

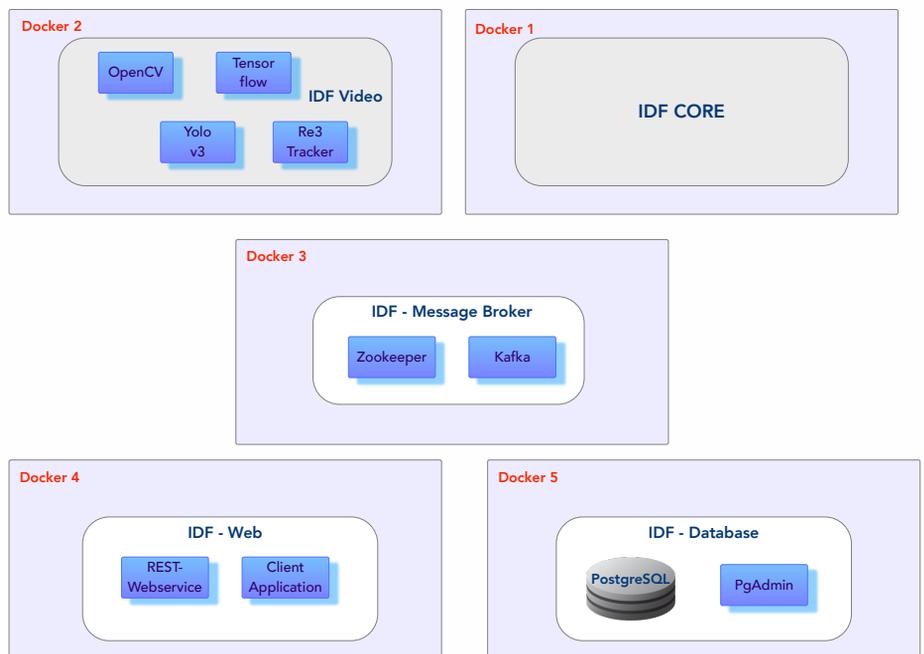
Il sistema IDF "Intelligent Data Fusion" SARA AI, è il modulo software della piattaforma IDF SARA dedicato alla rilevazione e analisi delle situazioni di intrusione basato sulle tecniche di intelligenza artificiale. IDF SARA AI fornisce informazioni sulle situazioni di allarme elaborando i dati di sistemi e sensori diversi. IDF è sviluppato su tecnologie "Open Source" e composto da più elementi, ciascuno dedicata a uno compito specifico.



Architettura:

- IDF AI è pensato come un sistema di tipo SOA (Service Object Architecture) e per poter operare su macchine di tipo diverso. Per questo le componenti del sistema sono sviluppate come "microservizi"
- Ogni componente IDF, costituisce un servizio indipendente, comunica con gli altri componenti via message-broker.
- I componenti IDF AI sono progettati per l'esecuzione all'interno di un "Docker container", ossia un contenitore virtualizzato che fornisce una astrazione del software dal sistema operativo della macchina ospitante. I diversi containers verranno orchestrati via "Docker Swarm/Compose"
- Le comunicazioni tra i vari componenti IDF avvengono attraverso un apposito message broker "Apache Kafka", un sistema di messaggistica istantanea ideale per lo scambio di dati tra microservizi, con elevato numero di operazioni in tempo reale e con un REST-Webservice «Flask» appositamente sviluppato

Il sistema IDF "Intelligent Data Fusion" SARA AI, è il modulo software della piattaforma IDF SARA dedicato alla rilevazione e analisi delle situazioni di intrusione basato sulle tecniche di intelligenza artificiale. IDF SARA AI fornisce informazioni sulle situazioni di allarme elaborando i dati di sistemi e sensori diversi. IDF è sviluppato su tecnologie "Open Source" e composto da più elementi, ciascuno dedicata a uno compito specifico.



IDF Core – Docker 1:

- a. Autonomia, microservizio autonomo in grado di avviarsi anche in caso di non disponibilità del web-service e/o del database
- b. Parametrizzazione " tempo di interazione " (tra eventi di una stessa area)
- c. Parametrizzazione " numero sensori " (presenti nell'area)
- d. Parametrizzazione "tipologie_sensori" con livelli di "attendibilità" e "gerarchia"
- e. Parametrizzazione "stato_sensori": "nome_stato" (attivo, disattivo, guasto)
- f. Parametrizzazione "only_detection": True/False per telecamere mobili (dome)
- g. Parametrizzazione area: per segnalazione "detection" e/o "movimento"
- h. Impostazione "schedulazione" di esclusione allarmi per: periodo, giorno della settimana e orario
- i. Impostazione "area di interesse" valutazione se l'evento si trova all'interno di un'area poligonale definita sulla scena
- j. Impostazione "bounding box min dimension" per valutazione della scena su tre fasce "near", "middle", "far" con dimensioni minime parametrizzate



Docker 2



Docker 1

IDF CORE

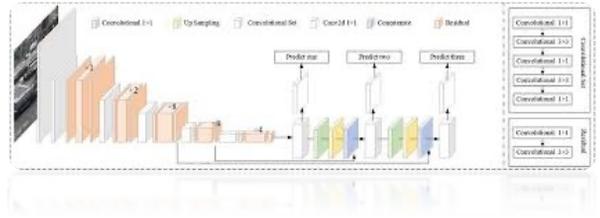
IDF Video – Docker 2:

- a. Autonomia, microservizio autonomo in grado di avviarsi anche in caso di non disponibilità del web-service e/o del database
- b. Algoritmo di AI con motion detection, object detection e tracking
- c. Compatibilità con tutti i tipi di telecamere esistenti con protocollo H264 e H265
- d. Parametrizzazione del flusso video (primario/secondario)
- e. Gestione della perdita di connettività con telecamera (riconnessione automatica)
- f. Parametrizzazione numero frame elaborati al secondo, su ogni flusso video separatamente
- g. Parametrizzazione indice di variazione dei bounding box, su ogni flusso video separatamente
- h. Parametrizzazione stationary time e moving time, su ogni flusso video separatamente
- i. Impostazione della percentuale di utilizzo della GPU
- j. Impostazione del numero di core CPU dedicati ad ogni singolo modulo idf-video
- k. Ottimizzazione della rappresentazione a video dei

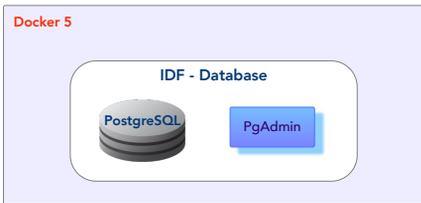


IDF System:

- a. Schedulazione backup database giornaliera parametrizzabile
- b. Schedulazione svecchiamento dati allarmi su database giornaliera parametrizzabile
- c. Svecchiamento automatico immagini dei rilevamenti giornaliero parametrizzabile
- d. Svecchiamento automatico dei file di log dei moduli IDF-Video e IDF-Core giornaliero parametrizzabile
- e. Comando di avvio/stop generale del sistema IDF
- f. Reinizio delle notifiche di evento in caso di mancata ricezione da parte del modulo IDF dedicato



PostgreSQL: The World's Most Advanced Open Source Relational Database



IDF Database Docker 5:

- a. Gestisce le configurazioni del sistema e i settaggi dei singoli componenti.
- b. Il motore del DB è "PostgreSQL" esso è un ORDB (object-relational database) con alta rispondenza agli standard SQL

IDF Network: Internal Network Docker 3

La comunicazione dei dati avviene attraverso gestori di messaggi (message-broker) su protocolli standard con garanzia di consegna. L'uso del gestore "Apache Kafka" permette di supportare elevati volumi secondo una logica publish/subscribe e permette di definire un tempo di persistenza dei messaggi. Questa rete di messaggi denominata IDF-Net è appositamente costruita in maniera isolata e criptata ricorrendo alla tecnologia Docker Swarm.



IDF Network: External Network – Docker 3

E' la rete di messaggi verso i sistemi esterni, principalmente per la segnalazione degli allarmi.

E' integrata con gli attuali servizi S.A.R.A. (XML-RPC). Essa prevede l'utilizzo di message-broker conformi al protocollo AMQP (Advanced Message Queuing Protocol) ad es. "RabbitMQ".



IDF WebService – Docker 4

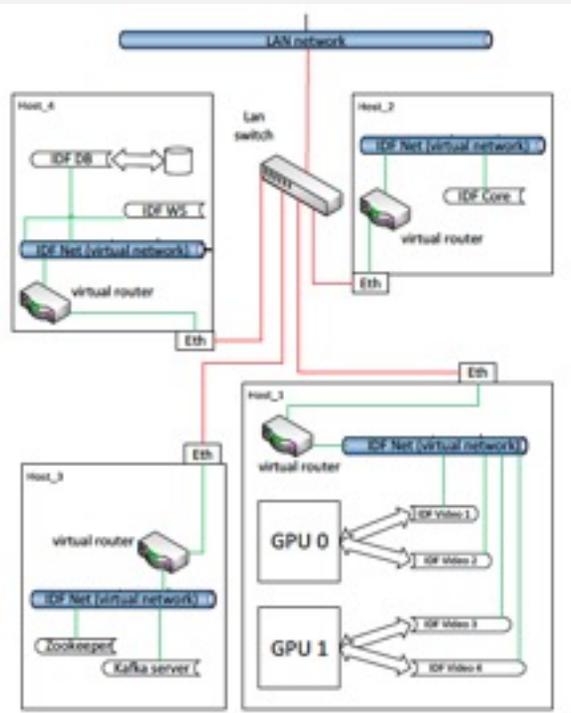
E' il componente che fornisce l'interfaccia delle funzioni di gestione CRUD dei dati verso il database. Si tratta di un RESTful webservice Flask. Flask permette l'ORM (Object Relational Mapping) la validazione e serializzazione dei dati.



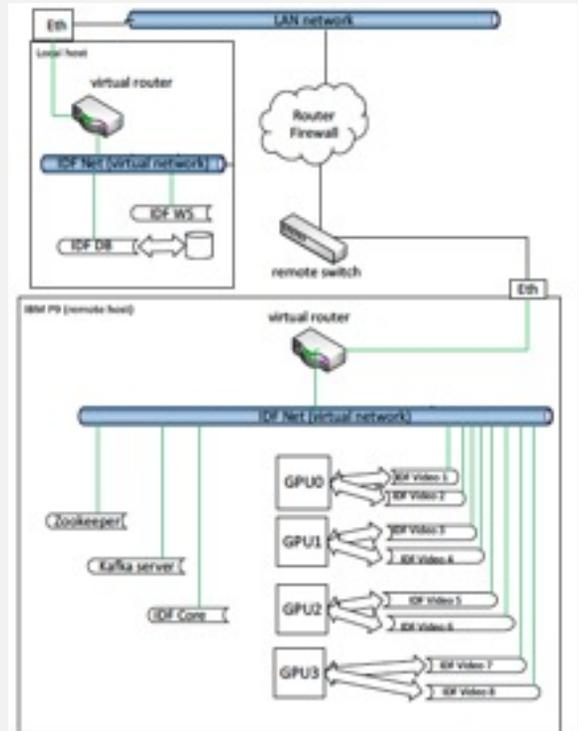
Scalabilità HW

Il sistema IDF permette inoltre completa scalabilità hardware poiché i vari componenti possono essere installati su una o più macchine linux diverse, ognuna con capacità hardware GPU/CPU/RAM diverse per rispondere ad ogni particolare esigenza.

Architettura IDF con tecnologia docker Intel



Architettura IDF con tecnologia docker (IBM P9)



Scalabilità SW

Il sistema IDF è basato su "moduli" all'interno di singoli "Docker Containers" questa architettura permette di massimizzare in modo efficiente tutti i "core" delle CPU disponibili assegnando a ciascuno il corretto set di "core" da utilizzare.

La scalabilità è applicata anche alle GPU suddividendo i moduli in modo efficiente tra tutte le GPU disponibili e assegnando a ciascuno la corretta percentuale di utilizzo della memoria

Numero flussi testati su diverse architetture:

- 2xGPU Nvidia Tesla v100 fino a 100 telecamere,
- 1xGPU Nvidia Quadro RTX6000 fino a 50 telecamere,
- 1xGPU Nvidia Xavier, fino a 8 telecamere,
- 1xGPU Nvidia Nano, fino a 4 telecamere





| ID Modulo | Nome Modulo | Descrizione | IP Address | Port | Host Name | IP Remote Host | Nome | Descrizione | IP Address | Port |
|-----------|---------------------------|-----------------|--------------|------|-----------------------|----------------|-----------------|-----------------|--------------|------|
| 80 | SP-Modulo LACASELLA, SP01 | SP-Modulo CPU02 | 192.168.1.10 | 8080 | SP-LACASELLA-Server01 | 192.168.1.10 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.10 | 8080 |
| 81 | SP-Modulo LACASELLA, SP02 | SP-Modulo CPU02 | 192.168.1.11 | 8080 | SP-LACASELLA-Server01 | 192.168.1.11 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.11 | 8080 |
| 82 | SP-Modulo LACASELLA, SP03 | SP-Modulo CPU02 | 192.168.1.12 | 8080 | SP-LACASELLA-Server01 | 192.168.1.12 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.12 | 8080 |
| 83 | SP-Modulo LACASELLA, SP04 | SP-Modulo CPU02 | 192.168.1.13 | 8080 | SP-LACASELLA-Server01 | 192.168.1.13 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.13 | 8080 |
| 84 | SP-Modulo LACASELLA, SP05 | SP-Modulo CPU02 | 192.168.1.14 | 8080 | SP-LACASELLA-Server01 | 192.168.1.14 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.14 | 8080 |
| 85 | SP-Modulo LACASELLA, SP06 | SP-Modulo CPU02 | 192.168.1.15 | 8080 | SP-LACASELLA-Server01 | 192.168.1.15 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.15 | 8080 |
| 86 | SP-Modulo LACASELLA, SP07 | SP-Modulo CPU02 | 192.168.1.16 | 8080 | SP-LACASELLA-Server01 | 192.168.1.16 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.16 | 8080 |
| 87 | SP-Modulo LACASELLA, SP08 | SP-Modulo CPU02 | 192.168.1.17 | 8080 | SP-LACASELLA-Server01 | 192.168.1.17 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.17 | 8080 |
| 88 | SP-Modulo LACASELLA, SP09 | SP-Modulo CPU02 | 192.168.1.18 | 8080 | SP-LACASELLA-Server01 | 192.168.1.18 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.18 | 8080 |
| 89 | SP-Modulo LACASELLA, SP10 | SP-Modulo CPU02 | 192.168.1.19 | 8080 | SP-LACASELLA-Server01 | 192.168.1.19 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.19 | 8080 |
| 90 | SP-Modulo LACASELLA, SP11 | SP-Modulo CPU02 | 192.168.1.20 | 8080 | SP-LACASELLA-Server01 | 192.168.1.20 | SP-Modulo CPU02 | SP-Modulo CPU02 | 192.168.1.20 | 8080 |

- IDF è pensato come un sistema di tipo SOA (Service Object Architecture) e per poter operare su macchine di tipo diverso. Per questo le componenti del sistema, chiamate "MODULI IDF", sono sviluppate come "microservizi".
- ogni componente IDF, costituisce un servizio indipendente, comunica con gli altri componenti via message-broker.
- I componenti IDF sono progettati per l'esecuzione all'interno di un "Docker container", ossia un contenitore virtualizzato che fornisce una astrazione del software dal sistema operativo della macchina ospitante. I diversi containers verranno orchestrati via "Docker Swarm/Compose"
- Il sistema pertanto risulta scalabile secondo le esigenze e le risorse hardware disponibili. E' quindi possibile installare uno o più "MODULI IDF" di ogni componente. Per ogni "MODULO IDF" è possibile configurare la GPU sulla quale attestarlo, la percentuale di memoria da utilizzarlo e il numero di core della CPU dedicati.
- Il sistema può essere configurato , adattando risoluzione e frequenza delle immagini, per poter gestire con una scheda video circa 45 flussi.



Uno sguardo al futuro

Evoluzioni future:

IDF è in continua evoluzione per integrare le più avanzate tecnologie e fornire all'utente prestazioni sempre migliori in termini di attendibilità e puntualità delle segnalazioni.

Nuovi algoritmi e nuove reti neurali vengono integrate e testate, la roadmap attuale prevede l'integrazione di YOLOv4 e l'introduzione di nuovi algoritmi di tracking. IDF viene sempre aggiornato alle ultime librerie disponibili tra le quali nvidia-driver, CUDA, Tensorflow, OpenCV.

Desideri ulteriori informazioni?

Invia una e-mail a info@ingesw.com o contatta il team di pre-sales in I&SI al numero +39 (06) 928651 e scoprirai come IDF S.A.R.A.[®] può aiutarti a far crescere il tuo business in totale sicurezza

I&SI S.p.A., via della Meccanica , 2b 04011 Aprilia (LT), Italy.
Tel +39 (06) 928651 Fax +39 (06) 92865255
web www.ingesw.com e-mail to: info@ingesw.com

