFN(Q3MINQZDAS) TEXT
Q5MINQZDAS	QZDASOINIT jobs, triggers PEX stats	QSYS/ADDJWDFN DFN(Q5MINQZDAS) TEXT
Q5SEC	5 second intervals, Call stacks	QSYS/ADDJWDFN DFN(Q5SEC) TEXT('5 secc
Q5SECJ	5 second intervals, Call stacks, J9	QSYS/ADDJWDFN DFN(Q5SECJ) TEXT('5 sec
Q5SECSQL	5 second intervals, Call stacks, Sql	QSYS/ADDJWDFN DFN(Q5SECSQL) TEXT('5 :
Q5SECSQLJ	5 second intervals, Call stacks, Sql, J9	QSYS/ADDJWDFN DFN(Q5SECSQLJ) TEXT('5

The context menu for the selected definition (Q5SECSQLJ) includes the following options: Change Definition..., Add Definition..., Start Collection..., Start Monitor..., Delete, and Properties.


Job Watcher : Concepts du moniteur

- En général, il n'y a pas vraiment de surveillance ; il s'agit simplement de collecter des données.
 - ✓ Mais peut être configuré pour déclencher des actions basées sur des règles
- Collecte continue, nettoyage automatique - comme les services de collecte
- Nécessité de redémarrer manuellement après l'IPL ou d'automatiser le démarrage après l'IPL
- Conservez autant d'heures d'historique que vous le souhaitez pour remonter le temps
 - ✓ 24 heures ou plus, c'est très bien, si vous avez l'espace nécessaire
- Il est recommandé de l'exécuter 24 heures sur 24 et 7 jours sur 7 sur toutes les partitions IBM i de production
- Pas de charge de ressources notable en utilisant les définitions recommandées

Job Watcher : Configuration du moniteur

Options

< Back Next > Cancel



Monitor name:

Library name: ASP limit: %

Max duration: 1.00 - 1440.00 minutes

Max size: 1 - 9999999 megabytes

Max collections: 2 - 999

Run JW active summary Run analyses automatically

Resubmit JW on failure

Max resubmits:

Description:

Definition:

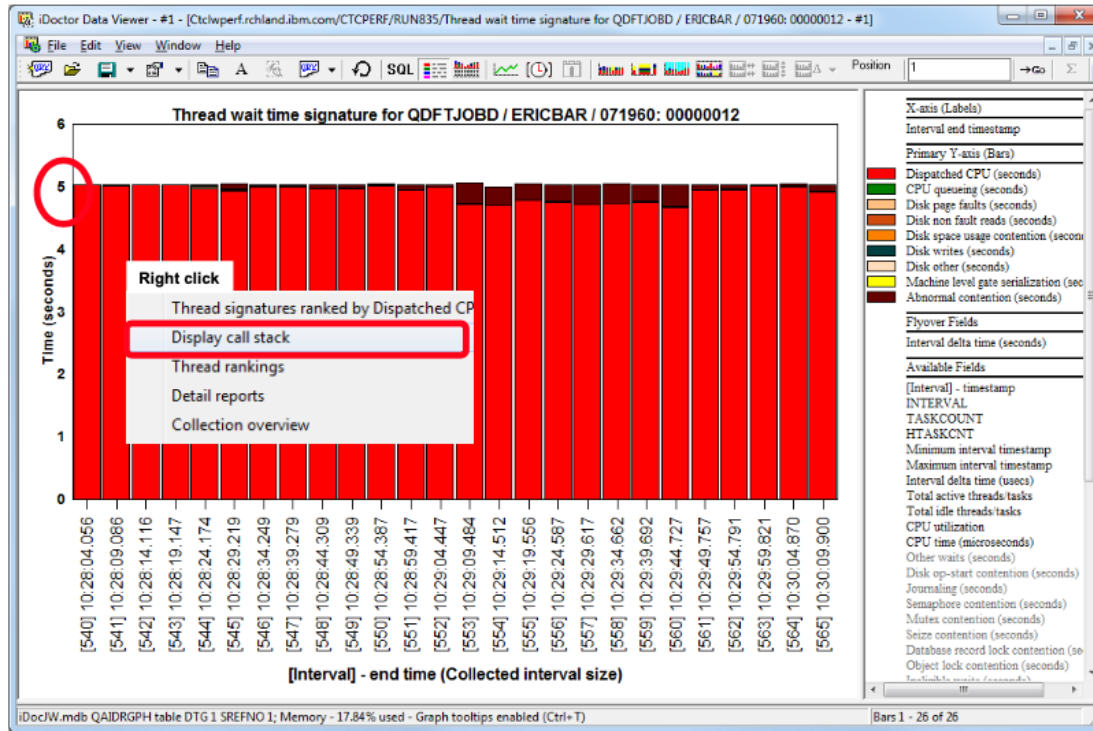
Job Watcher

Disk Watcher

PEX Analyzer

Utiliser les définitions Q5SECSQL ou Q10SECSQL

Job Watcher Single Thread Run/Wait Signature



Il s'agit de la durée totale du Thread (pas de barres de défilement). Correspond à la période qui nous intéresse.

Job Watcher Call Stack

The screenshot shows the 'iDoctor Data Viewer' interface. The 'General' section displays job details for 'QDFTJOB / ERICBAR / 071960: 0000012'. The 'Wait duration' is highlighted in red and shows '0 microseconds'. The 'Interval' is set to '545'. The 'Call stack contents' table below shows a sequence of calls, with call levels 004, 010, and 011 highlighted in red. Call level 004 is 'QDBOPEN', call level 010 is 'ORNXIO', and call level 011 is 'BAD_OPEN'.

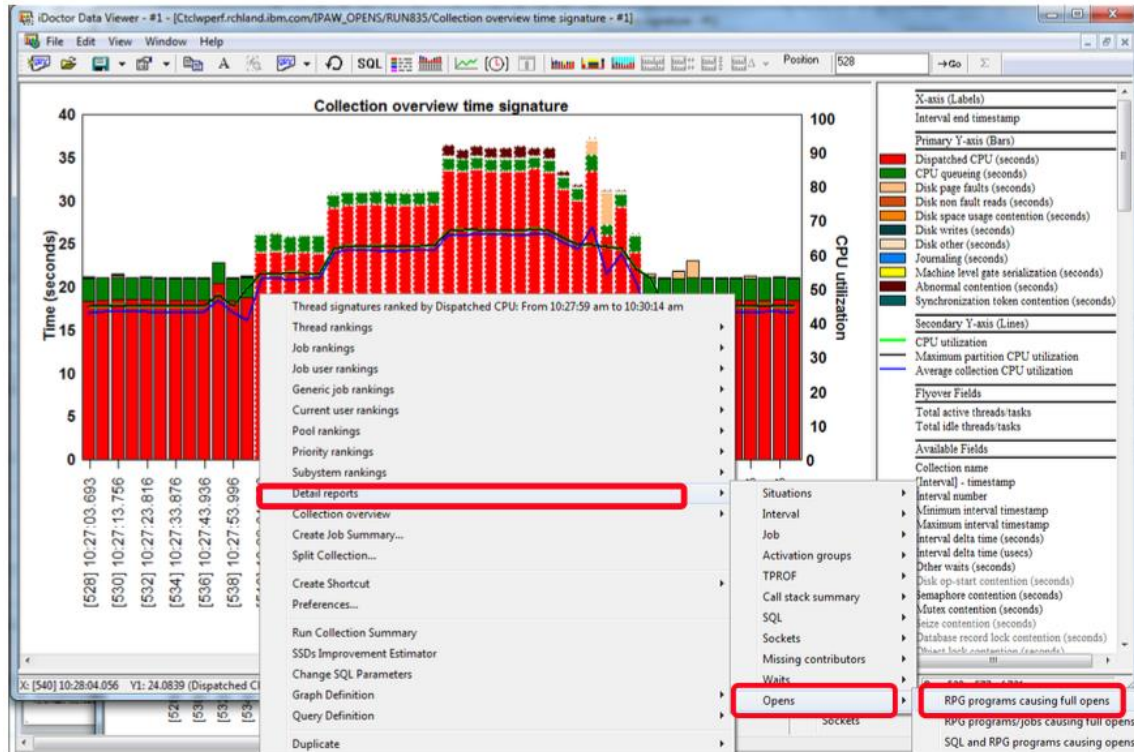
Call level	Program	Module	Offset	Procedure	TBT address	Procedure start address
001			00000060	syscall_3_portal	FFFFFFFF490E4CE8	FFFFFFFF
002	QDBGGENSP	QDBGGOBJLN	00002308	QDBGGEN_GET_OBJ_SYP_LIB_NAME	0B2C7CA044012A98	0B2C7CA0
003	QDBXREF	QDBXNOSYNA	000001AC	QDBX_NO_SYNC_REQ_ALLOWED	251063F4A6022220	251063F4
004	QDBOPEN	QDBOPEN	0000BEA0	QDBOPEN	1799E8CDF7036968	1799E8CD
005			000001D0	cblabbranch	FFFFFFFF492C9728	FFFFFFFF
006			000000C4	aiuser_program_call_portal	FFFFFFFF38285870	FFFFFFFF
007	QDMCOPEN		0000AA28		39A15363550237A0	39A15363
008			000001D0	cblabbranch	FFFFFFFF492C9728	FFFFFFFF
009			000000C4	aiuser_program_call_portal	FFFFFFFF38285870	FFFFFFFF
010	ORNXIO	ORNXCOCIO	00000138	ORNX_OPEN	00FB71B4A202A148	00FB71B4
011	BAD_OPEN	BAD_OPEN	00000B5C	BAD_OPEN	2263BCD3B5003C98	2263BCD3
012	BAD_OPEN	BAD_OPEN	00000B5C	_QRNF_FET_BAD_OPEN	2263BCD3B5003F88	2263BCD3

QDBOPEN : Open a database file

Vérifier plusieurs intervalles

De nombreux intervalles peuvent ne pas avoir de call stacks complet

Job Watcher : Rapport détaillé des Full Opens



Job Watcher : Rapport détaillé des Full Opens

The top screenshot shows a table with the following data:

Hit count	Program library	Program name	Procedure name
11	IPAW_OPENS	BAD_OPEN	BAD_OPEN

The bottom screenshot shows a table with the following data:

Hit count	JOB	USER	OPENTYPE	Program library	Program name	Procedure name
10	QDFTJOB	ERICBAR	_QRNX_OPEN	IPAW_OPENS	BAD_OPEN	BAD_OPEN
1	QPADEV0004	ERICBAR	_QRNX_OPEN	IPAW_OPENS	BAD_OPEN	BAD_OPEN

Cela correspond à ce que nous avons vu dans le call stack


Trouver la source des Full Opens

- Performance Explorer (PEX) pour identifier les programmes et les fichiers
 - ✓ Lors de l'exécution de PEX, assurez-vous que les PTF de PEX sont à jour.
- Définition unique ou utilisation de l'assistant d'analyse iDoctor PEX (diapo suivante)
 - ✓ `ADDPEXDFN DFN(DBOPENS) TYPE(*TRACE) JOB((*ALL *ALL))
MAXSTG(400000) TRCTYPE(*SLTEVT) SLTEVT(*YES) MCHINST(*NONE)
OSEVT((*DBOPEN *NONE *FORMAT2)) TEXT('Système entier - full opens avec
call stack')`
- Peut également effectuer un ou plusieurs travaux spécifiques.
- Peut collecter plusieurs traces
 - ✓ `STRPEX SSNID(nom_unique) DFN(DBOPENS)`
 - ✓ `ENDPEX DTALIB(nom_bib)`, 200k événements à la fois, c'est bien
- Conseils
 - ✓ Peut collecter beaucoup de données très rapidement
 - ✓ Charge de ressource minimale lors de la collecte

PEX Analyzer assistant : Sélection du problème

Welcome

< Back Next > Cancel



Welcome to the Start PEX Collection Wizard.

This wizard will guide you through the process of creating a PEX collection.

Your request to create a collection will be submitted to a batch job on the server. This collection may take a considerable amount of time to complete.

NOTE: If you are already familiar with the types of PEX data that can be collected we recommend using Advanced mode.


Selected desired mode:

Basic

Advanced

Problem Type

< Back Next > Cancel




What type of problem do you suspect?

- Excessive CPU utilization
- Excessive disk I/O operations
- Excessive file opens
- Excessive disk space consumption
- Excessive allocations to the system heap
- Job(s) perform badly when created/destroyed
- Slow data queue operations
- Slow data area operations
- Slow logical I/O operations
- Websphere application(s) is running poorly

PEX Analyzer assistant : Options de collection

Options

< Back Next > Cancel



Definition type: PEX-Analyzer-supplied User-defined

Definition: *DB_OPEN Details...

Start in standby (suspended) mode

Collection name:

Library: QPADATA

Description:

Duration: 1 - 1440 minutes

Max data: 1 - 250 GB

Max events: Leave blank for no max

Include PMCO events


CPU interval sample: 0.1 - 1000.0 ms

Advanced options:

Scheduled start time: Immediate

PEX Analyzer assistant : Sélection des travaux

Job/Task Options < Back Next > Cancel



Collect over all jobs and/or tasks or pick specific ones.

Job selection:

All jobs Selected jobs None

Task selection:

All tasks Selected tasks None


Indicate the jobs to include in your collection below:

Name: QZ* User: *ALL Number: *ALL Add

Current user: Subsystem: Status: Active Refresh

Jobs matching the job filter information: Reset Reset Statistics Add Selected

Job Selection < Back Next > Cancel



Select the jobs to include below:

Job selection criteria: Add...

Remove All Remove

Job Name	User	Number
QZDASOINIT	QUSER	118397

PEX Analyzer - Run Database File Full Opens/Closes

(après la collecte des données via l'interface graphique iDoctor ou STRPEX/ENDPEX)

The screenshot shows the PEX Analyzer interface. On the left is a tree view with categories like Libraries, Monitors, SQL tables, Browse collections, Active collections, PEX objects, Super collections, Definitions, Filters, Saved collections, Work management, ASPs, and Disk units. The main area displays a table of collections. A context menu is open over the collection 'Db0530a', with the 'Analyses' option selected. A sub-menu is also open, showing various analysis options, with 'Run Database File Full Opens/Closes' highlighted. Red boxes and arrows highlight the 'Analyses' menu item and the specific analysis option.

Collection	Status	Type	Collection size (MB)	Partition collected on VRM	Start time
SQL tables: (IPAW_OPENS*)					
Db0530a	Ready for analysis	Trace	241.8 MB	V7R1M0	2013-05-30-09.48.44.422

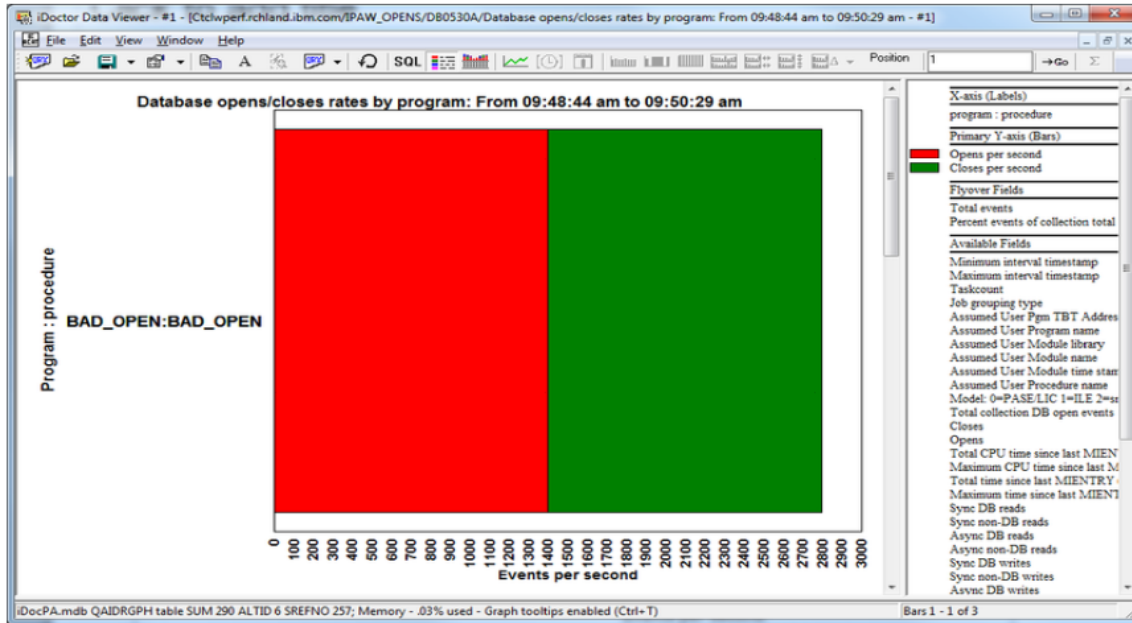
- Analyses
 - Analyze Collection...
 - Run ALL Default Analyses
 - Run Events
 - Run Call Stacks (Format 2 or 4)
 - Run Database File Full Opens/Closes
 - Run Trace details (SMTRMOD-like output)

PEX Analyzer – Analyse des Full Opens

The screenshot displays the PEX Analyzer interface. On the left, a tree view shows the project structure, including 'Libraries: (IPAW_OPEI)', 'Ipaw_opens', 'Monitors', 'SQL tables', 'Browse collections', 'Active collections', 'PEX objects', 'Super collections', 'Definitions', 'Filters', 'Saved collections', 'Work management', 'ASPs', 'Disk units', and 'Objects owned by ERIC'. The main pane shows a table of SQL tables with the following columns: Collection, Status, Type, Collection size (MB), Partition collected on VRM, Start time, and End time. The table contains one row for 'Db0330a' with a collection size of 7771 MB and start/end times of 2013-05-30-09:48.44.422357 and 2013-05-30-09:50.29.61962 respectively. A context menu is open over the table, listing various actions. Three options are highlighted with red boxes: 'Database file opens/closes', 'Opens/closes totals rankings', and 'Database opens/closes rates by program: From <<MINDTETIM>> to <<MAXDTETIM>>'. The menu also includes options for 'Explore', 'Record Quick View', 'Analyses', 'Events', 'PEX file(s) starting points', 'PEX collection files', 'User-defined queries', and 'Generate Reports'. Below the menu, a list of report options is visible, including 'Database opens/closes rates by thread', 'Database opens/closes rates by job', 'Database opens/closes rates by job user name', 'Database opens/closes rates by generic job name', 'Database opens/closes rates by file', 'Database opens/closes rates by program', and 'Database opens/closes rates by program and file'.

Collection	Status	Type	Collection size (MB)	Partition collected on VRM	Start time	End time
SQL tables: (IPAW_OPENS*)						
Db0330a			7771	M0	2013-05-30-09:48.44.422357	2013-05-30-09:50.29.61962

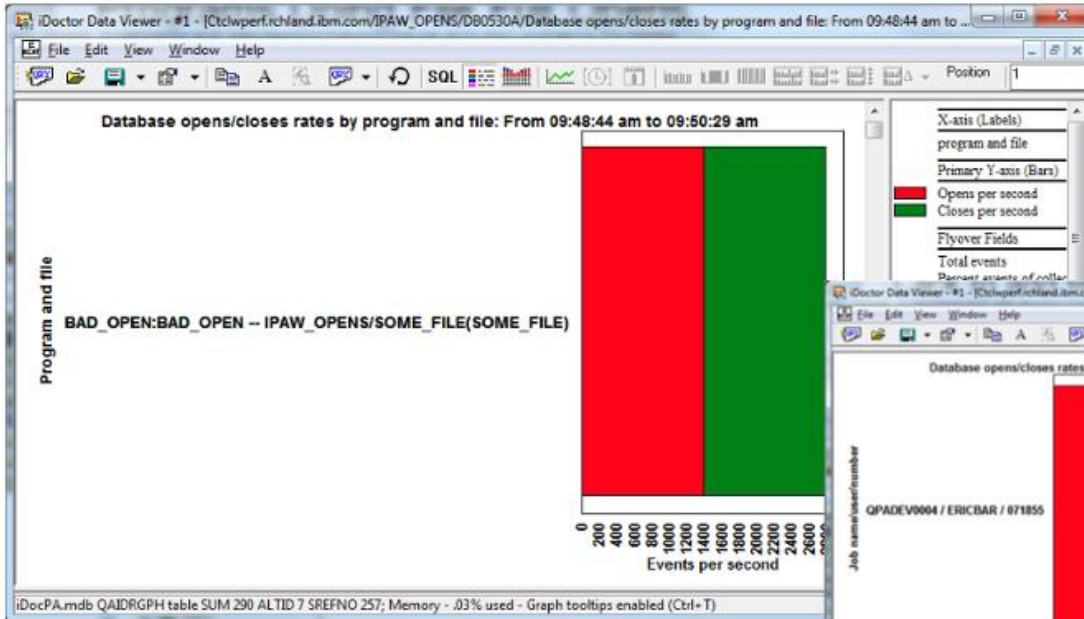
Programmes effectuant des Full Opens (et fermetures) de fichiers complets



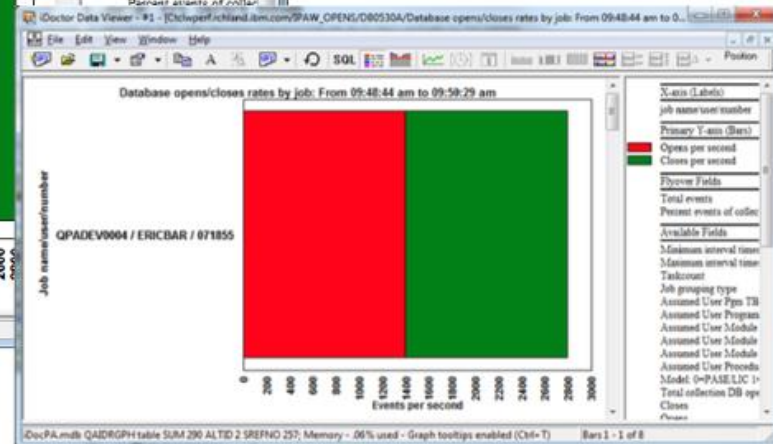
Correspond au programme vu dans le call stack de Job Watcher.

- ✓ Collecte toutes les ouvertures complètes
- ✓ 1400/seconde montré ici
- ✓ n'a capturé que 11 ouvertures complètes en 2 minutes de données de Job Watcher montrées plus haut

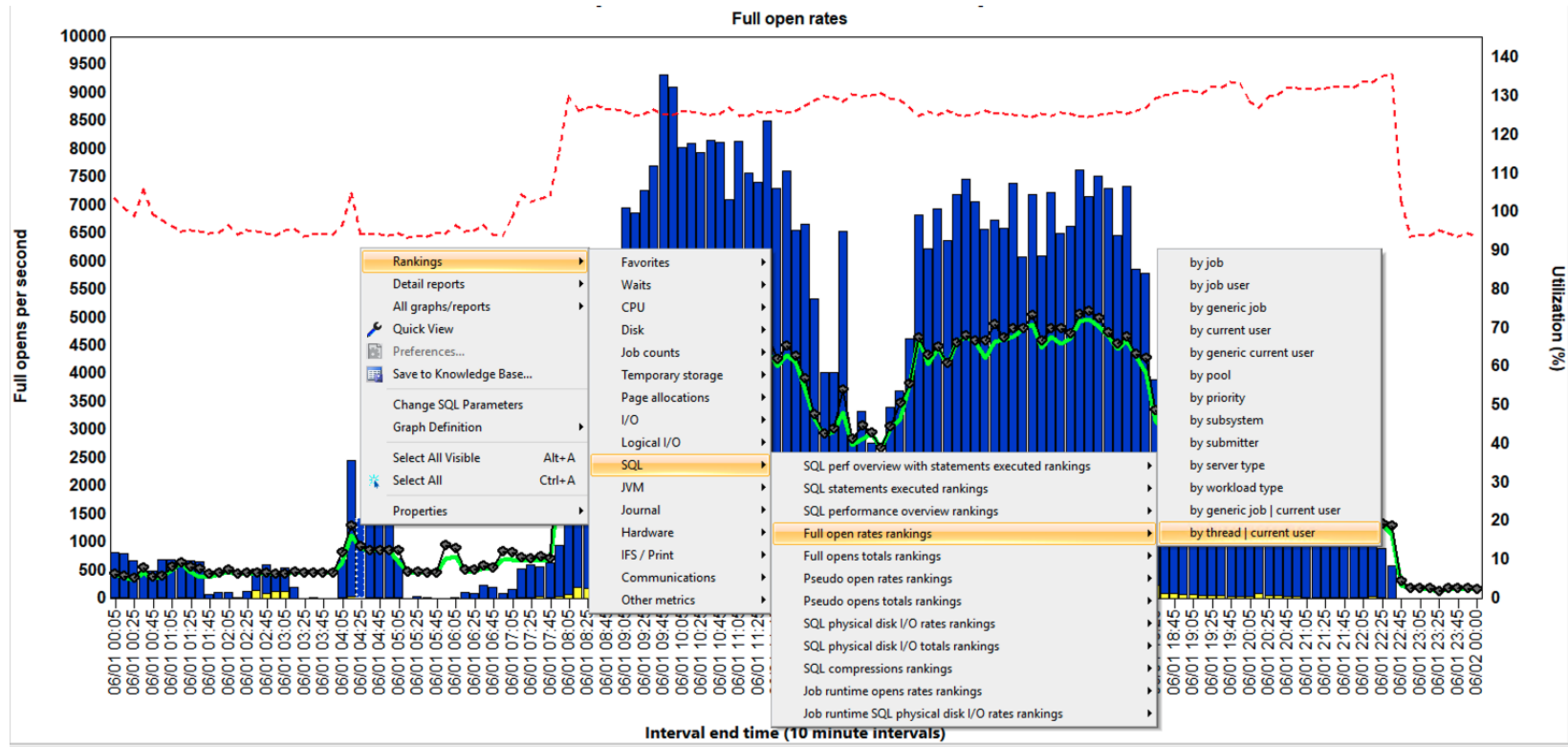
Full opens par Programme + fichier et par Job



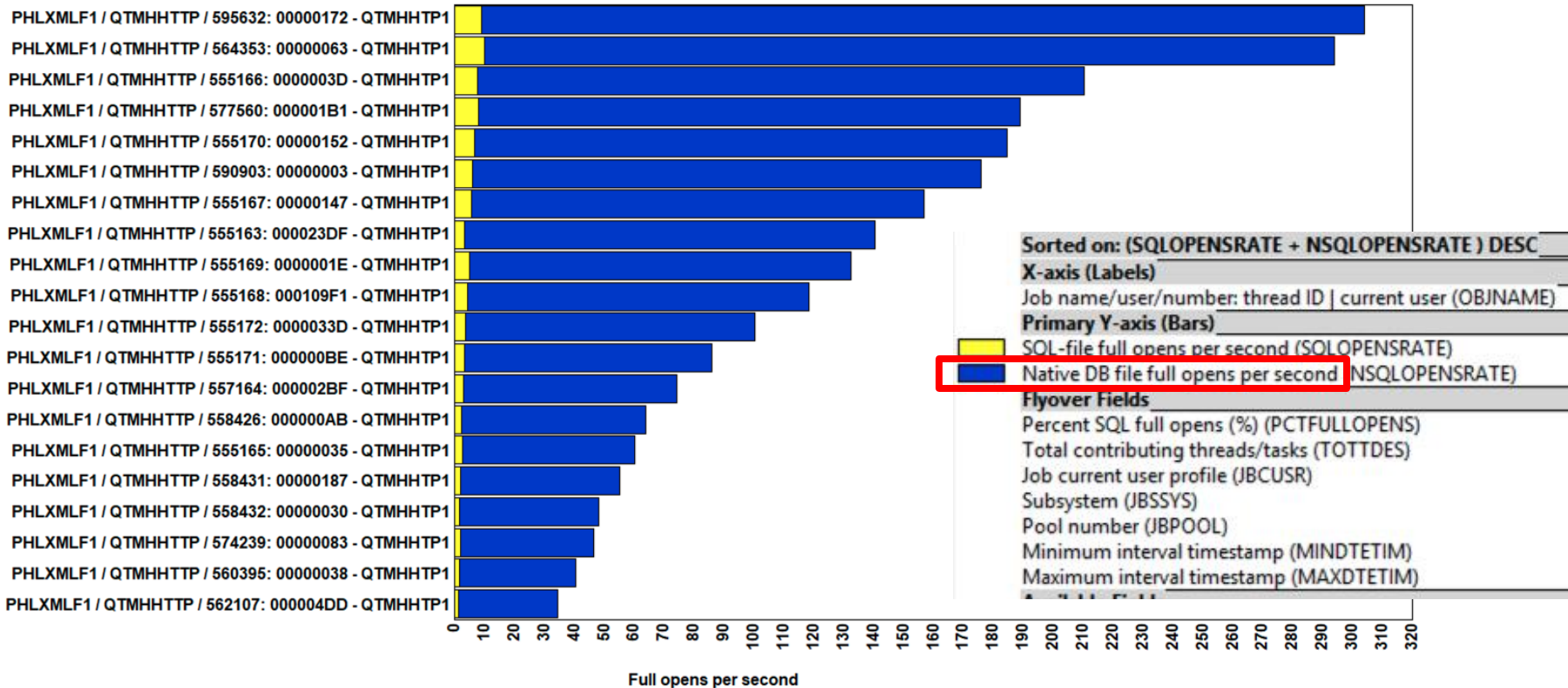
De nombreuses autres vues sont disponibles.



Collection Services : Full Opens



Collection Services : Full Opens Rates By Thread



En résumé

- Méthodologie de performance utilisée - Analyse ciblée
 - ✓ Utilisation des services de collecte pour cerner le problème
 - ✓ Utilisation de Job Watcher pour obtenir plus de détails et trouver des suspects
 - ✓ Utilisation de PEX pour identifier la cause première du problème
- Résultat final (avec modifications de l'application)
 - ✓ Réduction de l'utilisation de l'unité centrale
 - ✓ Réduction des E/S de synchronisation lors de la fermeture des fichiers
 - ✓ Augmentation de l'évolutivité de l'application

IBM i / Db2 for i Services liés aux Full Opens

IBM i Services that output Db2 information, e.g., **ACTIVE_JOB_INFO**

<https://www.ibm.com/support/pages/ibm-i-services-sql>

Db2 for i Services, e.g., **ACTIVE_QUERY_INFO**

<https://www.ibm.com/docs/en/i/7.5?topic=optimization-db2-i-services>

Catalog tables or catalog views, e.g., **SYSTABLES**

<https://www.ibm.com/docs/en/i/7.5?topic=views-i-catalog-tables>

ACTIVE_JOB_INFO – IBM i Service w/SQL Info

```
SELECT JOB_NAME, AUTHORIZATION_NAME,  
FULL_OPEN_CURSOR_COUNT  
FROM TABLE (QSYS2.ACTIVE_JOB_INFO(DETAILED_INFO => 'ALL')) X  
WHERE FULL_OPEN_CURSOR_COUNT > 0  
order by FULL_OPEN_CURSOR_COUNT desc;
```

JOB_NAME	AUTHORIZATION_NAME	FULL_OPEN_CURSOR_COUNT
107178/QBRMS/Q1ACPDST	QBRMS	576889
107089/QSRVAGT/QS9PALMON	QSRVAGT	803
118397/QUSER/QZDAS0INIT	QSECOFR	355
118393/QUSER/QZDAS0INIT	QSECOFR	313
118388/QUSER/QZDAS0INIT	QSECOFR	310

SYSTABLESTAT – Db2 Catalog Table/View

Informations utiles pour la performance, comme le nombre de **file full opens**, enregistrements supprimés, etc..

```
SELECT TABLE_SCHEMA, TABLE_NAME, OPEN_OPERATIONS,  
DATE(LAST_USED_TIMESTAMP) AS "LAST USED", DAYS_USED_COUNT  
FROM QSYS2.SYSTABLESTAT  
WHERE TABLE_SCHEMA NOT LIKE 'Q%'  
ORDER BY OPEN_OPERATIONS DESC FETCH FIRST 100 ROWS ONLY
```

TABLE_SCHEMA	TABLE_NAME	OPEN_OPERATIONS	LAST USED	DAYS_USED_COUNT
PREZIBMI	QAPMJOBOS	1871	2024-09-23	2
PREZIBMI	QAPMJOBMI	1870	2024-09-23	2
PREZIBMI	QAPMCONF	252	2024-09-23	120
PREZIBMI	QAPMSYSTEM	24	2024-09-23	3
PREZIBMI	QAPMDISK	17	2024-09-23	2
PREZIBMI	QAPMSQLPC	13	2024-09-23	2
PREZIBMI	QAPMSYSCPU	12	2024-09-23	2

Marche à suivre pour identifier le problème d'ouverture et de fermeture complète d'un fichier de base de données natif

- Utilisez les services de collecte et iDoctor CSI pour identifier...
 - ✓ Taux d'ouvertures complètes au niveau des partitions
 - ✓ Les tâches et les threads qui effectuent le plus grand nombre d'ouvertures complètes de fichiers
 - ✓ Le meilleur moment pour exécuter les traces PEX afin d'obtenir des détails
- Les piles d'appels de Job Watcher peuvent être utilisées pour identifier certains programmes.
- Les traces de Performance Explorer peuvent être utilisées pour identifier...
 - ✓ les programmes effectuant des ouvertures complètes
 - ✓ les fichiers ouverts par les programmes
 - ✓ les numéros de déclaration dans les programmes, lorsque des ouvertures complètes se produisent.
- Utiliser IBM i ou Db2 for i Services pour identifier...
 - ✓ Les fichiers les plus ouverts
 - ✓ Les tâches actives avec le plus grand nombre d'ouvertures complètes de fichiers

Options pour la suppression des ouvertures complètes (E/S natives)

- Garder les fichiers ouverts entre les appels au même programme dans le même Thread
 - ✓ Supprimer les fermetures implicites
 - RPG : suppression de « MOVE *ON *INLR » / « set on LR ».
 - COBOL utiliser « EXIT PROGRAM » au lieu de « STOP RUN ».
 - ✓ Supprimer les fermetures de fichiers explicites
- Partage des fichiers ouverts entre différents programmes d'un même processus («shared opens »)
 - ✓ Changements de QDBOPEN à QDBSOPEN
 - Doit vérifier que le curseur est toujours positionné correctement lors de la ré-entrée (et de l'entrée pour le partage)
- Changer les E/S natives en SQL avec CLOSQLCSR(*ENDACTGRP)
 - ✓ Laisser SQL s'en occuper

Revue de performance d'une partition IBM i

- **La prestation de service inclut:**
 - ✓ Analyse de performance de votre partition IBM i
 - Par la collecte de performance de la journée / weekend
 - Par la collecte d'un plan cache
 - Utilisation des outils QMGTOOLS pour la collecte d'informations de l'environnement de l'IBM i
 - Restitution de l'analyse
 - Discussion / recommandations par une session de visioconférence

- **Me contacter pour connaître les conditions financières.**
 - Cette prestation est éligible au programme IBM i vouchers : fournir les numéros de série pour vérifier si vos systèmes ont des vouchers.

Workshop iDoctor for IBM i

➤ **La prestation de service**

- ✓ Introduction iDoctor
 - ✓ Historique, composants, installation, bases de l'analyse de performance
- ✓ Utilisation des outils Collection Service Investigator (CSI) et Job Watcher
 - ✓ Préparation de l'analyse, concepts de base, analyse de la situation
 - ✓ Mise en situation sur votre partition
- ✓ Utiliser un SQL plan cache et l'analyser à l'aide de Plan Cache Analyzer
 - ✓ Recherche de requêtes SQL consommatrices de temps, de CPU, etc...

➤ **Me contacter pour connaître les conditions financières.**

- Cette prestation est éligible au programme IBM i vouchers : fournir les numéros de série pour vérifier si vos systèmes ont des vouchers.



IBM i

Soyez proactif et non réactif

MEREC