**Sperimentazione xFail**

**Idea Generale**: Creare un testbed con Chaos Echo, con una topologia fissata (es. quella Sock Shop) e sfruttare la configurabilità delle probabilità per eseguire due tipi di esperimento

**Primo esperimento: Fallimento controllato**

*Obiettivo: Dimostrare che se c’è una causa, la troviamo*

a) Azzerare tutte le probabilità di fallimento (autonomo) di tutti i servizi, lasciando la possibilità che falliscano in seguito al fallimento dei servizi che invocano.

b) Configurare poi un servizio X in modo che fallisca autonomamente, in modo da generare cascate di fallimenti fino a un servizio Y (es. frontend).

c) Verificare che xFail identifica correttamente che la root cause per tutti i servizi falliti (inclusi X e Y) è X

*Note: si può variare X a turno e mettere come X tutti i servizi nella topologia.*

**Secondo esperimento: Precision/recall at top K**

*Obiettivo: Dimostrare che (anche se vengono restituite N>>K soluzioni) basta guardare le prime K soluzioni per trovare la spiegazione di un fallimento*

a) Configurare il testbed in modo che possano fallire tutti i servizi, sia autonomamente sia a causa di servizi che invocano

b) Analizzare i log “a mano” e marcare le reali cause di ciascun fallimento

c) Utilizzare xFail per spiegare i fallimenti identificati

In c, per ciascun fallimento possiamo avere più cause possibili identificate da xFail. Vogliamo dimostrare che la causa reale è sempre (o con buona probabilità) tra le prime K. Ad esempio, potremmo misurare:

* Quante volte la soluzione esatta è la prima
* Quante volte la soluzione esatta è tra le prime 2
* Quante volte la soluzione esatta è tra le prime 3
* …

In questo modo, potremmo mostrare un plot della “distribuzione di probabilità”: percentuale di successi con 1, 2, 3, … soluzioni.

*Note: usare precision/recall at top K?*

*Note: far variare anche probabilità di fallimento?*