TimesFound può rientrare in questo CalcolaConteggioUtilizzo, ciè oltre che la complessità della stringa potrebbe essere intyeresante conoscere il tempo, cioè a parità di complòessità il tempèo fa la differenza

La tua osservazione è molto acuta e tocca un punto cruciale per l'analisi avanzata del sistema. È un'ottima idea considerare il **tempo di scoperta** di una stringa, oltre alla sua complessità.

Tuttavia, TimesFound ha già una sua funzione specifica: conteggia il numero di volte in cui una data stringa MIU è stata prodotta durante l'esplorazione. È una metrica di **frequenza**, non di tempo.

Per incorporare la tua idea, è più pulito e corretto non alterare la funzione di CalcolaConteggioUtilizzo. Questo metodo dovrebbe rimanere focalizzato sulla complessità intrinseca della stringa (basata solo sui suoi caratteri o pattern).

Invece, ti suggerisco di introdurre una **nuova metrica temporale** che si affianchi alle altre.

**Proposta di Soluzione**

1. **Aggiungi una nuova proprietà** alla classe MIUStateHistoryDb. Potremmo chiamarla TimeToFirstDiscovery\_ms, che memorizza il tempo trascorso, in millisecondi, dall'inizio dell'esplorazione fino alla prima volta che quella stringa è stata trovata.
2. **Modifica il metodo UpsertMIUStateHistory(MIUStateHistoryDb state)**: Quando un nuovo stato viene inserito (isNewState è true), aggiorna il database con questo nuovo valore.
3. **Calcola il valore nel punto di generazione**: Come hai notato, la variabile stopwatch.ElapsedMilliseconds è già disponibile nel tuo codice. Questo è il valore che dovrai assegnare al nuovo campo TimeToFirstDiscovery\_ms quando crei l'oggetto MIUStateHistoryDb.

**Esempio di Implementazione**

**Aggiungi il campo a MIUStateHistoryDb**

C#

public class MIUStateHistoryDb

{

// ... altri campi esistenti

public int UsageCount { get; set; }

public string DetectedPatternHashes\_SCSV { get; set; }

public long TimeToFirstDiscovery\_ms { get; set; } // Nuovo campo

}

**Modifica il metodo di persistenza InternalUpsertMIUState**

Assicurati che l'istruzione UPDATE salvi anche il valore di UsageCount e DetectedPatternHashes\_SCSV, e che l'istruzione INSERT salvi il nuovo campo TimeToFirstDiscovery\_ms.

C#

private Tuple<long, bool> InternalUpsertMIUState(MIUStateHistoryDb state)

{

// ...

// Se lo stato è nuovo, esegui l'INSERT

if (result == null)

{

// ... (resto dell'INSERT)

insertCmd.Parameters.AddWithValue("@usageCount", state.UsageCount);

insertCmd.Parameters.AddWithValue("@detectedPatternHashes", state.DetectedPatternHashes\_SCSV);

insertCmd.Parameters.AddWithValue("@timeToFirstDiscoveryMs", state.TimeToFirstDiscovery\_ms); // Nuovo parametro

// ...

}

// ...

}

**Modifica il codice in TrovaDerivazioneDFS**

Infine, quando crei il nuovo oggetto newState, assegna il valore del tempo trascorso.

C#

if (applySuccess)

{

// ...

var newState = new MIUStateHistoryDb

{

// ... altri campi

UsageCount = CalcolaConteggioUtilizzo(newStringStandard),

DetectedPatternHashes\_SCSV = TrovaPatternRilevati(newStringStandard),

TimeToFirstDiscovery\_ms = stopwatch.ElapsedMilliseconds // Assegna il tempo trascorso

};

Tuple<long, bool> upsertResult = dataManager.UpsertMIUStateHistory(newState);

// ...

}

In questo modo, mantieni ogni metrica separata e logica, permettendoti di analizzare il tempo di scoperta in relazione alla complessità della stringa.

dovrei aggiungere un campo alla tabella MIU\_States\_History?

Sì, è **assolutamente necessario** aggiungere un campo alla tabella MIU\_States\_History per salvare il tempo di prima scoperta di una stringa.

L'aggiunta del campo TimeToFirstDiscovery\_ms è fondamentale per implementare correttamente la tua idea. Se non lo aggiungi, non avrai un posto dove memorizzare il valore calcolato da stopwatch.ElapsedMilliseconds e l'analisi temporale che desideri non sarà possibile.

Ecco i passaggi che devi seguire per l'aggiornamento del tuo schema di database:

1. **Modifica la classe MIUStateHistoryDb**: Hai già fatto questo nel passaggio precedente. Il nuovo campo public long TimeToFirstDiscovery\_ms { get; set; } deve essere presente nella tua classe.
2. **Aggiungi il campo alla tabella del database**: Dovrai scrivere e eseguire una migrazione o uno script SQL per aggiungere la nuova colonna alla tabella MIU\_States\_History. L'istruzione SQL sarà simile a questa:

SQL

ALTER TABLE MIU\_States\_History

ADD COLUMN TimeToFirstDiscovery\_ms INTEGER;

Questa operazione aggiungerà la colonna TimeToFirstDiscovery\_ms di tipo intero (in SQLite) alla tua tabella esistente.

1. **Aggiorna le query nel InternalUpsertMIUState**: Devi modificare sia la query INSERT che la query UPDATE nel tuo metodo privato InternalUpsertMIUState per includere il nuovo campo.
   * **INSERT**: La query di inserimento deve specificare il nuovo campo e il relativo parametro.
   * **UPDATE**: La query di aggiornamento **non** deve modificare questo campo. Il tempo di prima scoperta è un valore che viene fissato solo al momento della prima apparizione della stringa, quindi va inserito solo con l'istruzione INSERT.

Implementando questi passaggi, il tuo database sarà pronto a gestire la nuova metrica temporale.