

Quality of Services

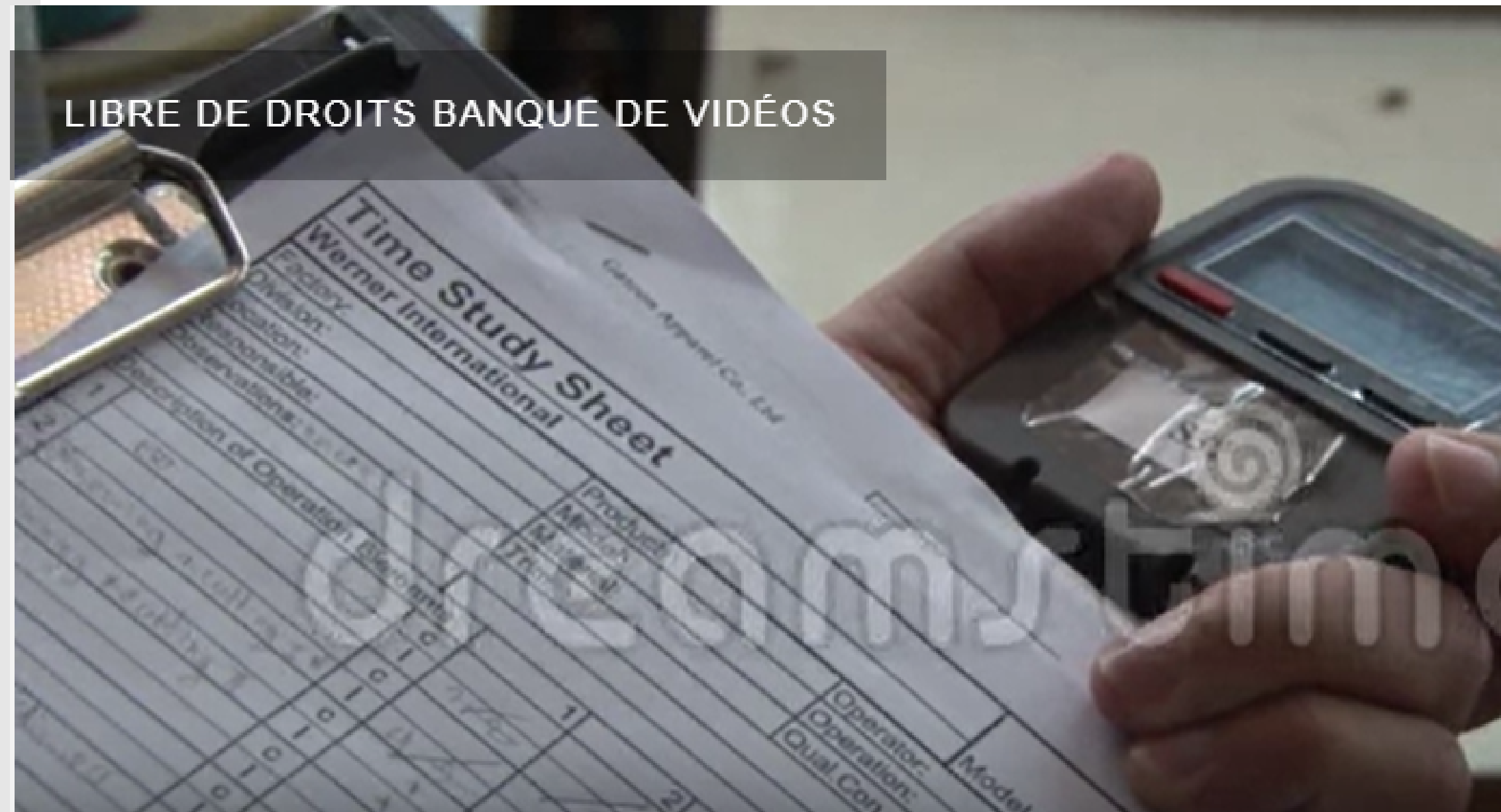
AYMAN BEN AMARA
ALEXANDRU PERJU



Avant

CHRONOMÉTRAGE MANUEL

LIBRE DE DROITS BANQUE DE VIDÉOS



- **Mesure de la latence et des erreurs** : Le système surveille la latence des opérations ainsi que le nombre d'erreurs par client et par opération.
- **Alertes en cas de dépassement des contraintes** : Lorsque les opérations dépassent les contraintes définies, des alertes doivent être déclenchées immédiatement.
- Composants de surveillance :
- **AverageLatencyMonitor** : Calcul des statistiques de latence par client et par opération pour les 100 derniers événements.
- **DynaLatencySpikeMonitor** : Détection des pics de latence en utilisant des limites définies dynamiquement.



- **ErrorRateMonitor** : Surveillance du taux d'erreur au cours des 10 dernières minutes en utilisant l'opérateur de minuterie "at" dans un modèle d'événement.
- **ServiceHealthMonitor** : Alertes en cas de survenance de 3 erreurs.
- **SpikeAndErrorMonitor** : Alertes en cas de dépassement d'une latence fixe ou de signalement d'un état d'erreur.
- Utilisation des modèles d'événements : Les modèles d'événements sont utilisés pour définir des règles de surveillance et de déclenchement d'alertes en fonction des données reçues.
- **Extraction de données** : Plutôt que de simplement s'abonner aux événements publiés, ce système extrait activement des données des instructions pour l'analyse et la prise de décision.



La qualité de service (QoS) est un ensemble d'outils et de techniques utilisés pour gérer et améliorer les performances d'un réseau. Il est utilisé pour hiérarchiser le trafic, allouer les ressources et garantir que les paquets de données sont livrés en temps opportun. La QoS peut être utilisée pour améliorer les performances d'applications telles que la VoIP, le streaming vidéo et les jeux en ligne.



Lancement requête

Observation: Nous observons qu'en paramétrant la requête avec 3 opérations(S1,S2,S3) pour 3 clients C1,C2,C3 en lui donnant des latences à respecter pour chacun d'entre eux le programme il nous montre à différents instances quelle était la latence pour le client en faisant une certaine opération. On observe que la latence n'est pas toujours la même mais elle ne dépasse jamais le seuil imposé.

Spike Monitor : Ce code met en place un moniteur de pics de latence dynamiques. Lorsqu'un nouvel événement de limite de latence est reçu, il crée dynamiquement deux requêtes EPL pour surveiller les pics de latence et arrêter la surveillance lorsque la limite de latence change.

Ce code installe un moniteur de pics de latence dynamiques. Lorsqu'un événement de limite de latence est reçu, il crée deux requêtes EPL pour surveiller les pics de latence et arrêter la surveillance lorsque la limite de latence change.

MonitorUTIL : Récupère l'IdCustomer et Le OperationName des 100 dernires résultats

Merci
