Ristrutturazione ER

1. Eliminazione attributi **multivalore**
   1. Non ne sono presenti all’interno dell’ER
2. Eliminazione delle gerarchie di generalizzazione
   1. L’accorpamento della gerarchia “Metric” conviene farlo verso il padre, seppur ci siano tutti i presupposti per fare accorpamento verso il basso (copertura della gerarchia **totale** ed **esclusiva**). Ma dato che gli accessi saranno molto probabilmente fatti da experiment instant verso Metric, allora a quel punto conviene in questo modo.
3. Partizionamento/accorpamento di entità
4. Scelta identificatori

Per prima cosa, vista che l’operazione è solo una si è valutata la fattibilità di essa sullo schema:

* Lettura di tutte le metriche associate ad un certo istante di eseuzione.
  + Input:
    - (tipo di nodo, {(#fa, wla), (#fb, wlb), (#fc, wlc), …})
  + Output:
    - Tutte le metriche associate

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Descrizione | Concetto | Costrutto | Accessi | Tipo [L-S] | Totale |
| Selezione del tipo di nodo | NODE | E | 1 | L | 1L |
| Navigazione “has” per tipo nodo | HAS | R | 140 | L | 140L |
| Navigazione “related to” | RELATED TO | R | 560 | L | 560L |
| Selezione dell’ID relativo all’expet instant la situaizone richiesta (wl e numero repliche) | EXPERIMENT INSTANT | R | 1 | L | 1L |
| Navigazione “gather” per gli experiment instant selezionati | GATHER | R | 8 | L | 8L |
|  |  |  |  |  | Tot: 710 L |

Note:

* HAS ha circa 140 record per ogni nodo, mediamente per ogni tipo di nodo ci sono 20 file, ognuno dei quali ha 7 istanti di esecuzione. 20x7=140.
* RELATED TO viene navigata dopo aver circa 140 record relativi alla navigazione della precedente relazione. Supponendo che nel caso peggiore siano deployate tutte e tre le funzioni, nel caso pessimo si attraverseranno 140x3=560 record
* GATHER viene navigata una volta selezionati gli ID degli esperimenti che hanno tutte le caratteristiche richieste (in termini di wl e in termini di numero di repliche). Supponendo che il numero delle metriche per nodo sia pari a 2 (RAM e CPU USAGE) e che il numero delle metriche per funzione sia circa 6; allora il numero di accessi sarà pari a 1x2+1x6 = 8

Traduzione da **schema concettuale** a **logico**:

NODE (ID, Name, Ram, Cpu)

FUNCTION (ID, Name, Description)

EXPERIMENT\_INSTANT (ID, Timestamp, NodeID)

* FOREIGN KEY: NodeID REFERENCES NODE

METRIC (ID, Type, Value, Unit, Description, ExpInstantID, NodeID\*, FunctionID\*)

* FOREIGN KEY: ExpInstantID REFERENCES EXPERIMENT\_INSTANT
* FOREIGN KEY: NodeID REFERENCES NODE
* FOREIGN KEY: FunctionID REFERENCES FUNCTIONS

DEPLOY (ExpInstantID, FunctionID, MaxRate, NumReplicas, Workload)

* FOREIGN KEY: ExpInstantID REFERENCES EXPERIMENT\_INSTANT
* FOREIGN KEY: FunctionID REFERENCES FUNCTIONS

**Note**:

1. Function forse ha più senso che abbia come ID il **nome**, così nella tabella pivot abbiamo già la chiave della funzione.

Query per avere gli ID degli istanti di esecuzione che hanno tutte e tre le funzioni deployate (funca, qrcode, ocr) e che hanno quel terminato wl

SELECT Exp\_Instant.ID

FROM Node JOIN Exp\_Instant JOIN Deploy JOIN Function

WHERE

( Node.ID == node\_id) AND

(

( Function.Name == ‘funca’ AND Deploy.Workload == wl-funca AND replicas)

OR

( Function.Name == ‘qrcode’ AND Deploy.Workload == wl-qrcode AND replicas)

OR

( Function.Name == ‘ocr’ AND Deploy.Workload == wl-ocr AND replicas )

)

GROUP BY Exp\_Instant.ID

HAVING COUNT(Deploy.FunctionID) == 3

**Spiegazione**:

* Viene fatto JOIN tra le 4 tabelle Node, Exp\_Instant, Deploy e Function in modo da avere record completi su cui eseguire il filtro.
* Viene applicato il filtro della clausola WHERE, filtrando per tipo di nodo, e numero di repliche per le funzioni desiderato con un certo WL.
* Sulla base di questi viene effettuato un raggruppamento per Exp\_Instant.ID in modo tale da selezionare con clausola HAVING solo quelli con COUNT(Deploy.FunctionID) == 3, ovvero che hanno tre funzioni deployate in quel momento, ovvero le tre funzioni che si sono filtrate inizialmente.

**Nota**: il risultato di tale query sarà da utilizzare come elemento nella clausola WHERE della select sulle metriche

**Problema**:

* Ho visto che c’era un problema riguardo al fatto che alcuni esperimenti su node\_1 sono stati fatti con max\_rate su qrcode = 2 e = 3.
* Quindi quando va a fare la query per una configurazione data da
  + (node\_type, [(funcName, wl, replicas), …, [(funcName, wl, replicas)]) potrebbero apparire più funzioni che però hanno max\_rate diversi per via di questi esperimenti, e quindi alcuni raggruppamenti non funzionare come sperato.
* Se si imposta il setup sperimentale e si setta un certo **max\_rate** x replica, allora tanto vale caricare i dati di un solo set di esperimenti in questa situazione
* Decidere come fare per generazione casi nella simulaione
  + Al momento prende solo casi presenti nel DB.
  + Andrebbe bene per ora, in quanto non si ha una componente che genera e simula esperimenti che non siano presenti all’interno del DB.
* Cosa succede con più configurazioni uguali che vengono ritornate insieme?
  + Si utilizza una media sui valori data dal raggruppamento.