Livre blanc – Mesure des performances sous Windows Embedded Standard 7

Table des matières

Résumé

La boîte à outils Windows Performance Analysis est un puissant outil de mesure des performances et de diagnostic pour Windows Embedded Standard 7 (WE Standard 7). Il vous permet de capturer et d'afficher les informations de performance pour pratiquement n'importe quel élément du système, y compris vos propres applications.

Introduction

Les performances constituent un aspect crucial de tous les systèmes embarqués. Quel que soit le système d'exploitation utilisé, le niveau de performance dépend essentiellement de la conception. Une étape importante du processus de conception consiste donc à identifier, à l'aide des outils disponibles, les problèmes de performances et les moyens de les améliorer. Très souvent, les développeurs doivent établir manuellement des moyens de mesurer les performances du système d'exploitation et des programmes, et écrire des outils pour collecter et analyser les résultats. Heureusement, Windows Embedded Standard 7 s'appuie sur Windows 7 et peut donc tirer parti des puissants instruments et outils gratuits créés pour ce système.

Améliorer les performances d'un système embarqué exécutant Windows Embedded Standard 7 suppose trois grandes étapes. La première consiste à identifier les scénarii importants du point de vue des performances. La seconde correspond à la mesure de l'activité du système pendant l'exécution de ces scénarii. Enfin, la troisième consiste à analyser les résultats pour identifier les moyens d'améliorer les performances.

Windows Embedded Standard 7 comprend de puissants nouveaux outils pour vous aider à mesurer et à identifier les problèmes de performance. Les dépendances requises pour exécuter ces outils sont intégrées au Windows Embedded Foundation Core (chaque image Windows Embedded Standard 7 contient ce Foundation Core) et vous n'avez donc aucune opération particulière à effectuer pour les exécuter (à une exception près, décrite en détail dans la section « Mesure des performances de démarrage »).

Utilisation de la boîte à outils Windows Performance Analysis

L'outil vedette pour mesurer les performances dans le code base de Windows 7 est la Boîte à outils Windows Performance Analysis (WPA). Fourni avec le kit de développement logiciel (SDK) de Windows 7, la boîte à outils WPA vous permet de mesurer la plupart des aspects du fonctionnement de votre système. Pour obtenir cet outil, vous n'avez pas besoin de télécharger l'ensemble du SDK : vous pouvez le sélectionner seul pour accélérer le téléchargement.

Le SDK Windows 7 est disponible à cette adresse :

http://www.microsoft.com/downloads/details.aspx?FamilyID=c17ba869-9671-4330-a63e-1fd44e0e2505&displaylang=en

La boîte à outils WPA est incluse dans l'option « Outils de développement » du SDK Windows 7.

Fonctionnement

Dans Windows Embedded Standard 7, les événements sont signalés comme événements ETW (Event Tracing for Windows, suivi des événements pour Windows). Ces événements proviennent de centaines de points différents, l'un des principaux étant le noyau lui-même. Les événements de lecture/écriture sur disque, de changement de tâche, d'entrée ISR (sous-routine d'interruption), d'appels de procédure différés (DPC) sont quelques exemples parmi bien d'autres. Ces événements sont recueillis dans des journaux des traces d'événements avant d'être analysés. La boîte à outils WPA contient des outils permettant de capturer et visualiser ces traces.

Il existe essentiellement deux exécutables pour capturer et afficher les traces. Le premier, xperf.exe, permet de capturer les traces d'exécution et de lancer l'interface graphique utilisateur sur les traces déjà capturées. Le second, xbootmgr.exe, capture également les traces pour les transitions d'état (démarrage, arrêt, mise en veille prolongée, veille et reprise), mais il ne permet pas de les afficher. Nous allons d'abord nous intéresser aux traces d'exécution capturées avec xperf.exe.

Mesure des performances d'exécution

Après avoir installé la boîte à outils WPA, vous pouvez l'exécuter pour capturer une trace. Nous allons d'abord exécuter une trace de base utilisée pour identifier les performances générales avec xperf.exe. Pour capturer une trace, saisissez ce qui suit dans une invite de commande avec élévation de privilèges :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf –on DiagEasy

À présent, exécutez les scénarii pour lesquels vous cherchez à obtenir des informations.

Lorsque l'exécution des scénarii est terminée, arrêtez la trace et fusionnez toutes les données dans un fichier « Journal des traces d'événements ». Pour cela, saisissez ce qui suit dans une invite de commande avec élévation de privilèges :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf –d MaTrace.etl

L'application xperf.exe fusionne toutes les données recueillies dans le fichier MaTrace.etl, qui regroupe dans un format binaire tous les événements suivis. Notez que pour des exécutions plus longues, ces fichiers peuvent devenir extrêmement volumineux.

Pour visualiser les données, il vous suffit de taper :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf MaTrace.etl

L'interface utilisateur permettant de visualiser graphiquement les données s'ouvre. La Figure 1 ci-après montre un exemple de fichier-journal des traces d'événements dans l'outil xperfviewer. Sur cette vue, nous voyons s'afficher l'utilisation du processeur par UC (CPU Usage by CPU), l'utilisation du processeur par processus (CPU Usage by Process) et l'utilisation du processeur par thread (CPU Usage by Thread). De nombreuses autres vues s'affichent sous la vue illustrée. Pour sélectionner les vues à afficher, il suffit de cliquer sur la flèche de sélection, à gauche de l'écran.

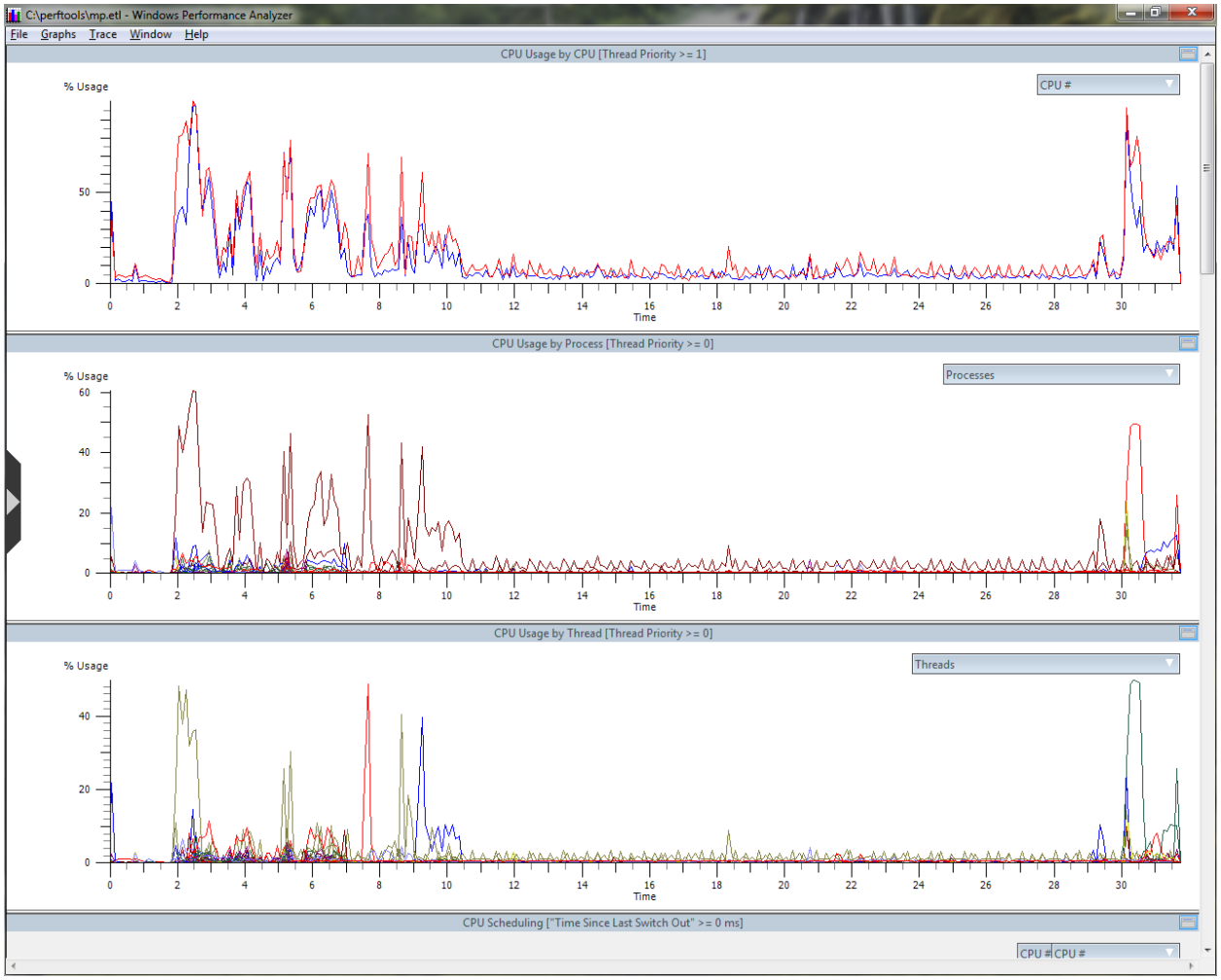


Figure 1 Affichage d'un journal des traces d'événements

Outre les graphiques, vous pouvez afficher des informations complémentaires en sélectionnant une période (cliquez et faites glisser le curseur pour sélectionner la période). Faites ensuite un clic droit et sélectionnez « Summary Table » (Table de résumé) ou « Detail Graph » (Détails du graphique), selon le graphique examiné.

La Figure 2 ci-après indique comment sélectionner les graphiques à afficher. En cliquant sur la flèche de sélection située à mi-hauteur sur le côté gauche de l'écran, vous pouvez choisir les graphiques à visualiser et à comparer.

Vous trouverez plus d'informations sur la capture de traces d'exécution et l'utilisation de l'interface graphique dans le fichier d'aide de la boîte à outils WPA.

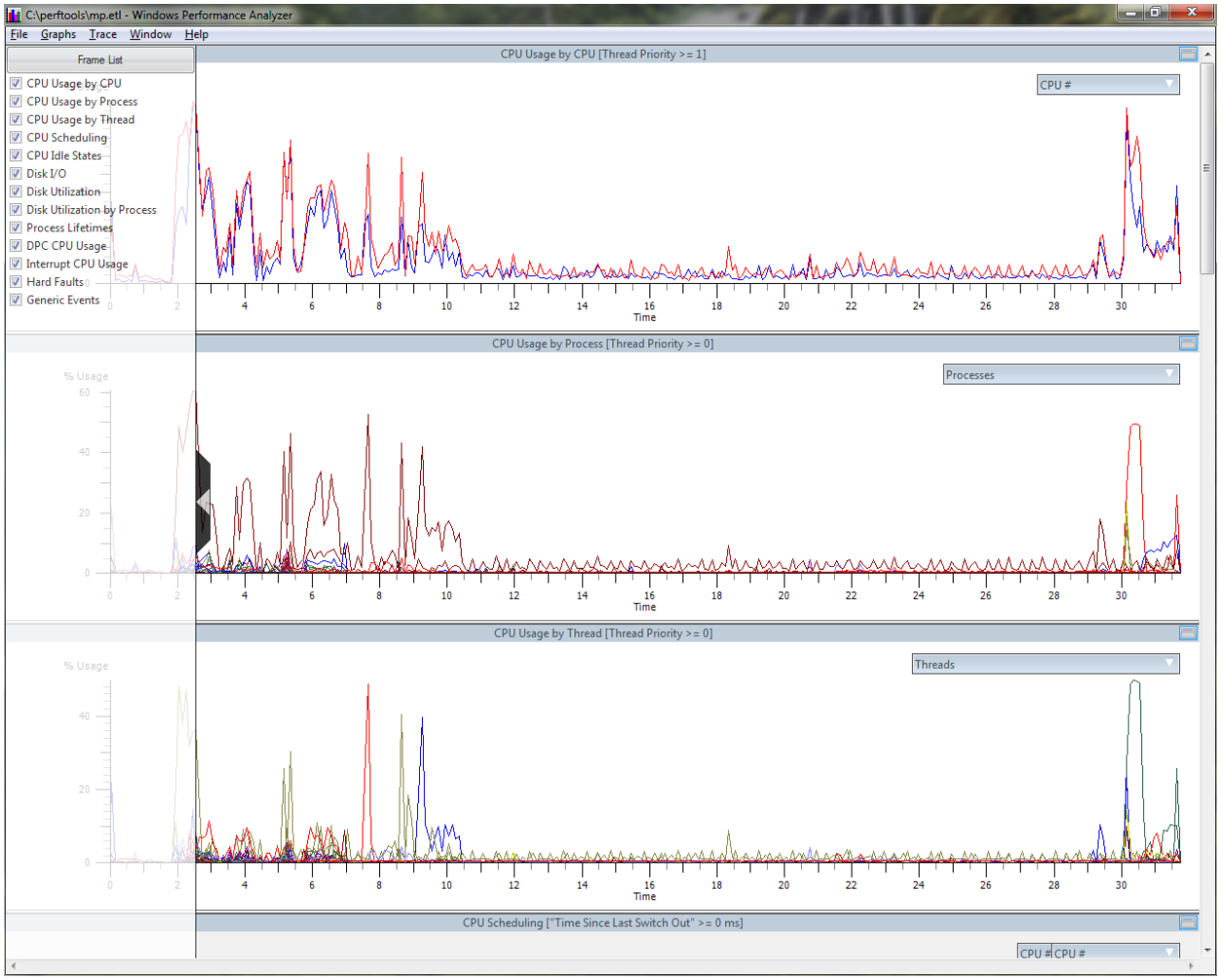


Figure 2 Sélection des graphiques à afficher

Mesure des performances lors des transitions d'état

Outre xperf.exe pour la mesure des performances d'exécution, vous pouvez utiliser xbootmgr.exe pour mesurer les performances de démarrage, arrêt, interruption, mise en veille prolongée et reprise.

Dans l'ensemble, la procédure est la même que pour xperf. À l'invite de commande avec élévation de privilèges, saisissez la commande suivante.

C:\Program Files\Microsoft Windows Performance Toolkit\xbootmgr –trace boot –numRuns 3 –resultPath c:\myBootTraces

Le programme xbootmgr.exe signale qu'il va fermer le système. Ensuite, il redémarre trois fois (-numRuns 3) et, à chaque fois, recueille les informations de suivi sur le processus de démarrage. L'action « boot » permet ensuite de rechercher dans ces informations les données de démarrage qui vous intéressent. Cette action « boot » analyse le fichier de trace .etl et recherche les données qui concernent

le processus de démarrage, notamment les temps d'E/S, les temps de chargement des processus et plus globalement l'intervalle entre les démarrages. Il inclut également plusieurs mesures de démarrage essentielles, notamment BootDoneViaPostExplorer qui correspond à la durée totale, en millisecondes, avant que système ne soit utilisable, à laquelle on ajoute dix secondes. Ces dix secondes correspondent à dix secondes de temps d'inactivité cumulées, à soustraire pour trouver le moment à partir duquel l'utilisateur aurait pu lancer une application ou utiliser le système d'une façon ou d'une autre. Cette durée est définie en surveillant les ressources clés du système et en établissant à quel moment le système est devenu suffisamment inactif pour que l'utilisateur puisse réellement commencer à utiliser l'ordinateur. Voici comment appliquer l'action « boot » à un fichier de trace .etl capturé avec la commande indiquée précédemment :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf –i c:\myBootTraces\boot_BASE_CSWITCH_1.etl –o boot1.xml –a boot

Le fichier boot.xml obtenu contient les informations de performance relatives au processus de démarrage. La Figure 3 en donne un exemple :

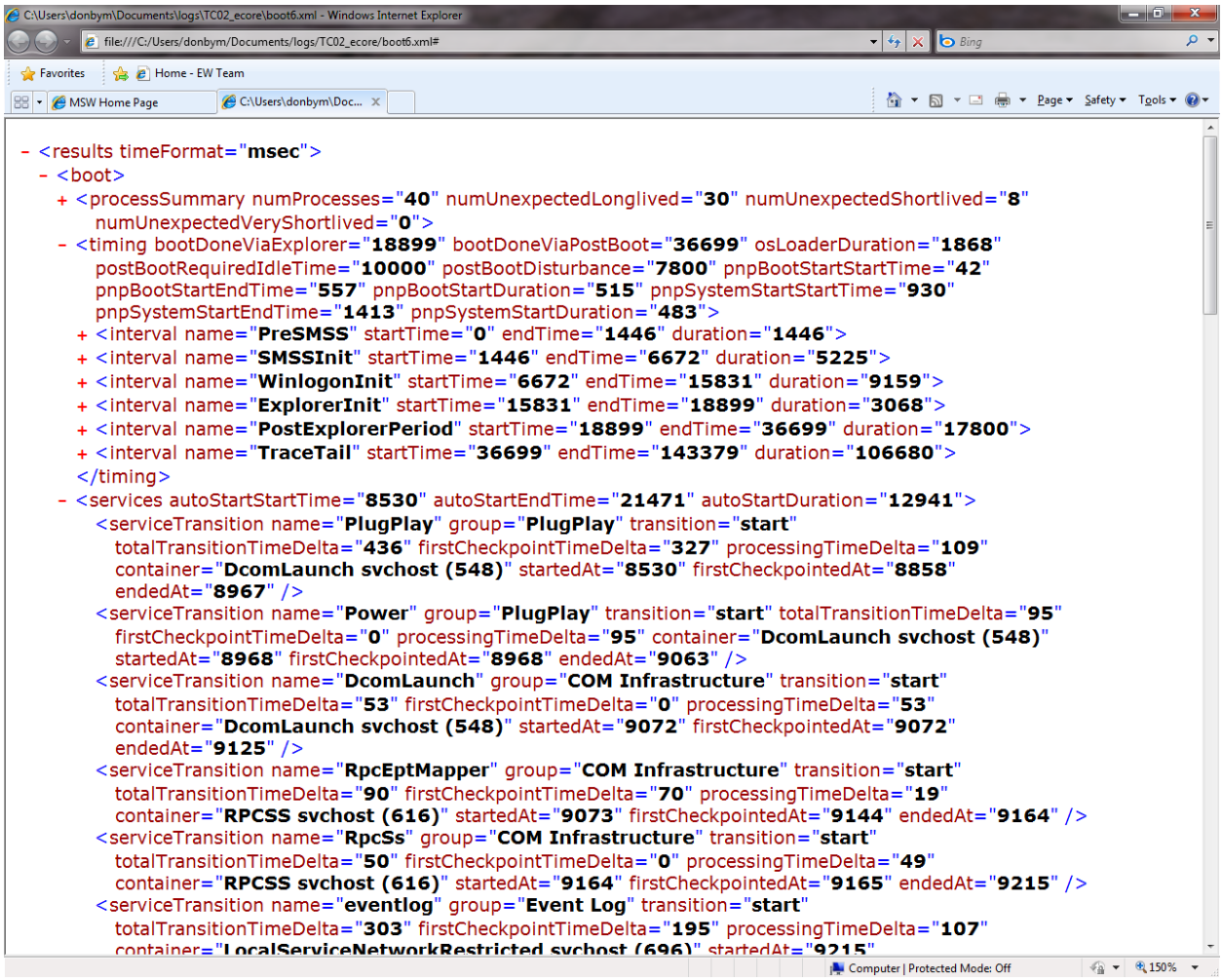


Figure 3 Fichier XML d'action Boot

Dans cet exemple, le système a été effectivement démarré en « bootDoneViaPostBoot – 10000ms », soit 26,699 secondes. Vous pouvez afficher diverses autres informations, notamment les services lancés, le temps de démarrage de chaque service, les événements Plug and Play, ainsi que la durée et l'activité E/S de chaque intervalle du processus de démarrage.

Vous pouvez également expérimenter d'autres actions. Pour afficher une liste complète des actions applicables aux différents types de traces :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf –help processing

Exemple : Analyse de l'utilisation du processeur

L'une des mesures de performance les plus courantes concerne l'utilisation du processeur. Chaque système embarqué comporte un processeur qui doit réaliser les tâches au moment approprié. Dans cet exemple, j'examinerai l'utilisation du processeur pendant la lecture d'un fichier audio dans l'application Zune Client. Je souhaite en effet connaître l'impact de cette tâche sur mon système en termes d'utilisation du processeur. Pour analyser ce scénario, je dois capturer une trace des performances pendant la lecture du fichier audio dans zune.exe et la saisie de ce livre blanc.

Microsoft Office 2007 Word est en cours d'exécution et l'application Microsoft Zune Client lit une piste audio. En même temps, je saisis les informations suivantes dans l'invite de commande à élévation de privilèges :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf –on DiagEasy

Mon scénario consiste à saisir cette phrase, après quoi je tape la commande suivante dans l'invite de commande :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf –d zune.etl

À présent, j'ai un fichier appelé zune.etl que je peux ouvrir dans l'application xperf.exe :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf zune.etl

Puisque je suis surtout intéressé par les mesures relatives au processeur, je déplace la barre de sélection sur la gauche de l'écran et je désélectionne tout, à l'exception de « CPU Usage by CPU » (Utilisation du processeur par UC), « CPU Usage by Process » (Utilisation du processeur par processus), « CPU Usage by Thread » (Utilisation du processeur par thread) et « Interrupt CPU Usage » (Utilisation du processeur à l'interruption). Le résultat est indiqué à la Figure 4 ci-après.

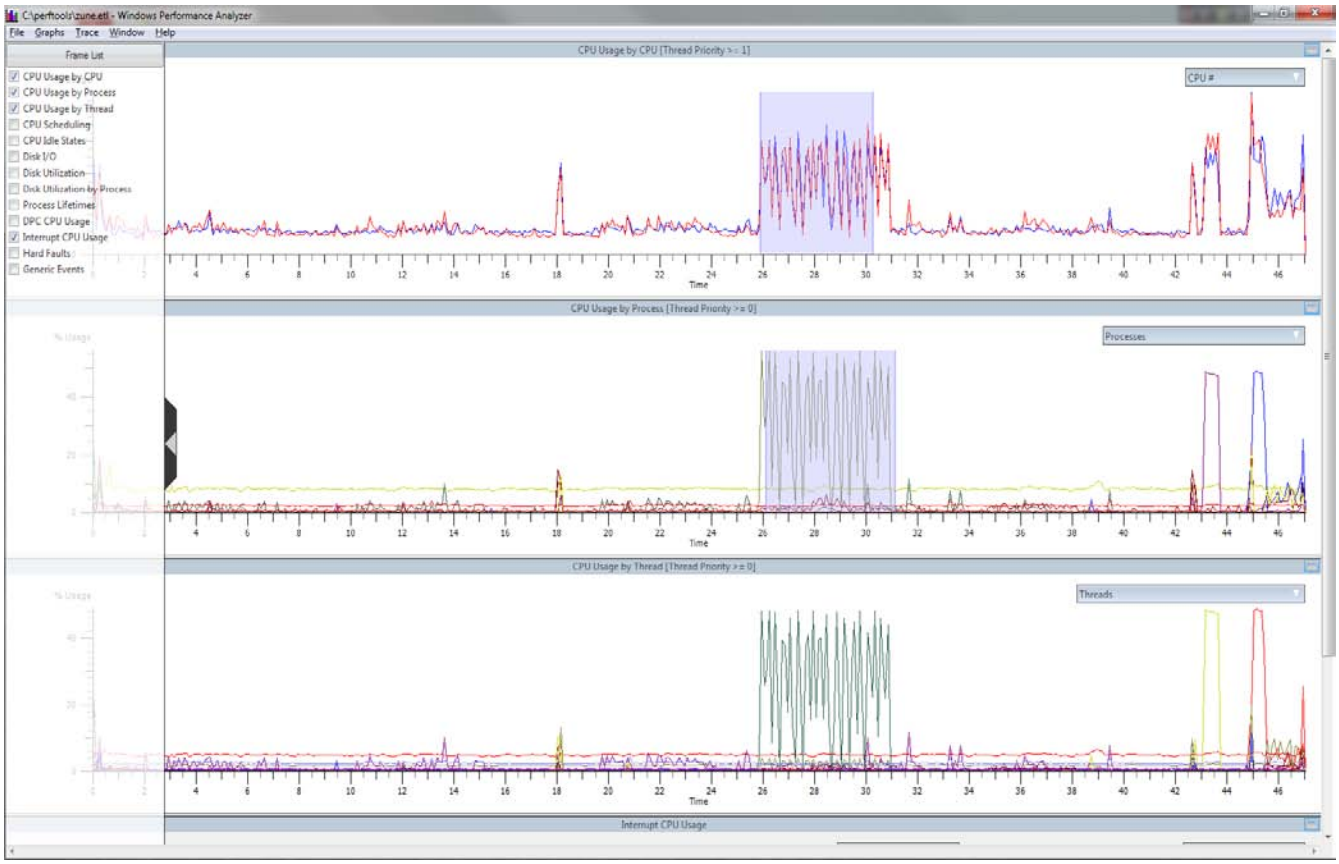
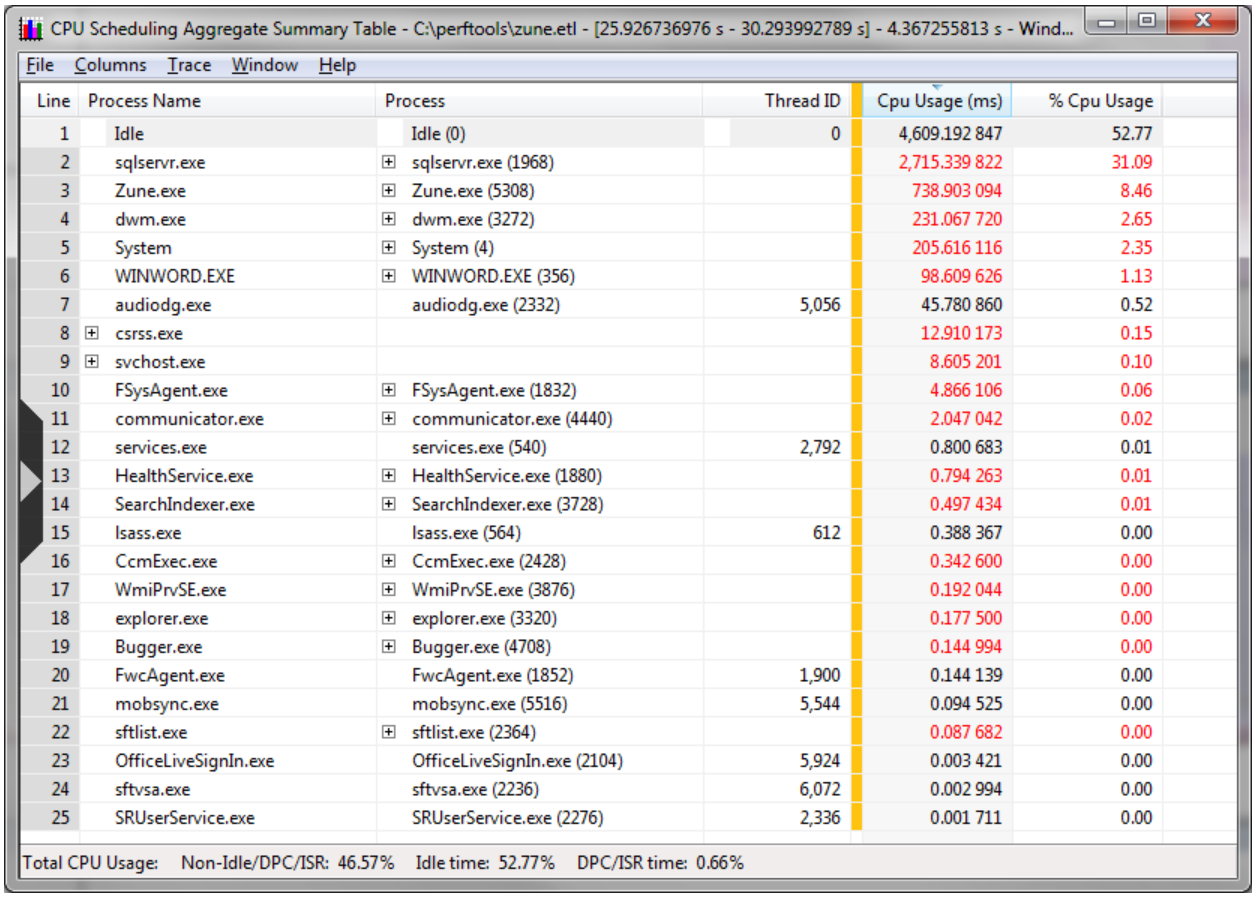


Figure 4 Exemple de vue de traces

Comme vous le voyez sur la période en surbrillance, le processeur semble très actif 26 secondes après le début de la trace. Je mets donc cette période en surbrillance dans le graphique « CPU Usage by CPU », je fais un clic droit sur la sélection et je choisis « Summary Table » (Table de résumé). Une nouvelle fenêtre apparaît avec la table de résumé. Elle est illustrée à la Figure 5 ci-après.



Line	Process Name	Process	Thread ID	Cpu Usage (ms)	% Cpu Usage
1	Idle	Idle (0)	0	4,609.192 847	52.77
2	sqlservr.exe	sqlservr.exe (1968)		2,715.339 822	31.09
3	Zune.exe	Zune.exe (5308)		738.903 094	8.46
4	dwm.exe	dwm.exe (3272)		231.067 720	2.65
5	System	System (4)		205.616 116	2.35
6	WINWORD.EXE	WINWORD.EXE (356)		98.609 626	1.13
7	audiodg.exe	audiodg.exe (2332)	5,056	45.780 860	0.52
8	csrss.exe			12.910 173	0.15
9	svchost.exe			8.605 201	0.10
10	FSysAgent.exe	FSysAgent.exe (1832)		4.866 106	0.06
11	communicator.exe	communicator.exe (4440)		2.047 042	0.02
12	services.exe	services.exe (540)	2,792	0.800 683	0.01
13	HealthService.exe	HealthService.exe (1880)		0.794 263	0.01
14	SearchIndexer.exe	SearchIndexer.exe (3728)		0.497 434	0.01
15	lsass.exe	lsass.exe (564)	612	0.388 367	0.00
16	CcmExec.exe	CcmExec.exe (2428)		0.342 600	0.00
17	WmiPrvSE.exe	WmiPrvSE.exe (3876)		0.192 044	0.00
18	explorer.exe	explorer.exe (3320)		0.177 500	0.00
19	Bugger.exe	Bugger.exe (4708)		0.144 994	0.00
20	FwcAgent.exe	FwcAgent.exe (1852)	1,900	0.144 139	0.00
21	mobsync.exe	mobsync.exe (5516)	5,544	0.094 525	0.00
22	sftlist.exe	sftlist.exe (2364)		0.087 682	0.00
23	OfficeLiveSignIn.exe	OfficeLiveSignIn.exe (2104)	5,924	0.003 421	0.00
24	sftvsa.exe	sftvsa.exe (2236)	6,072	0.002 994	0.00
25	SRUserService.exe	SRUserService.exe (2276)	2,336	0.001 711	0.00

Figure 5 Exemple d'utilisation du processeur

Dans cette table, la ligne 2 montre que, pendant cette période, sqlservr.exe semble effectuer une opération qui demande beaucoup de travail au processeur. À partir de là, vous pouvez rechercher pourquoi sqlservr.exe s'exécute et, éventuellement, réduire l'utilisation du processeur pendant ce scénario. Pour un système embarqué fonctionnant sur batterie, cette possibilité peut être particulièrement utile. La ligne 3 montre qu'à d'autres moments du suivi, l'application zune.exe est le principal utilisateur du processeur, qu'il occupe à 8,3 % environ, avec une marge d'inactivité importante (52,8 %).

Je peux non seulement afficher la trace dans l'interface utilisateur, mais également « agir » sur cette trace pour traiter les données et obtenir des mesures importantes. Par exemple, je peux exécuter l'action « cpudisk » pour obtenir un résumé, dans un fichier texte, de l'activité du processeur et du disque pour la durée du suivi. Pour cela, j'exécute la commande suivante :

C:\Program Files\Microsoft Windows Performance Toolkit\xperf –i zune.etl –o zunecpudisk.txt –a cpudisk

Le fichier zunecpudisk.txt est créé. Mon fichier ressemble à celui de la Figure 6 ci-après :

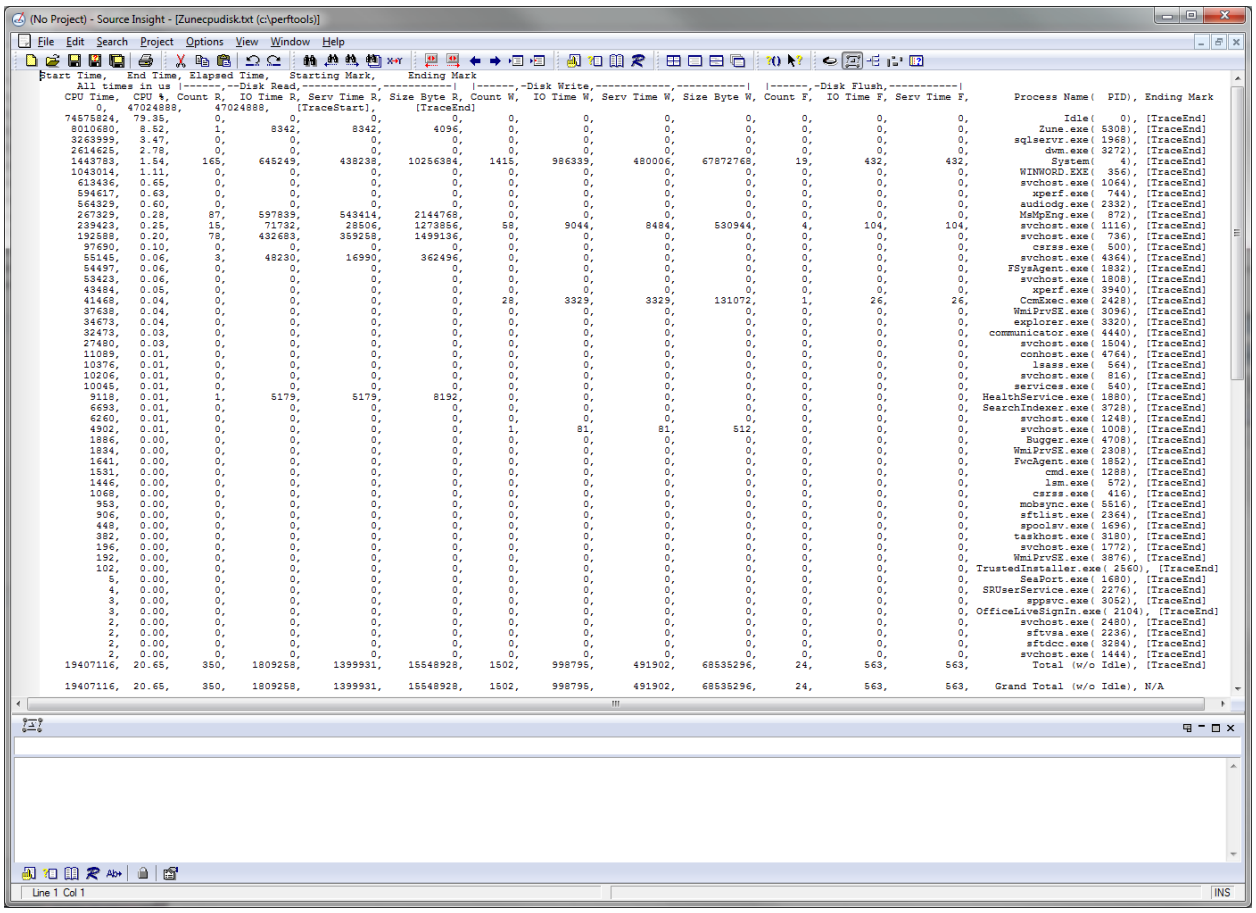


Figure 6 Action Processeur et disque

En zoomant sur la figure 6, vous verrez que le fichier de trace contient les chiffres moyens et totaux concernant l'activité du processeur et du disque. Par exemple, globalement, l'application zune.exe a utilisé 8,52 % du temps processeur, et celui-ci est resté inactif 79,35 % du temps.

Conclusion

Les systèmes et événements nécessaires pour utiliser la boîte à outils Windows Performance Analysis sont inclus dans Windows Embedded Standard 7 et permettent aux développeurs de réaliser des analyses de performances puissantes sur leurs systèmes embarqués. Vous trouverez la boîte à outils WPA dans le SDK Windows 7 client ici.

Les besoins et les solutions varient d'un projet de système embarqué à l'autre, mais de bonnes performances sont importantes quel que soit le système embarqué. Ces outils doivent permettre aux développeurs d'obtenir les meilleures performances possibles sur leur appareil Windows Embedded Standard 7.