

eXtraWay Platform Server - eXtraWay OverView

Le ragioni di una scelta

RDBMS e Data Base Post-Relazionali, il perché di una scelta

Ogni qual volta si parli, genericamente, di **Data Base**, la nostra mente va ai **Data Base Relazionali** in quanto essi, oltre ad affondare le **radici** nelle origini del concetto stesso di Data Base, sono indubbiamente i primi ed i più diffusi **strumenti software** in grado di dare risposta all'esigenza di conservare e rendere fruibili dei **dati**, mansione per la quale hanno rappresentato a lungo l'unica soluzione.

Parliamo infatti espressamente di **dati** quando pensiamo alle più comuni realizzazioni effettuate con **Data Base Relazionali**. L'insieme di questi dati assume un particolare significato e diviene di un certa utilità solo se visto nell'ambito di una precisa **struttura d'archivio** rappresentata in modo bidimensionale tramite **tabelle** e **colonne**, le quali devono essere combinate tra loro tramite legami complessi.

Il singolo **dato** infatti, altro non è che una parte di un **concetto** più ampio, quello di **informazione**, che assume una forma ricca, strutturata, articolata e che deve invece essere normalizzata per rientrare nei canoni di **tabelle** e **colonne** di un **RDBMS**.

Perché questi **dati** assumano appunto il ruolo di **informazioni**, e quindi risultino di effettiva utilità, è necessaria la realizzazione di applicazioni che guardino alla **struttura** del **Data Base** come ad una realtà pluridimensionale ottenuta da una razionale combinazione di singoli scenari bidimensionali.

Compito di queste applicazioni è dare all'utenza una visione d'insieme di questa realtà attraverso la quale il **dato destrutturato** viene reso **significativo** dandone una **rappresentazione** sostanzialmente strutturata.

Quello che la mente concepisce come **informazione** deve essere frazionato e destrutturato per poter essere rappresentato correttamente da un **RDBMS** per poi essere nuovamente organizzato in una **forma** complessa che lo renda nuovamente significativo e fruibile per gli utenti.

Quest'attività, che in alcune applicazioni risulta certamente sensata e funzionale, può incontrare **limiti** non semplicemente **prestazionali** ma anche legati alla complessità delle **informazioni** da rappresentare nell'ambito del **Data Base**. Il compito della **normalizzazione** può risultare particolarmente complesso ed il risultato sin troppo distante dalla **forma naturale** che l'informazione avrebbe nel mondo reale.

Laddove si intenda privilegiare il **ruolo** di **informazione** che un **insieme** di **dati** può esprimere viene quindi naturale concepire i **dati** stessi in **forma strutturata**. Una simile esigenza trova una naturale soluzione nell'adozione di **Data Base Post-Relazionali** ed ancor più nell'uso del **linguaggio XML**.

Parlando di **Data Base Post-Relazionale** si vuole esprimere un **concetto** che ha assunto diversi significati nel tempo.

Se le prime esperienze Post-Relazionali hanno condotto alla realizzazione di **sistemi** che avevano origine dagli **RDBMS** e si discostavano da essi solo in piccola parte, la naturale evoluzione del panorama informatico, specie attraverso la diffusione di Internet, ha condotto una schiera di **fornitori** e **fruitori** di **informazioni** a richiedere funzionalità più ampie ed una **rappresentazione** sempre più ricca delle informazioni reperite.

Mentre i **Data Base Relazionali** hanno continuato a soddisfare le esigenze di chi doveva gestire **dati**, i **gestori** e fornitori di **informazioni**, e soprattutto di conoscenza, hanno trovato in sistemi Post-Relazionali le risposte alle loro specifiche esigenze. La particolare attenzione alle funzionalità di **Full Text Retrieval**, evoluta nel tempo nella direzione della **multimedialità** delle basi dati, del **Web Semantico** e delle **Ontologie**, fanno dei Data Base Post-Relazionali la realtà in grado di affiancarsi agli RDBMS per completare a 360° il panorama dei Data Base. Diamo a Cesare quel che è di Cesare: un Data Base Relazionale è semplicemente uno strumento diverso rispetto ad un Data Base Post-Relazionale. I due svolgono compiti differenti, se pure simili, e si prestano a soddisfare differenti esigenze. Sarebbe irragionevole pensare che **Data Base Relazionali** e **Post-Relazionali** possano sostituirsi gli uni agli altri ed adattarsi a diversi obiettivi funzionali. Essi sono concepiti innanzitutto per rappresentare i **dati** ovvero le **informazioni**, in modo differente e, sulla base di questa diversa **architettura**, offrire diverse **funzionalità**.

Come detto, la diffusione di Internet e l'esigenza di distribuire **informazioni** e **servizi** tramite la rete ha portato alla crescita di un'ampia gamma di **strumenti** di **sviluppo** orientati alle **applicazioni Web**. Tali strumenti, nati nell'ambito della **programmazione** ad **oggetti**, hanno naturalmente un orientamento alla gestione dei dati che ne ricalca l'origine. Il **dato** è visto come **oggetto strutturato** non solo perché questa visione semplifica lo **sviluppo** delle **applicazioni**, ma anche e soprattutto perché in tal modo risulta maggiormente rappresentata la **forma** e l'**articolazione** che le informazioni, o ancor meglio la conoscenza, assumono nel mondo reale. Il processo di evoluzione che ha portato alla realizzazione di questi strumenti ha parallelamente condotto alla generazione di **Data Base orientati** agli **oggetti**, detti anche **ODBMS** in contrapposizione agli **RDBMS**.

Compito dei Data Base Post-Relazionali ed in particolare degli **Object Oriented Data Base Management Systems** è quindi quello di unire il meglio di questi scenari, favorendo tanto la **semplicità** nello **sviluppo** di applicazioni per la fruizione delle informazioni quanto la **gestione** delle stesse attraverso un formato intuitivo ed in grado di rispecchiare la loro reale natura.

Altre **esigenze** si miscelano a quelle già esposte.

In primo luogo sarebbe auspicabile che le **informazioni** fossero fruibili nel tempo limitando al minimo gli sforzi, e quindi i **costi** perché esse risultino intelligibili.

Una delle vie che intuitivamente va seguita per garantire il conseguimento di simili obiettivi consiste nella scelta di **Standard Aperti** per la **rappresentazione** dei **dati**. Se ad essa uniamo quanto precedentemente detto, ovvero scegliamo di porre l'accento sul concetto di **informazione** e sulla sua descrizione come singola **entità** che rappresenta un'unità di informazione consistente e



completa di ogni sua parte, la scelta del **linguaggio XML** per rappresentare queste unità si presenta come la più naturale. Questa unità di informazione, che può trovare facilmente rappresentazione anche sotto forma di **Oggetto** di un **ODBMS**, verrà di seguito identificata con il termine **record**. Un record rappresenta un'unione indivisibile di **dati strutturati**, arricchiti da **metadati descrittivi** ed eventuali **allegati informatici**.

La scelta di rappresentare tali **records** tramite il **linguaggio XML** si rivela particolarmente adatta anche a garantire la **sicurezza** delle **informazioni**.

Il progetto eXtraWay, un Native XML DBMS

Le scelte indicate sino ad ora sono quelle alla base del progetto eXtraWay di 3D Informatica. Ultimo di una lunga serie di esperienze nel mondo dell'**Information Technology**, il **progetto eXtraWay** sposa una ventennale attività nel settore dell'**Information Retrieval** con la **filosofia XML** conducendo alla realizzazione di un Native XML Data Base.

eXtraWay è un insieme di **moduli** che consente di gestire e conservare nel tempo in modo sicuro **record di informazione**, cioè **unità di informazione** consistenti, complete, in modo da garantirne l'**integrità**, **autenticità**, **confidenzialità** e **intelligibilità** nel tempo.

Per meglio comprendere cosa si intenda per **record** o **unità di informazione** e l'importanza ricoperta in particolare dalla sua **integrità** ed **autenticità**, approfondiamo il concetto di **documento** nell'accezione che tale termine assume nell'ambito del **progetto eXtraWay**.

Un **documento** è il risultato di un **atto di volontà**, la volontà di un singolo o di un'organizzazione di dare **testimonianza** di un evento o di un'intenzione, di comunicare a terzi una notizia o richiedere ad essi delle informazioni. Esso assume il suo reale **significato** quando è auto-consistente e completo in ogni sua parte e quando può rispondere ai più classici dei **quesiti**: chi, cosa, dove, come, quando e soprattutto perché. La sua **completezza** deve quindi essere garantita nel tempo così come la sua **disponibilità** ed **intelligibilità**.

Il compito di conservare questi **oggetti auto-consistenti** è pienamente svolto da **sistemi post-relazionali** come eXtraWay che ha fatto di un simile approccio alle **informazioni** una delle sue principali caratteristiche.

In un **sistema relazionale** il **documento**, così come l'abbiamo inteso, sarebbe frazionato in molteplici **tabelle**. Solo tramite le **relazioni** tra di esse sarebbe possibile ricostruire il documento nella sua forma originaria, o quanto meno in una forma ad essa prossima. Sempre in un simile scenario, **interventi di modifica** sui contenuti delle singole tabelle, avulsi dal concetto di documento, porterebbero lo stesso ad avere una forma ed un contenuto differenti da quelli originari, senza la possibilità di ricostruire la sua reale "sostanza". Ciò mostra chiaramente come l'**approccio relazionale** al documento non possa soddisfare pienamente il requisito dell'**integrità**.

Altra caratteristica fondamentale di eXtraWay consiste nell'aver scelto la **rappresentazione dei record** in forma **XML**. Ciò non vuol dire semplicemente che il **documento** viene "mostrato" in un **output XML** mentre si trova riposto in un qualche contenitore eventualmente in un diverso **formato**, comportamento che appartiene a molti altri **Data Base**, relazionali e non, bensì che esso si trovi fisicamente in **formato XML** nativo sul file system. Questa scelta offre la maggior garanzia di **intelligibilità** e **disponibilità** dei **documenti** nel tempo.

Il motore realizza la sua missione esponendo **comandi di salvataggio** ed **accesso ai record**, ovvero ai **documenti**, di tipo atomico, cioè non frazionabile, automaticamente corredati da metadati sia in codice che in chiaro, relativi all'utente, la data e l'orario, l'organizzazione corrente del produttore, del piano di classificazione, del sistema software e sigillati con un codice di integrità. Il **record** è quindi l'**unione** indivisibile e dimostrabile di dati, metadati ed eventuali allegati informatici. Ogni **cambio di stato** subito da un **record** viene salvato da un **sistema di tracking** persistente, che permette la **ricostruzione** dell'immagine corrispondente del record a seguito di ogni **azione**, dalla creazione fino alla versione corrente.

La sicurezza nei formati standard aperti

Gli **obiettivi** della **consistenza**, **completezza**, **integrità** ed **autenticità** dei **record** sono da considerarsi un tutt'uno con quello della loro **sicurezza**.

La **sicurezza informatica** risulta dalla somma di più **fattori**, in primo luogo:

Confidenzialità	Solo soggetti autorizzati possono accedere ai record
Autenticità	Il record è quello che dichiara di essere
Integrità	Il record è consistente e completo
Disponibilità	Un record è accessibile e intelligibile a chi ne ha i diritti

Confidenzialità

Se guardiamo con occhio critico alla **sicurezza** offerta da un qualsiasi **Data Base** avremo di fronte a noi due possibili scenari. Uno di essi è rappresentato da un insieme di **politiche di accesso** e di **protezione** delle **informazioni** gestite applicativamente, l'altro consiste nella **cifatura** dei **dati** perché siano disponibili esclusivamente al loro **proprietario**.

La **sicurezza** di queste **informazioni** mostra almeno due **punti deboli**. Nel primo scenario siamo comunque nelle mani dell'**amministratore** del **Data Base** il quale ha accesso a tutti i **dati** e, come lui, tutte le persone che sono preposte ad amministrare queste informazioni. Per ragioni pratiche le **password** di accesso amministrative sono spesso a conoscenza di diverse persone ed il dato, in quanto tale, è sicuro nella misura in cui è "fidato" l'amministratore. Nel secondo scenario, per contro, l'**informazione** che è effettivamente altamente protetta corre il rischio di essere "persa" laddove la sua **accessibilità** è legata a fattori estremamente dipendenti dal singolo. Sarà sufficiente una **password** persa o dimenticata ed il **documento** non sarà più fruibile.



La tendenza ad ottenere la **sicurezza** nascondendo le **informazioni**, **Security through Obscurity**, lascia quindi il posto ad un diverso approccio. Compito dell'amministratore è garantire un **accesso esclusivo** alle informazioni da parte del **motore eXtraWay** e non essendo necessario porre le informazioni in **CLOB** o **BLOB** di un Data Base Relazionale per metterle al sicuro da occhi indiscreti, l'uso di **formati aperti** è pienamente accettabile portando in dote altri vantaggi che risulteranno più evidenti in seguito. Questo approccio ha comunque la necessità di proteggere l'**accesso** ad alcuni **dati** anche dallo sguardo degli amministratori. In questo caso la **cifatura** tramite **algoritmi** standard basati su chiavi che verranno consegnate a persona diversa dall'amministratore di sistema (ad esempio il responsabile dell'archivio, che tipicamente non accede in sala server o comunque non ha le credenziali di amministratore di sistema) fa sì che solo l'**intervento** congiunto delle due figure permetta di decifrare il contenuto di tali **record** al di fuori delle **politiche di sicurezza**.

Autenticità

Si può presumere che un **record** sia autentico quando viene garantita anche a distanza di tempo tutta la catena di **operazioni** che ha subito da parte dei relativi **utenti**, documentandone la politica di accesso, i **riferimenti temporali** e quando è possibile individuando e circoscrivendo gli eventuali momenti di discontinuità nella **sicurezza** del **sistema**. A tale scopo eXtraWay registra in chiaro, in due **allocazioni** specifiche, sia i riferimenti alle **operazioni** di routine, sia gli **eventi** straordinari che possono abbassare temporaneamente il livello di sicurezza. Ciò in quanto l'**autenticità** dell'**archivio**, e quindi dei **record**, risulta tanto maggiore quanto tali **eventi** sono rilevabili indipendentemente dal **programma**, e quindi in chiaro ed in **posizioni note**, e siano conosciuti e circoscritti.

La capillare **registrazione** dei **cambi di stato** di ogni **record**, unitamente alla possibilità di identificare tutti i **momenti di discontinuità**, consentono appunto di dare ragionevoli **garanzie di autenticità** ad ognuno di essi, condizione che non può essere altrimenti garantita da **sistemi** privi di una **tracciatura** dettagliata.

Integrità

Un **formato aperto**, dotato di un **sigillo noto**, dà evidenza dell'**integrità** del **record**. eXtraWay completa ogni **record** con:

- L'impronta degli eventuali **allegati informatici** secondo un **formato noto (SHA1 o RIPMED160)**;
- L'impronta dell'intero **record** (eccetto i suddetti allegati) secondo un **formato noto**.

eXtraWay verifica l'impronta dell'intero **record** ad ogni suo **caricamento**, mentre il **controllo** dell'**impronta** degli **allegati** viene fatto al momento dell'**accesso** ad essi. Un **processo di background** può comunque effettuare entrambe i **controlli** nei **tempi** in cui il **sistema** non è utilizzato.

L'aspetto da evidenziare è che l'amministratore può verificare l'**integrità** del **record** e dei **file** ad esso associati anche senza utilizzare eXtraWay, e quindi anche a distanza di tempo e su un altro sistema con **comandi da prompt**.

Anche in questo caso eXtraWay non nasconde, bensì dichiara.

Disponibilità

La **disponibilità** va riferita al servizio, ai record e alla loro intelligibilità per chi ne ha i diritti.

La **disponibilità** del **servizio** non è tanto determinata dal **formato** delle **informazioni** ma dalla **robustezza** del **motore** e dalle misure di **prevenzione** degli **eventi dannosi** (intrusioni, manipolazioni). Il formato in chiaro permette però agli amministratori di evidenziare con più modalità e strumenti la **validità** almeno formale di **file** e **configurazioni**.

La **disponibilità** delle **informazioni** e la loro **intelligibilità** beneficia in modo evidente dall'**organizzazione** e dal **formato aperto**. Quando i **dati** ed i **file associati** sono in chiaro e fisicamente adiacenti il loro orizzonte temporale si estende oltre quello dei **programmi** che li producono e li gestiscono.

I **record** sono spesso un **patrimonio primario** di un'organizzazione e il loro mantenimento rischia di non essere economicamente sostenibile se richiede una **conversione** del pregresso ad ogni cambio di **tecnologia software**.

Supporto nativo del linguaggio XML

eXtraWay memorizza tutti i **dati** e **metadati** di ogni record in un'unica **porzione** indivisibile di un opportuno **file XML**.

L'**allocazione** del file e la sua **posizione** in sub-directories è lasciata ad una politica di gestione scelta dall'**amministratore** dell'archivio, non dal programmatore.

Ogni **porzione XML**, a parte i casi in cui viene cifrata perché contenente dati sensibili, è **intelligibile** a chi possa accedere fisicamente al file, tipicamente l'**amministratore** del server o i **conservatori** delle copie di **backup**. Il **motore di eXtraWay** riconosce la **struttura XML** del record creando uno specifico **indice** per ogni diverso percorso dalla **radice** del **record** ai suoi elementi ed attributi ed ogni altro indice esteso frutto dell'**elaborazione** dei contenuti del **record**.

Durante il **salvataggio** del **record**, eXtraWay verifica che sia ben **formato** e la sua **validità** rispetto ad un modello dichiarato (**DTD** o **schema**).

Gli **approcci** ai quali si deve far fronte sono normalmente due:

- **Strutture dati XML** già **esistenti**, provenienti dal porting di basi dati espresse in altro formato e quindi frutto di una conversione non sempre razionale o pensata per ottimizzare le prestazioni dell'applicazione che si andrà a realizzare. Gli stessi **file** da associare ai record identificati possono essere dislocati nel **file system** anche in modo scarsamente razionale. In un simile scenario è eXtraWay ad adattarsi alla natura dei **file XML**, a modellare la creazione dei canali di ricerca sulla base di quanto disponibile con o senza la presenza di **DTD** atte a dichiarare il corretto formato delle informazioni. L'**archivio** risultante può avere limiti di portabilità ai quali si può porre rimedio con una riorganizzazione dei file d'archivio per mezzo



delle **politiche** di **distribuzione** sul file system evidenziate nei paragrafi successivi;

- **Strutture realizzate** espressamente per rappresentare le **informazioni** nel modo più coerente e razionale. Studiate e create con l'ausilio dei **Tools** di eXtraWay conducono alla **realizzazione** di **archivi** assolutamente portabili e ad applicazioni in grado di sfruttare pienamente le potenzialità degli **indici** che eXtraWay produce.

Architettura Software

Innanzitutto, per comprendere l'**architettura** della **Piattaforma Documentale eXtraWay** è necessario comprendere alcuni fondamenti su cui l'intera architettura poggia.

In primo luogo vale la pena di sottolineare che un'**applicazione** realizzata con la **tecnologia eXtraWay** prevede normalmente una **struttura 3 tiers** ovvero la presenza di tre **strati software** che si occupano di:

- **Presentazione** delle **informazioni** raccolte e controllo delle informazioni inserite dall'utente (Web Server);
- **Elaborazione** delle **informazioni** inserite dall'utente e gestione delle comunicazioni con la banca dati (Application Server);
- **Gestione** dei **dati** vera e propria (DataBase Server).

Come collocare quest'**architettura software** nella più opportuna **architettura hardware** sarà più chiaro nell'apposito capitolo ma già in questo scenario è piuttosto evidente il **ruolo** svolto da ciascun **livello** del **software**. Per semplicità assumiamo che i tre **strati software** insistano sulla stessa macchina senza addentrarci nei casi particolari.

Come abbiamo visto avremo i seguenti **componenti**:

Web Server	Si tratta di un comune Web Server ¹⁾ . Assolve autonomamente al compito di regolare gli accessi e di rendere fruibile l'applicazione all'utenza
Application Server	Strato "intermedio" ove insiste l'effettiva Business Logic dell'applicazione documentale. Questo strato può essere realizzato con diverse tecnologie le quali fanno tutte riferimento ad una componente Java nota come Broker . Esso è lo strumento di più basso livello col quale dialogare con lo strato sottostante, ovvero con il Data Base Server e si avvale, solitamente, di un Java Container come Apache Tomcat ²⁾ . Sul Broker sono poi stati creati diversi altri strumenti di dialogo con il server di più alto livello, quali ad esempio un ampio set di Web Services. Questo strato di software, basato su Java, è nativamente Platform Independent .
Data Base Server	Il vero cuore pulsante della Piattaforma eXtraWay , il vero protagonista di questo capitolo.

Il **cuore** di eXtraWay è il **motore**, come già detto un **XML Native Data Base & Information Retrieval Engine**.

E' costituito da un **server multiprocesso, multithread**, scritto in linguaggio **C/C++** disponibile sui **sistemi operativi** più popolari:

- **Windows**;
- **SunOs** su processori Sparc;
- **Linux** ³⁾ su processori Intel x86;
- **Linux** su processori PowerPC ⁴⁾;
- **Aix** su processori Power PC;
- **MacOs 'X'** su processore Motorola;
- **MacOs 'X'** su processore Intel.

Se non lo si può considerare **Platform Independent** in quanto modulo eseguibile espressamente compilato per un sistema operativo, la varietà di **versioni** disponibili copre un area di utenza estremamente vasta. L'esperienza accumulata negli anni, inoltre, rende il **porting** verso altre **piattaforme** un'azione meno critica di quanto non si possa supporre.

Le Struttura SoftWare

Le componenti di terze parti

Lo schema precedente ha posto in evidenza l'esigenza, nei tre **strati software** indicati, di avvalersi di **strumenti** di terze parti quali **Web Servers** e **Java Containers**.

Il presente manuale non ha la finalità di descrivere come essi vadano installati o gestiti limitandosi, dove necessario, a dare **indicazioni** su come essi debbano essere configurati.

eXtraWay Server

Installazione e Struttura del Server

Nel momento in cui viene stilato questo documento non esiste un vero e proprio **Setup** per le **piattaforme Linux/Unix** ⁵⁾ diversamente da quanto avviene per le **piattaforme Windows**. Ad ogni modo, fatta eccezione per i **moduli** dei due strati precedenti, installare il **Server eXtraWay** corrisponde a realizzare una struttura di directory e compiere la copia in esse di alcuni files.

La **struttura** del **software** è, di fatto, molto semplice.

Di seguito vedremo qual è lo **standard** adottato anche se le diverse **installazioni** possono essere soggette a **variazioni formali** se pur non sostanziali.

Il primo è più importante “distinguo” va fatto tra le diverse **versioni** di **piattaforma Windows** e tutte le varie **piattaforme Linux/Unix** in quanto il **percorso di installazione iniziale** differisce.

	Percorso di installazione
Windows	c:\3di.it\extraway\xw
Linux/Unix	/opt/it-3di/extraway/xw

Che rispetti o meno questi standard, ciascuna **installazione** ha un “punto di partenza” dal quale ha **origine** tutto quanto ne consegue. Identificata questa **directory** diviene semplice identificare tutte le successive parti e poter interpretare il comportamento dell'applicazione.

Identificato il punto di **origine** come xw_root in esso avremo le seguenti **directory**:

bin ⁶⁾	Directory contenente tutti gli eseguibili Platform Dependent.
conf	Directory ove trovano collocazione tutti i files di configurazione dei moduli presenti nella directory bin .
context	Directory ove trovano collocazione i files di contesto.
db	Directory ove trovano collocazione naturale gli archivi. In tale directory viene realizzata una ulteriore directory per ciascun archivio ⁷⁾ entro la quale vengono collocati tutti i files che lo compongono. Il server è in grado di utilizzare archivi collocati in qualsiasi posizione ma questa è quella che potremmo definire preferenziale.
jobs	<i>Il contenuto di questa directory è funzionale allo svolgimento di attività off-line del server. Non è necessario, in questa sede, scendere nel dettaglio.</i>
logs	Directory ove trovano collocazione tutti i files di log prodotti dal server.
xreg	<i>Il contenuto di questa directory è funzionale allo svolgimento di attività di tenuta del registro delle attività svolte dal server. Non è necessario, in questa sede, scendere nel dettaglio.</i>
lazy	<i>Il contenuto di questa directory è funzionale allo svolgimento di attività near-on-line del server. Non è necessario, in questa sede, scendere nel dettaglio.</i>

Esistono altre **directory** la cui presenza non è obbligatoria ma può dipendere dalla **versione** del **server** o dal fatto che le **applicazioni** su di esso realizzate sfruttino o meno alcune funzionalità.

collect	Directory contenente tutte le raccolte, ovvero insiemi di documenti equiparabili a selezioni non volatili di documenti eventualmente provenienti da archivi diversi tra loro.
conversion	<i>Il contenuto di questa directory è funzionale allo svolgimento di attività di conversione di files, estrazione del testo per l'alimentazione dei vocabolari, realizzazione di miniature di immagini o files .PDF a partire da un altro tipo documentale. Non è necessario, in questa sede, scendere nel dettaglio.</i>
tmp	Alternativamente alla directory di sistema destinata ai files temporanei, essi possono essere indirizzati in questa collocazione.
wd	Directory per lo svolgimento di attività di importazione documenti. Essa può essere collocata in altra posizione ma, come per il caso della directory db questa è la sua posizione naturale.

Come opera il server eXtraWay

Indipendentemente dal **sistema operativo** sul quale si installeranno i **programmi** della **Piattaforma eXtraWay**, c'è un'importante aspetto da conoscere per meglio comprendere come **eXtraWay Server** operi.

L'esecuzione del programma, manualmente o sotto forma di servizio, avvia un'**istanza** del **server** detta **Master**. Essa offre un servizio su una porta **Socket** nota ⁸⁾ e dialoga con **Client** che condividano con esso un **protocollo personalizzato**. In seguito l'argomento verrà approfondito parlando del **Broker**.

L'**istanza Master** attende che venga richiesta una **connessione** al server da parte di un **client**. Quanto questo avviene l'**istanza Master** produce un **clone** di se stesso ⁹⁾ che è un **processo figlio** detto anche **Slave**. E' questo il processo che, effettivamente, dialoga col **client** sino a quando il client non avrà più bisogno dei suoi servizi.

Le attività di **generazione** di **processi Slave** non sono incontrollate: il **server Master** è sottoposto ad una **registrazione** che stabilisce una tantum quante **istanze**, ovvero quante **connessioni**, siano consentite. Un **server Master** non può generare più server **Slave** delle connessioni per le quali è stato registrato, il tentativo di connettersi ad un **server** che abbia esaurito le proprie connessioni scaturirà in un **errore** e la connessione verrà negata.

La **registrazione** è principalmente finalizzata ad identificare alcuni **metadati**, ad esempio l'organizzazione che gestirà i **dati** dei **DB**, che caratterizzano i dati dei DB stessi in forma di **Processing Instructions**. Essa ha inoltre lo scopo di proporzionare il massimo numero di **istanze** di **server eXtraWay** alle capacità dell'**hardware** in cui esso opera per evitare un proliferare incontrollato di istanze con un conseguente degrado prestazionale.

In pratica, tramite la **registrazione**, si mette sotto controllo l'**uso** che si farà del **server**.

Il Servizio eXtraWay

Quale che sia stata la procedura di **installazione** di **eXtraWay** così come delle componenti degli altri due **strati software** coinvolti dalla piattaforma, è normalmente necessario che essi operino come **servizi**, ovvero che siano eseguiti automaticamente dalla **macchina** e non influenzati dalle **attività** di un singolo operatore.

Se è stata effettuata la procedura di **installazione** in **Windows** il **servizio** noto come “**eXtraWay Server**” sarà stato regolarmente installato ed avviato. Discorso analogo può essere fatto per l'installazione **Linux/Unix** che spiega come impostare l'avvio e lo spegnimento del servizio ¹⁰⁾.

Vediamo comunque come avviare o interrompere il **servizio “eXtraWay Server”** ovvero come avviare manualmente i **moduli**



sulle diverse **piattaforme**.

Piattaforma Windows

Per l'avvio e l'interruzione del “**servizio eXtraWay**” se esso è stato regolarmente installato come tale, non sono necessarie particolari **attività** se non l'utilizzo dei **Servizi** di **Windows** nell'ambito degli **Strumenti** di **Amministrazione Comune**.
Differente discorso va fatto se il **servizio** non è stato ancora installato e si vuole procedere alla sua **installazione** ed esecuzione o se, all'opposto, si desidera eseguire manualmente **eXtraWay Server**.

Installazione del Servizio eXtraWay

Diversamente dalle precedenti **distribuzioni** dei **server eXtraWay**, l'installazione come servizio viene ora effettuata direttamente con **parametri** sulla riga di comando del **server** stesso. In sostanza sarà sufficiente eseguire da un qualsiasi **prompt** dei **comandi** una delle seguenti operazioni.

Operazione	Comportamento
xw.exe -service_install	Installa (ed avvia) il server eXtraWay come servizio.
xw.exe -service_start	Avvia il servizio correlato al server eXtraWay
xw.exe -service_stop	Ferma il servizio correlato al server eXtraWay
xw.exe -service_uninstall	Disinstalla (fermando se attivo) il servizio del server eXtraWay.

Tutti i **comandi** suddetti reagiscono anche al **parametro** -p<numero_porta> consentendo in questo modo di installare più **istanze** distinte di server operanti su porte **socket** diverse.
Una volta installato come servizio, anche se il **server** consente l'avvio e l'interruzione da riga di comando, si rimanda ai **comandi** del **pannello Servizi** di **Windows**.
Per quanto siano richiesti **diritti amministrativi** per poter installare eXtraWay, non è necessario che al servizio vengano dati diritti particolari ne tanto meno ai file presenti su disco cui esso dovrà fare accesso.

Avvio Manuale

Per quanto concerne l'**avvio manuale** del **modulo eXtraWay Server** sarà sufficiente, da Esplora Risorse, accedere alla **directory** bin degli eseguibili e fare un doppio click sul **modulo** denominato xw.exe . A quest'azione seguirà l'**avvio** del **server** dopo una breve **pausa**, di circa 20 secondi, necessaria al modulo a verificare di non essere stato eseguito come servizio.
Alternativamente esso può essere eseguito da un qualsiasi **Command Prompt** aggiungendo alcuni **parametri** su **riga di comando**. Quelli più comuni sono

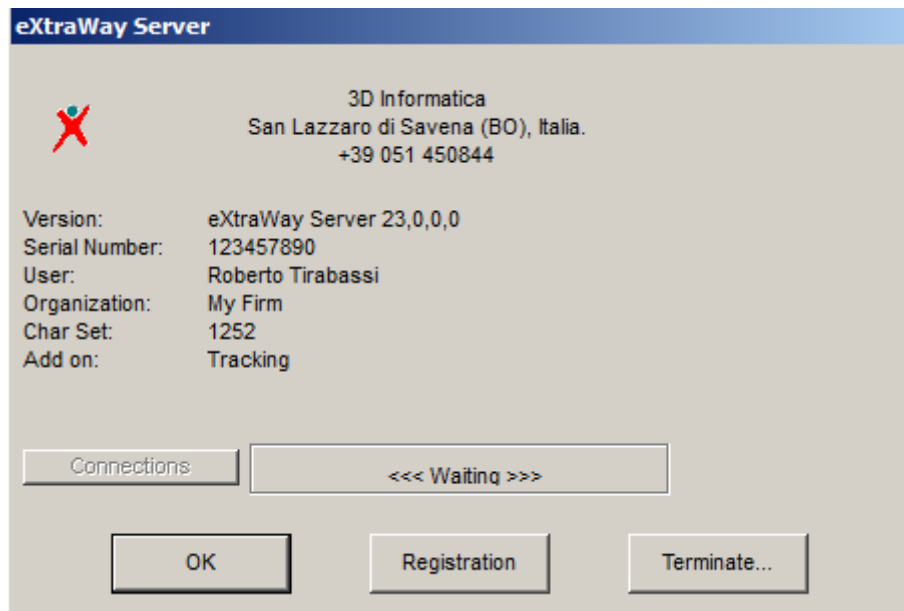
-noserv	Indica al server eXtraWay che non deve cercare il dialogo con il Service Control Manager di Windows. Questo comporta l'avvio immediato del server senza l'attesa di 20 secondi descritta precedentemente.
-p<numero>	Indica al server eXtraWay di avviarsi e porsi in ascolto su una specifica porta socket differente da quella di default ¹¹⁾ .

Gestione del Modulo

Quale che sia stato il metodo per mezzo del quale è stato posto in esecuzione il **Server eXtraWay** esso darà traccia evidente della propria presenza nella **Barra dei Processi** ¹²⁾. La sua **presenza** è resa manifesta dalla sua **icona** nella **Tray Bar**.



Ponendo il **mouse** sul simbolo presente nella **Tray Bar** verrà mostrato l'**ID univoco** del **processo** in **esecuzione** ¹³⁾.
Facendo **click** su tale simbolo verrà mostrato un dialogo di **informazioni** come il seguente:



Ad esso corrispondono **informazioni** che verranno spiegate meglio in un apposito **paragrafo** come il significato del **numero di serie** **numero di serie** così come **la registrazione** del server. Il **pulsante** Termina consente l'**interruzione** del **server principale** e, con esso, di tutti i processi figli. Se il server è stato eseguito come servizio di **Windows** questo tasto è disabilitato perché gli interventi di avvio ed interruzione del servizio devono essere compiuti dal **Service Control Manager**.

Utilizzando invece il **bottone** Connections verrà visualizzato l'**elenco** dei **processi figli**, indicando per ciascuno di essi l'utente che lo sta utilizzando, l'**IP Address** dal quale sta operando e da quanto tempo.



Tramite questo dialogo è possibile interrompere le **attività** di quell'**utente** provocando la chiusura del **processo figlio**.

Piattaforma Linux/Unix

Diversamente dalla piattaforma Windows l'installazione su **piattaforma Unix/Linux** richiede qualche **accorgimento di sicurezza** in più.

In primo luogo è fortemente sconsigliato lasciare che **eXtraWay Server** venga eseguito con **diritti** dell'**utente** root. E' prassi comune, nonché buona norma, creare un **utente**, solitamente extraway, cui verrà dato il compito di far eseguire eXtraWay Server. Tutti gli **archivi** da esso gestiti dovranno appartenere allo stesso utente.

Installazione del Servizio eXtraWay

L'installazione di eXtraWay Server come servizio richiede di utilizzare lo **script** `xwctl`^[14] direttamente in `/etc/init.d` ovvero `/etc/rc.d` a seconda del sistema operativo. Sarà sufficiente creare un **symlink** allo **script** presente nella **directory** dei binari ed utilizzato anche per l'avvio manuale del server.

Una volta compiuti questi passi si dovrà prestare attenzione al **file** di **configurazione** dello **script** `xwctl`, denominato `xwctl.conf` e presente nella **directory** `conf` dei files di configurazione.

In tale file vanno dichiarate le **impostazioni** di partenza del **server eXtraWay**. Il file è composto da una serie di **righe** sotto forma di **etichetta=valore** precedute dal **simbolo** `#` che le commenta, ovvero le rende inefficaci. Ciascuna riga riporta il **valore** di **default** o un'indicazione sul **contenuto** da impostare. Di seguito i **valori** che può essere necessario compilare:

xw_user	Utente con il quale viene eseguito extraway Server, che sia un servizio o eseguito manualmente ^[15]
----------------	--



xw_home	Collocazione della directory principale ¹⁶⁾ di eXtraWay ¹⁷⁾
xw_port	Porta sulla quale opera eXtraWay Server ¹⁸⁾
xwctl_log	Collocazione del file di log ove xwctl registra le attività svolte ¹⁹⁾
xw_ldpath	Può essere necessario che il server usi librerie di terze parti non presenti nella directory di default. In tal caso si deve indicare al server ove cercare tali librerie ²⁰⁾
xw_nicelevel	eXtraWay Server può essere eseguito con priorità più o meno elevata. Se si desidera che eXtraWay Server non limiti sensibilmente l'attività di altri servizi installati sullo stesso server si può modificare il suo livello di priorità ²¹⁾

Avvio Manuale

L'avvio manuale del modulo avviene per mezzo dello stesso **script** xwctl indirettamente utilizzato per l'**avvio** sotto forma di **servizio**. Eseguendo lo **script** da **command prompt** si può regolare l'attività della piattaforma eXtraWay.

./xwctl start	Richiede l'avvio del server eXtraWay ²²⁾ sulla porta stabilita. Se il servizio è già attivo lo script darà errore
./xwctl stop	Richiede l'interruzione del server eXtraWay. Unitamente al server interrompe anche il modulo xwls ²³⁾
./xwctl restart	Richiede, nell'ordine, l'interruzione e l'avvio del server eXtraWay
./xwctl status	Mostra lo stato del server in forma sintetica, adeguata all'uso entro ulteriori script che richiamino questo. Si limita quindi ad indicare se il server sia in esecuzione o meno
./xwctl vstatus	Mostra lo stato del server, se avviato o meno ²⁴⁾ . Utilizzabile sia per le verifiche manuali che per verificare il corretto comportamento del server eseguito come servizio. Mostra l'elenco di tutti i PID dei processi in esecuzione

Gestione del Modulo

Oltre a quanto indicato nel **precedente paragrafo** sono disponibili i seguenti **comandi**.

./xwctl version	Mostra il numero di versione di tutti i moduli binari presenti nella directory, che si tratti di moduli eseguibili o librerie dinamiche ²⁵⁾ . Particolarmente utile per verificare che tutti i moduli coinvolti siano di versioni compatibili tra loro
------------------------	---

Per ottenere altre **indicazioni** sul **server eXtraWay** sarà invece necessario richiamarlo direttamente eseguendo il **comando**:

```
./xw -v
```

che propone una serie di **informazioni** sul **server**, informazioni che saranno chiarite meglio nel prossimo paragrafo.

Informazioni sul Server

Il **server** mette a disposizione degli **utenti** una serie di **informazioni** inerenti:

Numero di Versione	Il numero di versione del server. Consente di identificarne le funzionalità e la compatibilità con altri moduli binari o librerie dinamiche
Numero di Serie	Quando intercorrono rapporti diretto con 3D Informatica questo numero viene assegnato al fine di rendere più semplice l'identificazione delle installazioni compiute, specialmente se lo stesso utente ne ha diverse. Affiancato a tale valore, separato da un trattino, viene normalmente presentato il numero delle istanze per le quali si è compiuta la registrazione . La registrazione può essere assente o prevedere particolari limitazioni che verranno chiarite meglio in seguito
Utente ed Organizzazione/Società	Contiene il nome dell'utente e della società che ha compiuto la registrazione, in caso contrario viene presentata la dicitura Versione Dimostrativa. L'utente si indica, solitamente, per identificare la figura responsabile dell'installazione
Tabella Codici	Solo Windows: Indica la tabella codici adottata dal sistema in uso. Sarebbe "teoricamente" possibile impostarne una differente ma non è pratica comune
Componenti aggiuntive	Solo Windows: Identificazione delle componenti aggiuntive , ovvero delle librerie dinamiche di cui il server può avvalersi per svolgere un più ampio range di funzionalità. Si tratta di funzionalità aggiuntive e non obbligatorie/necessarie al normale svolgimento dei compiti di eXtraWay Server

Registrazione del Server

La **registrazione** del **server** consiste nello stabilire il **numero** massimo di **connessioni** che si intende far supportare da questo **server** proporzionandolo alla portata dell'**hardware** disponibile e del **servizio** che si intende fornire. Tale operazione prevede quindi anche la **registrazione** di **dati** a corredo riferiti all'organizzazione che compie la **registrazione**, tali dati completeranno i **metadati** assegnati alle singole **unità informative** entro i **DB eXtraWay** prodotti e gestiti.

La **registrazione** dev'essere fatta una prima volta per avere un corretto comportamento del **server** e può essere effettuata nuovamente in particolar modo per modificare il **numero** di **connessioni accettate**.

Essa si compie come segue:

Windows	Premendo il tasto "Registration" dal dialogo mostrato precedentemente
Linux/Unix	Eseguendo il server col comando <code>./xw -r</code>

In ambo i **casi** verrà richiesto di impostare il **numero** di **connessioni** da abilitare²⁶⁾.

A seguito di questa **impostazione** verranno richieste, nell'**ordine**:

- **Numero di serie;**
- **Nome utente;**

• Nome Organizzazione o Società.

Al completamento di questa procedura l'**istanza** Master del **server eXtraWay** per mezzo del quale è stata effettuata la **registrazione** si chiuderà ²⁷⁾ e le **impostazioni** saranno attiva al prossimo riavvio ²⁸⁾.

E' importante osservare che lo **spostamento** dell'**installazione** o i **cambiamenti** delle **condizioni** della **macchina** possono impattare sulla **registrazione** ²⁹⁾.

Componenti aggiuntive

Come precedentemente accennato, le **componenti aggiuntive** non sono altro che **librerie dinamiche** delle quali si avvale **eXtraWay Server** per svolgere attività supplementari a quelle normalmente compiute. Esse devono essere presenti nella **directory** dei **binari** e devono avere un **nome** noto con l'**estensione** prevista dal **sistema operativo** ³⁰⁾.

Di seguito una sintetica **elencazione** delle **librerie dinamiche disponibili** e del loro compito, che verrà dettagliato in seguito.

libSQL2	Consente al server eXtraWay di eseguire alcune semplici attività di ricerca ed update con una sintassi SQL appositamente mappata sulle funzionalità del server
liburl	Componente dell'estensione libxcrwl
libxcrwl	Estensione Crawler. Consente di utilizzare eXtraWay Server, configurando un apposito archivio, come un crawler che interroga fonti dati pubbliche e pagina internet per alimentare un archivio con tali contenuti, una sorta di base di conoscenza
libxwsst	Componente dell'estensione libxcrwl
libxwwd	Componente necessaria per <i>Importare</i> nuovi documenti in archivi eXtraWay seguendo le regole del server e compiendo tutti i controlli del caso
libxw2occi	Componente necessaria all'interfacciamento con <i>Oracle RDBMS Server</i> per usufruire di esso come repository alternativo al <i>file system</i>

Oltre a queste **componenti** aggiuntive il **server** prevede che si possano dichiarare **librerie dinamiche** atte a svolgere **funzioni specifiche** correlate ad uno specifico **archivio**. Di esse si parlerà più avanti.

Dettaglio della Struttura

Directory 'bin'

La **directory bin** contiene gli **eseguibili** della piattaforma **eXtraWay**. Essi variano a seconda del **sistema operativo** ma hanno tutti la stessa funzione e comportamento. I **moduli**, infatti, garantiscono la piena **portabilità** degli **archivi** anche tra sistemi operativi solitamente ben poco compatibili. In sostanza, quindi, questa **directory** è la sola che richiede la **sostituzione** integrale del contenuto qualora si intendesse migrare un'**installazione** da un **server** ad uno differente.

I **principali moduli** contenuti sono di seguito descritti:

Modulo ³¹⁾	Funzione
xw	eXtraWay Server . E' il cuore principale del sistema, il vero Server che svolge tutte le funzionalità principali e si avvale di librerie dinamiche solo per svolgere funzioni supplementari.
libxwwd	eXtraWay WatchDoc Library . E' lo strumento utilizzato dal Server per acquisire in modalità batch nuovi documenti o aggiornare documenti esistenti. Svolge un controllo sul contenuto di una o più directory e se rileva in esse dei file con estensione <i>.xml</i> ne studia il contenuto, riconosce le <i>Unità Informative</i> in esso contenute e le aggiunge all'archivio. Se è stato opportunamente configurata, la libreria compie l'aggiornamento dei documenti dei quali venisse violata l'univocità, in caso contrario i documenti verranno aggiunti in coda all'archivio.
xwls	eXtraWay Log Server . E' il modulo che svolge la mansione di scrivere tutti i files di log nell'apposita directory. In sua assenza nessun log, ad eccezione del file di registro che viene collocato sempre in quella directory, verrà redatto.
hwadmin	E' il modulo per mezzo del quale è possibile controllare le attività base del Server quali accensione, spegnimento e così via. Utilizzato principalmente sulle piattaforme Linux/Unix per il controllo dei servizi, trova applicazione anche in Windows.
xw3rdp-setup ³²⁾	Insieme di librerie installabili per Windows. Comprende componenti quali libxml2 , libxslt , libiconv , openssl , libzip e zlib
msg.dat xw.rc	Files di risorse (stringhe e simili) necessarie al Server eXtraWay.

Ad essi si aggiungono questi ulteriori **moduli supplementari**:

Modulo	Funzione
xreg	eXtraWay Register Reader . E' il modulo per mezzo del quale è possibile consultare il contenuto dei files di registro che eXtraWay Server stila nella directory <i>logs</i> . La consultazione del registro viene effettuata principalmente per ricostruire le attività svolte, recuperare versioni precedenti di documenti ed altre attività simili.
libSQL2	eXtraWay SQL extension . E' la libreria dinamica per mezzo della quale eXtraWay Server consente di svolgere alcune attività ³³⁾ avvalendosi del linguaggio SQL.
libxcrwl liburl	eXtraWay Crawler Library . E' la libreria dinamica che il Server utilizza per svolgere attività di Crawling. Essa, in combinazione con una specifica applicazione ed un archivio opportunamente configurato consente di identificare delle fonti di dati ³⁴⁾ dai quali realizzare una base di conoscenza.



Directory 'conf'

Directory che ospita i **files** di **configurazione** del **server eXtraWay** e di ogni altro **modulo** del sistema.

[Documentazione in allestimento]

Directory 'context'

E' la **directory** dove trovano collocazione tutti i **files** detti **di contesto**. Essi hanno la finalità di indicare in quali **condizioni** il **server eXtraWay** stia operando sui **documenti** presenti nei diversi **archivi**.

La **directory** contiene i seguenti **files**:

Identificativo File	Contenuto
context.stat.xml	Il file contiene lo stato del contesto corrente ovvero indica quale versione della <i>Struttura Organizzativa</i> stia operando. Essa seguirà un <i>Manuale di Gestione</i> e si rifarà ad un <i>Piano di classificazione</i>. Nel file si indica quale versione di ciascuna di queste componenti viene utilizzata oltre a quella del <i>Technical Reference</i>. Inoltre si indica il nome breve, mnemonico, della struttura che sta operando ed il <i>namespace</i> necessario ad eXtraWay per il trattamento degli allegati.
	<pre><context checksum=""> <object id="struct" name="3D Informatica" version="001.000" date="20040915"/> <object id="classif" name="" version="001.000" date="20040915"/> <object id="gestman" name="" version="003.001" date="20070201"/> <object id="techref" name="ExtraWay Technical Reference" version="000.000.004" date="20030814"/> <object id="namespace" name="http://www.3di.it/ns/xw-200303121136" version="" date="20030312"/></pre>
<id>.<main_ver>.<low_ver>.<extension>	Sono i files che contengono le descrizioni degli id visti nel precedente file context.stat.xml. In sostanza avremo un file per struct, uno per gestman, uno per classif ed in fine uno per techref. Ciascun file sarà caratterizzato dall'etichetta presente nel primo file, seguita dalla versione in due o tre cifre³⁵⁾ ed in fine dall'estensione o da una parte libera ed un'estensione a scelta. Chiaramente di ogni file possono essere presenti più versioni. Nei documenti³⁶⁾ verrà esplicitata quale versione di ciascuna componente era in opera quando il documento ha subito l'ultimo intervento.
<modulo>.devhist.xml	Questi files sono del tutto facoltativi ma consentono agli utilizzatori di comprendere a quale evoluzione siano stati soggetti i vari moduli. Nella fattispecie verrà indicata la versione, la data di emissione, il fatto che possa essere deprecata e tutti gli interventi apportati in tale versione.
	<pre><development> <release module="xw" version="021.002.001.000" date="20090409"> <descr> * Corretto il comportamento dei flussi sul registro (xreg) in modo da evitare files nulli e non farsi bloccare dalla loro eventuale presenza * Corretto effetto collaterale prodotto dalla versione 21.1.4.* che impediva l'esportazione documenti con allegati. * Corretto errore nel trattamento dei files '.xrx' che in alcuni casi provocava registrazioni in files errati e crash del server. ... </descr> </release> ... <release module="xw" version="021.001.003.116" date="20080718" type="deprecated"> <descr> ... </descr> </release> </development></pre>

Directory 'db'

E' la **directory** naturalmente preposta a contenere gli **archivi** che, di fatto, possono in realtà essere collocati ovunque. Dal momento che essa rappresenta un **default** si può però fare affidamento su questo **comportamento**.

Come presumibilmente è noto al lettore, ogni **archivio eXtraWay** è accessibile per mezzo di un suo **identificativo logico**, ovvero un'**etichetta** che ne consenta l'identificazione. In questo modo l'applicazione non ha alcun bisogno di conoscere alcunché della **collocazione fisica** ed organizzazione dell'archivio, gli sarà sufficiente conoscere l'etichetta, l'**host** che ospita il servizio³⁷⁾ e la **porta socket** su cui tale **servizio** opera³⁸⁾.

A questo punto l'**applicazione** decide di accedere ad un **archivio** detto myarc ed il **server** col quale si sarà instaurato il **dialogo** opera come segue:

1. In prima istanza consulta il **file** xw.ini per stabilire se in esso vi sia l'**accoppiamento** tra tale **etichetta** ed un **percorso fisico d'archivio**.

2. Se il primo **test** non va a buon fine, compone il **nome** di **default** secondo questa **specifica**:

1. Data la **directory** degli **eseguibili**, si passa alla **directory** `.. \db`. In essa si accede ad una **directory** `myarc` ed in essa si cerca la presenza del file `myarc.stat.xml` ³⁹⁾

Directory 'logs'

E' la **directory** candidata a contenere tutti i **logs** prodotti dal **server eXtraWay** e da tutti gli altri **moduli** appartenenti alla **piattaforma**.

Nel leggere il seguente schema si ricordi che i **logs** sono di due **tipi**:

Ciclici	Sono i logs che sono destinati a sparire da soli quando sono vetusti. La loro stesura può essere configurata per indicare la dimensione di ciascun file ed il numero complessivo dei files. Quando il file di log ciclico raggiunge la sua soglia dimensionale parte un meccanismo di rimozione/rinominazione. Il file con numerazione maggiore viene rimosso, tutti gli altri vengono rinominati aumentando di 1 la propria numerazione e viene generato un nuovo file di log di base.
Statici	Sono logs che, diversamente da quelli ciclici, non sono destinati ad essere rimossi per alcuna ragione. Essi solitamente hanno una denominazione che ha una componente cronologica così da non realizzare un unico file di dimensioni incontrollate

Nella fattispecie vanno presi in esame i seguenti **files**:

Identificativo File	Contenuto
xw[1,2,3...].log	Questi sono considerati logs di servizio e sono ciclici . In questo file il server eXtraWay e le librerie dinamiche da esso utilizzare registrano le proprie attività secondo una sintassi esplicitata in altra documentazione ⁴⁰⁾ . Ogni riga contiene un <i>timestamp</i> completo di millisecondi che consente di identificare il momento cui essa si riferisce, il <i>Process ID</i> del processo che ha causato la registrazione, un'etichetta tra parentesi quadrate che indica che tipo di registrazione sia e, di seguito, i dettagli dell'operazione. In particolare si ricorda che le righe [Q] indicano l'inizio di un'attività richiesta da un'applicazione al server mentre le righe [A] indicano la fine di tale attività. Le prime rendono evidente il comando e l'archivio in cui si opera, le ultime il tempo necessario per condurre a termine l'operazione precedentemente richiesta.
xw<annomese>.log	Viene detto il <i>Log Giornale</i> . In esso vengono indicate solo le operazioni più importanti quali avvio e spegnimento dei servizi principali e secondari, avvio di attività di acquisizione documenti tramite <i>WatchDoc</i> , abbattimento di dati ed indici e loro ricostruzione. Vengono inoltre segnalati gli allegati per i quali non risulta più che vi sia alcun riferimento in documenti dell'archivio cui essi appartengono.
xreg<annomese>.log	Viene detto il <i>Log di Registro</i> . In esso vengono registrate in modo capillare tutte le operazioni che comportano attività modificanti per l'archivio quali inserimenti, modifiche, cancellazioni, e le altre principali attività registrate anche nel <i>Log Giornale</i> . A differenza di esso, però, in questo registro vengono mantenute tutte le copie pregresse dei documenti che hanno subito modifica e vi è un maggior livello di dettaglio sulle operazioni svolte. In questo modo, se necessario, si possono operare manualmente o automaticamente attività di <i>Roll Back</i> e tutte le più importanti attività di verifica.
wd_journal[1,2,3...].log	File di log per le attività di <i>WatchDoc</i> per mezzo delle quali vengono acquisiti nuovi documenti, modificati documenti esistenti ed eventualmente cancellati documenti tramite la procedura suddetta. Le registrazioni di questo tipo di attività si trovano tanto in <code>xw.log</code> quanto in <code>wd_journal.log</code> , anch'esso ciclico se pure riportano almeno in parte informazioni diverse. Consente di monitorare il buono o il cattivo funzionamento di queste attività che sono tipicamente non <i>presidiate</i> .
rmattach[1,2,3...].log	Là dove presente e configurato, in questo log ciclico vengono registrate le attività di rimozione di quegli allegati che non risultino più o non siano mai stati referenziati da un documento. L'attività comporta impostazioni a livello di server eXtraWay e di archivio.
TextAgent[1,2,3...].log	Là dove presente e configurato, in questi log ciclici vengono registrate le attività dei servizi FCA ⁴¹⁾ ed FCS ⁴²⁾ .
xcrwl[1,2,3...].log	Se presente ed utilizzato, questo log ciclico registra le attività specificamente svolte dall'estensione <i>Crawler</i> di eXtraWay Server.

Directory 'collect'

Nella **directory** `collect` trovano posto le **raccolte** che gli **utenti** possono realizzare. Una raccolta è l'equivalente di una **selezione** o una combinazione di più **selezioni**, o ancora un elenco di **documenti** che un operatore ha facoltà di raggruppare a suo piacimento per **categoria**, **contenuto**, **tema** o qualsiasi altra ragione.

Nel parlare di **documenti** e **selezioni** ci si riferisce non propriamente ai documenti veri e propri bensì ai loro **identificatori**, ovvero i **numeri** di **record** che essi hanno entro ogni singolo **archivio**.

Nelle **raccolte** possono confluire **identificativi** di **documenti** appartenenti ad **archivi** diversi.

Le **raccolte** possono essere di due tipi, **pubbliche** o **private**.

Tipo Raccolta	Funzionalità
Pubblica	Realizzata da un qualsiasi utente è messa a disposizione di tutti gli utenti del sistema i quali possono avvalersene ma non possono effettuarvi modifiche. L'utente <i>proprietario</i> di qualsiasi raccolta ha facoltà di renderla pubblica ovvero di riportare la raccolta al rango di privata per intervenire su di essa e pubblicarla nuovamente.
Privata	Le raccolte private di un utente sono visibili solo a lui che è il solo che può compiere interventi sopra aggiungendo o togliendo documenti.

Una **raccolta** può essere utilizzata per accedere al suo **contenuto** esattamente come si fa nel consultare l'**esito** di una **selezione**,



ovvero come strumento di **raffinamento** di una **selezione** indicandola entro una **frase di ricerca**. In tal caso della raccolta indicata si utilizzeranno solo i **documenti** che siano in essa presenti e riferibili all'**archivio** sul quale si sta facendo la selezione. I dettagli sulle modalità d'**utilizzo** delle **raccolte**, pubbliche e private, nell'ambito della **ricerca** saranno meglio chiariti nel capitolo dedicato appunto alla **sintassi** di ricerca.

Directory 'tmp'

Il **server eXtraWay**, così come tutti gli altri moduli della piattaforma, necessita di un'area nella quale collocare **files temporanei** per tutte le **attività** proprie del server ed in particolare per le attività di **ricerca**.

Entro la **directory identificata**, eXtraWay Server crea a sua volta una directory **hwtemp** al fine di avere pieno **controllo** sui **files** in essa presenti così da poter avviare processi in background che si occupino di regolare la crescita dei files in tale directory e provvedere alla **rimozione** di quelli non più necessari ⁴³⁾.

La **directory prescelta** è quella che corrisponde a quella configurata come segue:

- **Directory** esplicitamente dichiarata nel file **xw.conf.xml**;
- **Variabile ambientale** TEMP ;
- **Variabile ambientale** TMP ;
- **Variabile ambientale** TMPDIR ;

In assenza di una valida **configurazione**, il server assume nella **directory tmp** la sua **directory di default**.

Si noti che le diverse **piattaforme** hanno comportamenti diversi che richiedono qualche attenzione quando non sia stata esplicitamente configurata la **directory** desiderata.

Piattaforma	Accorgimento
Windows	In Windows si deve tener conto di come venga eseguito il modulo. Se eseguito “manualmente” da un determinato utente eXtraWay Server assumerà come variabili ambientali quelle specifiche di utente mentre se il modulo viene eseguito come servizio saranno le variabili ambientali dell'utente SYSTEM ad essere prese in esame. Normalmente l'utente SYSTEM non ha proprie variabili TEMP o TMP ⁴⁴⁾ e quando le ha o le ha acquisite, esse possono essere rappresentate sottoforma di regole composite. La stringa %SystemRoot%, ad esempio, identifica la directory di Windows e così via. Versioni datate dei sistemi operativi Windows, come ad esempio Windows 2000, compiono in maniera non corretta la <i>traduzione</i> di queste stringhe compromettendo quanto interpretato dal server. Per esse si suggerisce di scrivere in chiaro il valore di tali variabili ambientali.
Linux/Unix/Solaris/Aix/Mac	L'impostazione delle variabili ambientali di un utente avviene solitamente a livello di file di risorse della shell utilizzata per accedere al sistema. Ciò comporta che lo stesso utente che acceda al sistema con shell diverse, ad esempio bash e zsh, potrebbero avere impostazioni delle proprie variabili ambientali diverse tra loro. Inoltre non è infrequente che un servizio come eXtraWay, pur operando come uno specifico <i>user</i> del sistema, venga eseguito dall'utente root . Per avere un comportamento uniforme si agisce solitamente sul file xwctl.conf in modo da impostare ed esportare le variabili ambientali necessarie e garantire un corretto comportamento di eXtraWay Sever

Directory 'lazy'

Directory che ospita i **files** da elaborare near on line.

[Documentazione in allestimento]

Directory 'xreg'

Nelle **versioni** più **recenti** del **server** le attività di **registrazione** nel file **xreg<annomese>.log** possono avvenire sia nella modalità canonica, vale a dire per mezzo di una **comunicazione sincrona** tra i **Server Slave** ed il **Server Master**, ovvero riportando su **file system** le attività da svolgere per ridurre ai minimi termini il **colli di bottiglia** derivanti dalla comunicazione sincrona in presenza di elevata **concorrenza d'accesso**.

Il **Server Master**, che ha il compito di effettuare la stesura del suddetto **file** di **log**, consuma i **files** presenti in questa **directory** in **ordine cronologico**.

La **scelta** tra le due **modalità** va compiuta a seconda delle **criticità** dell'**installazione** su cui ci si trova ad operare in quanto la modalità canonica fa ampio uso di **Socket**, la modalità asincrona su file va invece ampio uso del **File System**.

Directory 'wd'

Directory che ospita i **files** da acquisire.

[Documentazione in allestimento]

Architettura HardWare

Introduzione

Il presente documento vuole dare evidenza di quali possano essere gli **scenari hardware** e **software** per soddisfare le esigenze di qualsiasi **progetto** realizzato con **eXtraWay**.

In primo luogo vale la pena di sottolineare che un'**applicazione** realizzata con la **tecnologia ExtraWay** prevede normalmente una **struttura** “3 tiers” ovvero la presenza di tre **strati software** che si occupano di:

- **Presentazione** delle **informazioni** raccolte e **controllo** delle informazioni inserite dall'utente (Web Server);
- **Elaborazione** delle **informazioni** inserite dall'**utente** e **gestione** delle **comunicazioni** con la **banca dati** (Application Server);
- **Gestione** dei **dati** vera e propria (DataBase Server).

Una simile **struttura** si presta naturalmente per consentire una **scalabilità verticale** e, se le **condizioni** lo consentono, anche orizzontale. Il primo ed il secondo **strato software** possono essere scalati orizzontalmente senza limitazioni mentre il terzo, vale a dire il **DataBase Server**, ovvero **ExtraWay**, può essere scalato orizzontalmente solo in alcuni casi.

Va inoltre ricordato che un ulteriore **livello** di **separazione** dei **compiti** può essere ottenuto distinguendo il **Back Office** di un'applicazione dal suo **Front Office** se la **gestione** dei due **accessi** ai **dati** può essere **desincronizzato**.

Ciascuno di questi **concetti** sarà più chiaro al **lettore** nei capitoli successivi.

Di seguito vedremo tutti i possibili **scenari** soffermandoci sugli **aspetti** più **significativi** per ciascuna delle **possibilità** esposte.

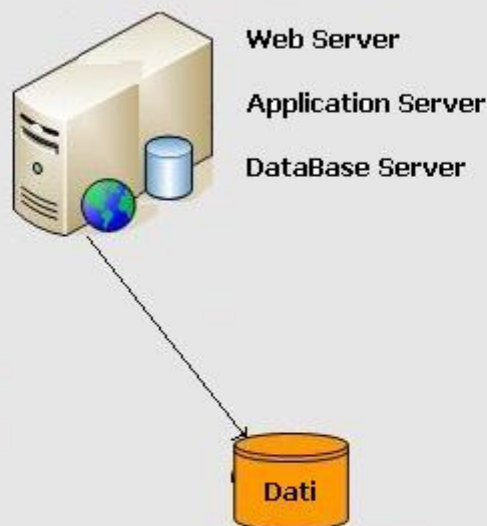
Gli scenari di base disponibili

Si considerano **scenari di base** quelli che offrono il **servizio minimo** indispensabile senza garantire particolari **livelli** di **performance**, di **continuità** di **servizio** e di separazione tra **settore redazionale** e mera **consultazione**.

La rappresentazione dei "Dati" separatamente dalla macchina è solo indicativa. Esclusivamente nei casi di Cluster tale separazione è effettivamente un requisito.

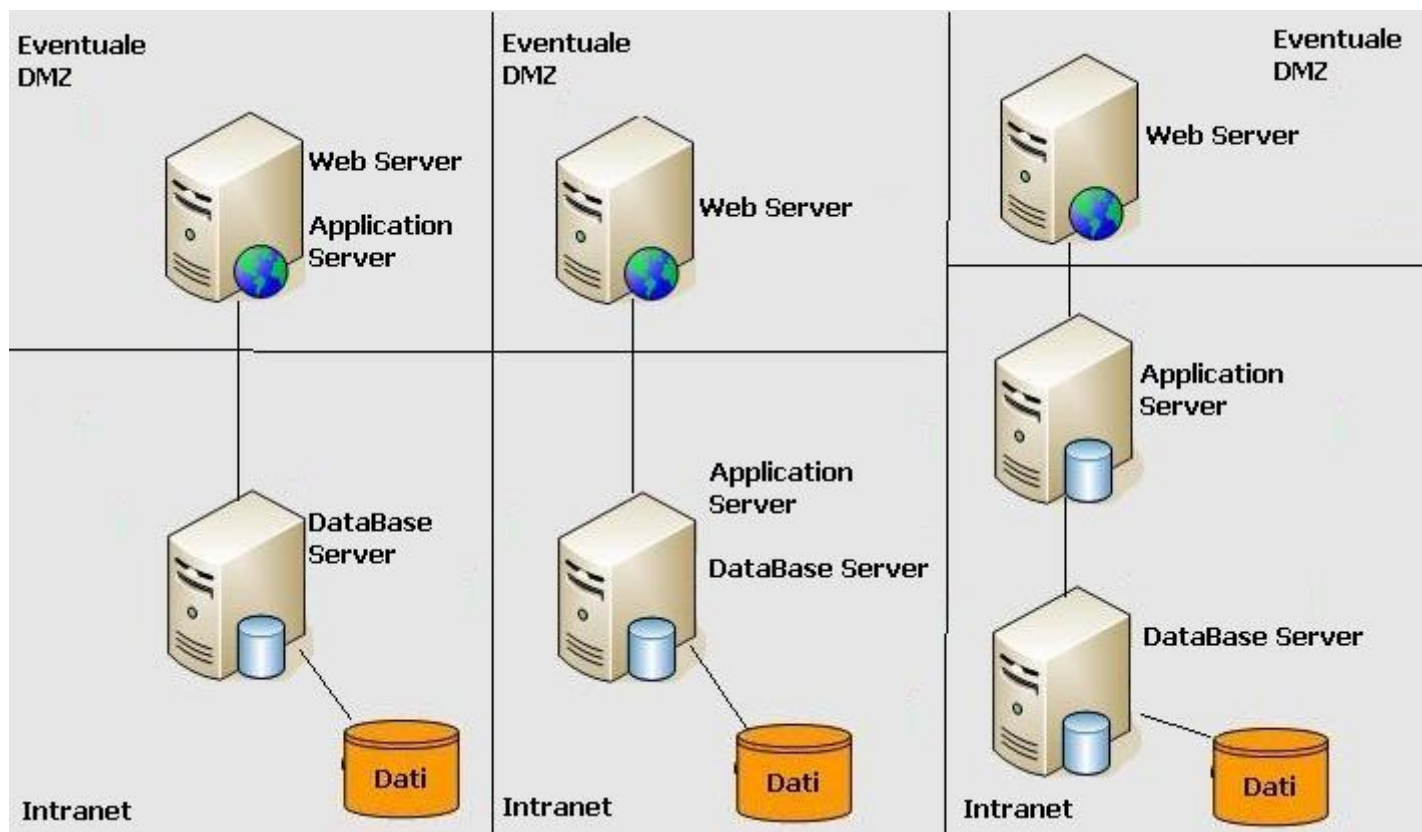
Configurazione Minima

Lo **scenario "minimo"** va concepito con la compresenza sullo stesso **server** di tutti e tre gli **strati software** descritti precedentemente. Non sfrutta alcuna possibilità di distribuire il **carico** di **lavoro** rappresentato dai singoli **strati** scalando l'hardware verticalmente e non garantisce una **continuità** nel servizio in **caso** di **guasto**.



Configurazione Scalata Verticalmente

Deriva direttamente dalla **configurazione minima** e consiste nell'avere due o tre **strati hardware**, non replicati tra loro, che consentano di distribuire verticalmente il **carico** di **lavoro**. Sfrutta quindi la possibilità di distribuire il carico (da una configurazione minima su due livelli ad una "canonica" su tre) ma non garantisce una **continuità** nel **servizio** in caso di **guasto**. Di seguito i **tre esempi** possibili. In due casi si sfruttano solo due **macchine** per distribuire i tre **livelli** di **software** mentre il terzo caso rappresenta la condizione più ampiamente distribuita e quella solitamente adottata quando si decide di compiere una **scalatura verticale** dell'**hardware**.



Scenari Complessi

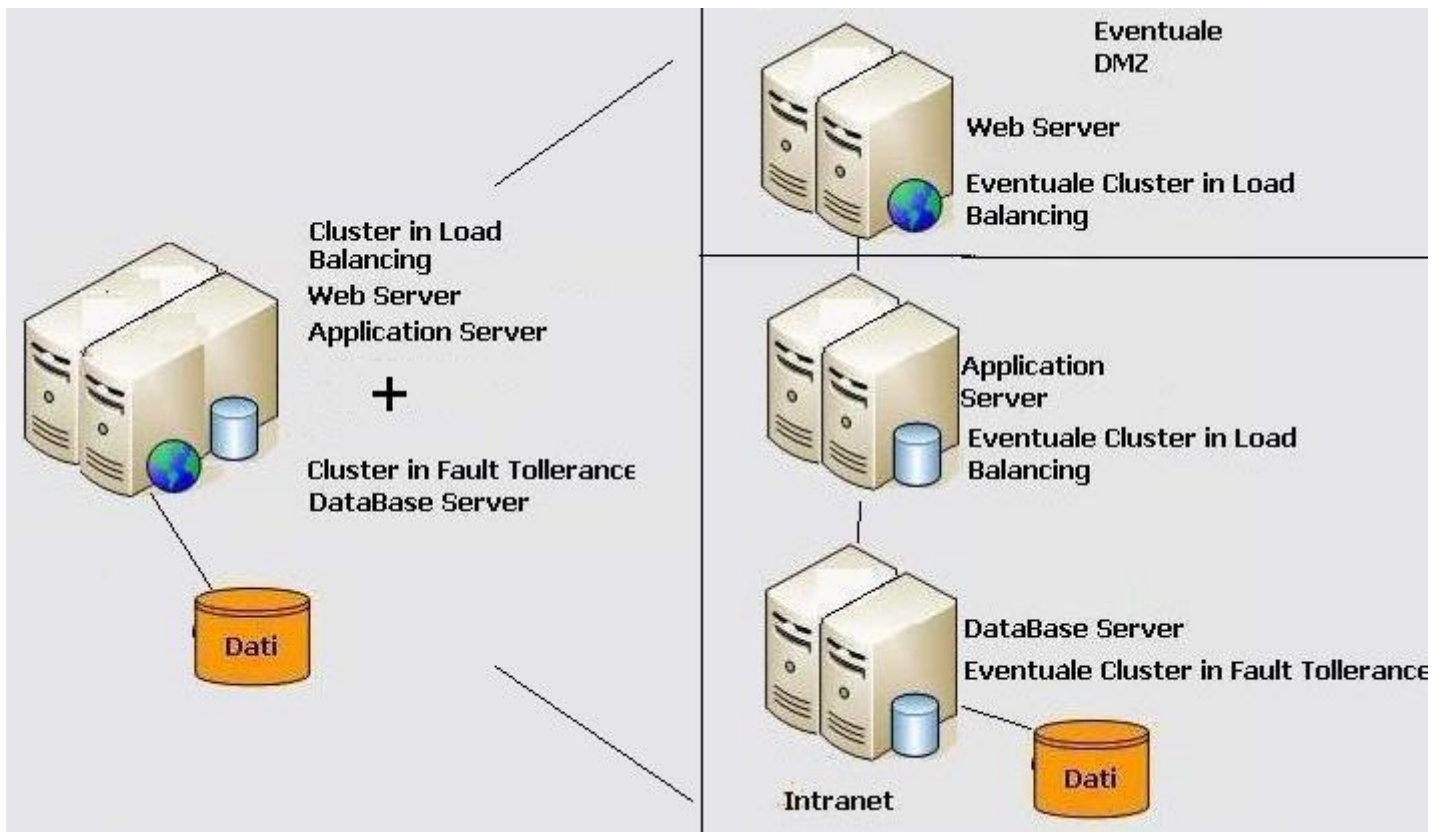
Negli **scenari complessi** trovano posto tutte le diverse **configurazioni** che possono risultare maggiormente garantiste per **prestazioni**, **continuità del servizio** e **separazione** tra **settori**. In particolare si prenderanno in esame i **sistemi** che sfruttano una **tecnologia** di **Load Balancing** e/o **Fault Tolerance** avvalendosi di uno o più **Cluster**.

Configurazione con l'uso di Cluster

Le **configurazioni** che si possono avvalere di **sistemi** di **Cluster** possono essere viste come dirette **derivazioni** degli **scenari di base** descritti precedentemente. In tal caso, in virtù della **suddivisione verticale** degli **strati software** sull'**hardware** posto in gioco, si possono assoggettare a **Cluster** gli strati preferiti. Si può infatti pensare di porre in **Load Balancing** (quindi in cluster di macchine tutte operative) il primo e/o il secondo strato software coinvolto (Presentazione ed Elaborazione) mentre per il terzo strato, la **Gestione Dati**, è necessario fare affidamento ad un sistema di **Fault Tolerance**.

Vedremo infatti che anche in tutti gli **scenari** ad eccezione della separazione tra Back Office e Front Office, la **Gestione Dati** non può essere amministrata correttamente solo con una configurazione in **Load Balancing**.

Nulla vieta, comunque, che anche in presenza di una sola coppia di **macchine** in **cluster** tra loro, i diversi **strati** di **software** possano essere amministrati separatamente usufruendo della distribuzione di **carico** almeno per i primi due di essi. L'immagine seguente vuole dare evidenza del fatto che le **macchine** possono essere poste in **cluster** quale che sia la **scalatura verticale** scelta secondo le ipotesi avanzate nel precedente capitolo. In particolare l'esempio di destra indica come "eventuale" ogni **cluster** in quanto si può decidere a quale livello esso è necessario ed applicarlo solo in tal caso. In caso di Cluster sul **DataBase Server** l'adozione di unità dischi esterna condivisibile dalle due macchine è pressoché inevitabile.



Configurazione separata tra Back Office e Front Office

In una simile **configurazione** c'è la massima disponibilità di scalare l'**hardware** coinvolto. Questa configurazione deriva dalla precedente ma **replica** su due **serie** di **macchine** i **ruoli redazionali** (**Back Office**) e di **consultazione** (**Front Office**) con una **sincronizzazione** tra i due **ambienti** da definire.

Lo **scenario** prevede quindi che il **Back Office**, strutturato in una qualsiasi delle precedenti **modalità** descritte, con o senza **Cluster**, mantenga aggiornato con uno **strumento** di **distribuzione** (**Spreader**) una batteria di macchine di **Front Office** (idealmente da due in su) che non hanno la necessità di essere poste in cluster. Sarà il procedimento di **distribuzione**, avvalendosi di un **processo** detto **arbiter** a sincronizzare tutte le macchine del **Front Office**. L'**arbiter**, a sua volta, potrà esporre all'utenza le sole **macchine** effettivamente sincronizzate.

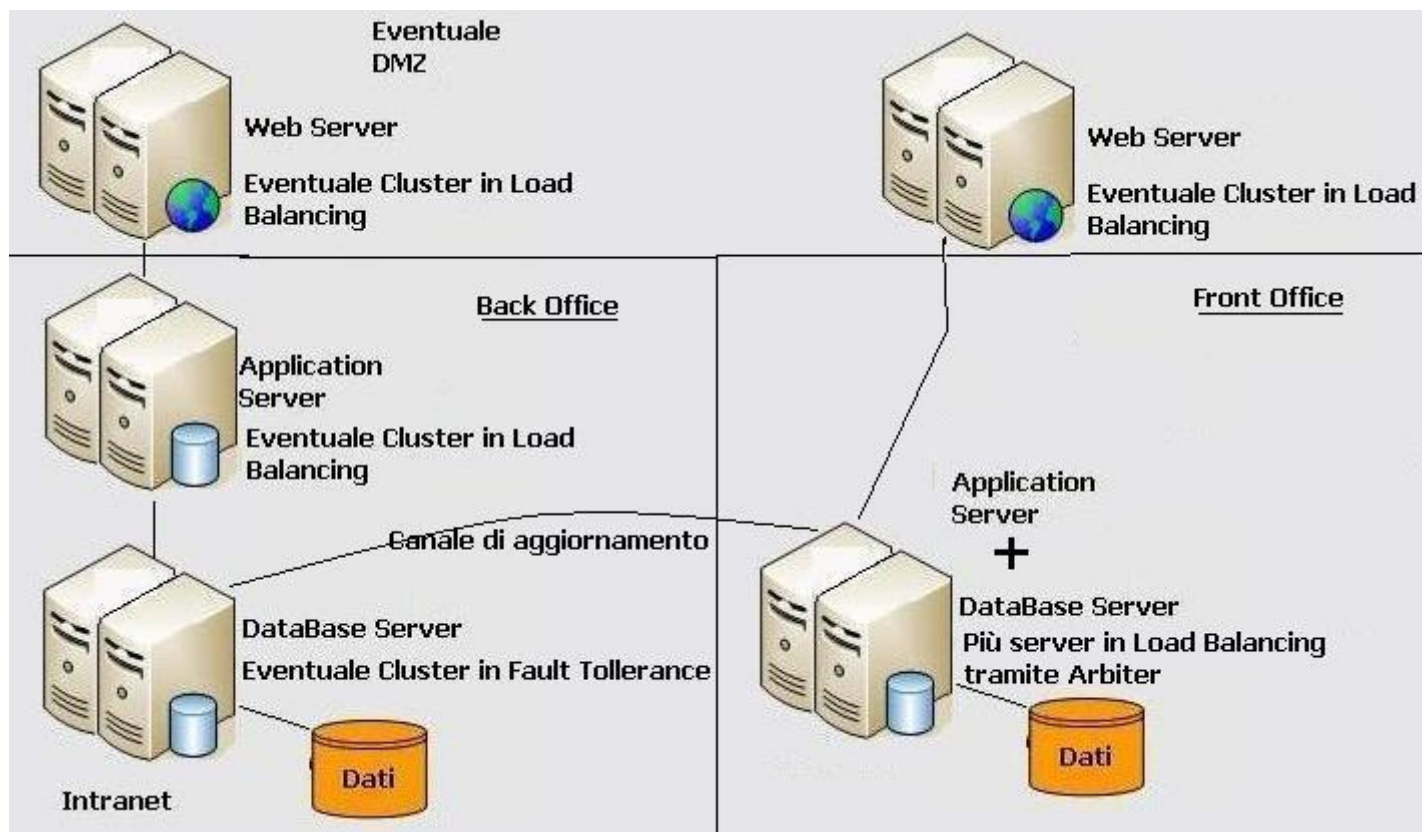
In sostanza, per il nostro esempio, il Back Office può essere strutturato in una qualsiasi delle modalità sin ora viste, quella prescelta è solo esemplificativa.

Oltre a quanto detto bisogna sottolineare che una simile configurazione si può prestare solo per archivi nei quali la fase redazionale differisca da quella di pubblicazione, solitamente legata ad una sorta di validazione dei dati introdotti o modificati, in quanto l'attività dello Spreader può essere ragionevolmente avviata solo alcune volte al giorno e non può essere considerato uno strumento adatto a mantenere costantemente aggiornato un Front Office "near-on-line".

L'**esempio** che segue deriva dall'applicazione di un **Cluster** completo, scalato nell'**hardware** sui tre **livelli software** (parte di sinistra).

A destra vediamo invece l'**aspetto** che avrebbe il **front office** con l'uso di **Web Server** distinti ed una batteria di due o più macchine tutte contemporaneamente in grado di fungere tanto da **Application Server** quanto da **DataBase Server** grazie all'intervento di un **Arbiter**.

La prima immagine rappresenta il caso tipico mentre la seconda mostra un caso più complesso nel quale si sia voluta compiere un'eventuale **suddivisione** del **carico di lavoro** separando anche lo **strato software** inerente l'**Application Server**.



Nella seconda immagine, fondamentalmente equivalente alla prima, si introduce solo l'**Application Server in Load Balancing**.



Note sullo Spreader

Come detto in precedenza, una separazione tra **Back Office** e **Front Office** può essere ottenuta in diversi modi ed esistono senza alcun dubbio strumenti in grado di compiere una **replicazione** più o meno "complessiva" dello stato di **dati** ed **indici** (gli archivi) dalla macchina del Back Office a quelle del Front Office, assumendo che esse siano due macchine in **Load Balancing**. Condividere via **NFS** o altro strumento di rete i **dischi** di un server verso altri server per ampliare la batteria di **macchine** a disposizione dell'utenza è una pratica percorribile ma che, in particolare a fronte di un **utenza** di ampie dimensioni, risulta scarsamente performante.

A tale scopo 3D Informatica ha realizzato uno **strumento** denominato **Spreader**, avente lo scopo di distribuire alle macchine di un Front Office uno o più **archivi** provenienti dal Back Office operando in modo differenziale e, collaborando con l'**Arbiter** che compie la **distribuzione** di **carico** tra le macchine Front, mantenendo aggiornato lo **stato** delle **macchine** in modo sempre allineato.

Cerchiamo quindi di spigarne il comportamento tramite un esempio. Abbiamo un **server** di **BackOffice** (BO1) e 5 server di **Front Office** (FO1, FO2, FO3, FO4 ed FO5).

Quando l'**amministratore** decide di compiere l'**aggiornamento** dal Back Office al Front Office, ovvero tramite un **automatismo** a tempo, lo **Spreader** controlla richiedendo le **informazioni** alle macchine del Front Office, quale sia il loro stato di aggiornamento. Supponiamo in un primo momento che esse siano tutte aggiornate al giorno precedente.

Lo **Spreader** richiede al **Server BO1** di valutare le differenze intercorse in termini di **dati** ed **indici** dalla **data** ed **ora** (time stamp completo sino ai millisecondi) di **ultimo aggiornamento** "comune". Il **file** prodotto, chiamato **delta** viene in seguito usato per l'**aggiornamento** delle **singole macchine**.

L'**aggiornamento** avviene secondo questa **tecnica**.

Lo **Spreader** chiede all'**Arbitro** di porre in "stand by" **FO1** che da quel momento non verrà più usata per distribuire il **carico** di **lavoro** richiesto dall'**utenza**. Poi invia a **FO1** il **delta** e gli chiede di applicarlo. Una volta applicato il **server FO1** viene mantenuto in **stand by**.

Poi fa altrettanto con **FO2** e con **FO3**. Avendo superato "la metà" dei server disponibili sul Front Office, lo **Spreader** richiede all'**Arbitro**, con un'operazione atomica, di porre in "stand by" le **macchine FO4** ed **FO5** e di riportare "on line" **FO1**, **FO2** ed **FO3** che essendo tutte aggiornate ed in "maggioranza" possono meglio servire l'utenza.

Di seguito provvede a compiere l'aggiornamento su **FO4** ed **FO5** richiedendo che vengano poste "on line" appena aggiornate per rimpinguare il **pool** di **macchine disponibili** all'utenza.

Se nel **procedimento** descritto una **macchina** risulta non raggiungibile, spenta, precedentemente posta in "stand by" e non rimessa "on line" a causa di un **errore** di **aggiornamento**, lo **Spreader** stabilisce quale sia la **combinazione migliore** di **macchine** da lasciare "on line" per garantire il miglior **servizio** al **miglior stato d'avanzamento**. Se le **macchine FO** sono allineate in modo diverso, il **delta** che viene calcolato comprende il **materiale** necessario a condurle tutte allo stato d'**aggiornamento** corrente anche a costo di applicare aggiornamenti già applicati in parte (senza con questo impattare sulla **validità** degli **archivi**).

Estensione degli Scenari, il Disaster Recovery

Gli **scenari**, semplici e complessi, sin ora descritti consentono di costruire una **struttura** con diversi **livelli** di **sicurezza** già in essa. Ad esempio, un **Back Office** separato e configurato con un **Cluster** è ragionevolmente solido ed ha un **back up** naturale nel **Front Office**. Una **struttura** che preveda **più Front Office** ha in essa una **sicurezza** intrinseca di continuità del **servizio**.

Se a questi **scenari** vogliamo affiancare un ulteriore livello di **sicurezza** entriamo nell'ambito del **Disaster Recovery** per il quale valgono le considerazioni già fatte negli scenari descritti. Anche in quel caso possono essere ipotizzati scenari semplici e complessi per il **nodo "secondario"** indipendenti dalla **configurazione** data a quello "primario".

Esso consiste, solitamente, nella **replicazione** dell'**hardware** (nella forma più opportuna) e di tutti i **dati** disponibili ed allineati in una **struttura** differente. Per struttura, in questo caso, facciamo riferimento fisicamente ad un luogo diverso, servito differentemente sia dal **punto di vista elettrico** che per ogni altro aspetto **logistico**.

Quello che può essere replicato in questa differente **struttura** è un qualsiasi **scenario** "minore o uguale" a quello primario che può essere mantenuto allineato con degli **automatismi** con **cadenza giornaliera** o con quella ritenuta più opportuna.

Gli **strumenti** a disposizione, dal trasferimento e l'applicazione di un **backup** all'uso dello **Spreader**, consentono di avere un **allineamento** ragionevolmente "fresco" della copia presso la **struttura secondaria** ed essa può essere concepita come sola **salvaguardia** dei dati (quindi una copia fedele del Back Office consultabile) sia come una completa **replicazione** della **configurazione primaria**.

Anche se per il **nodo primario** si fosse scelta una **configurazione minima** in essa dovrebbe trovare posto per lo meno uno **spreader** che provvederà a mantenere aggiornata una delle **replicazioni** che seguono.

La **replicazione** dello **scenario** dovrebbe essere **integrale**. La replicazione integrale va vista come la **riproduzione**, presso diversa **struttura**, di una **configurazione** minima o scalata come dagli scenari descritti. Essa può essere una **versione** "ridotta" (ovvero priva di scalatura orizzontale o verticale o con una scalatura ridotta) di quella primaria ma comunque pienamente funzionante.

Tramite **spreader** (considerando il BackOffice dell'installazione "secondaria" al pari di un Front Office da aggiornare da parte del nodo "primario") si può mantenere aggiornato il secondo **nodo** ed in esso si provvederà, se anche il **nodo secondario** è scalato verticalmente, ad aggiornare il proprio **Front Office** avvalendosi di un'installazione "**locale**" dello spreader.

In caso il **nodo secondario** venga chiamato in causa per indisponibilità del **primario** esso prevederà un **ambiente** completo che consentirà di proseguire tanto nell'**attività redazionale** quanto in quella di fruizione del **sistema** mantenendo eventualmente distinte le due **realità**.

Si tratta attualmente dello scenario più completo ed esaustivo.

Possono essere presi in esame altri scenari in cui il **nodo secondario** ha una replicazione parziale del **nodo primario**, replicando in esso il solo **Back Office** o il solo **Front Office**.

Conclusioni

Concludendo, gli scenari che si possono realizzare sono stati ampiamente descritti e possono essere ottenuti combinando in vario modo le possibilità esposte. La scelta sulla configurazione più opportuna deve tenere in considerazione due fattori evidenti: la scalabilità orizzontale garantisce di adeguarsi alle performance richieste dal sistema in virtù del carico di lavoro previsto sulla base dell'utenza che ne farà uso, potenzialità in parte ottenibile anche con la scalabilità verticale, ma rispetto a quest'ultima offre tipicamente maggiori risultati e da garanzie di continuità nell'erogazione del servizio che la sola scalabilità verticale non può offrire.

L'unico reale vincolo si ha nella realizzazione di un Cluster per il DataBase Server del BO (o globale se non v'è distinzione tra le due parti) in quanto non è possibile adottare una configurazione in Load Balancing ma solo in Fault Tolerance per lo meno sino a quando alcune limitazioni esistenti nella piena accessibilità delle risorse condivise non saranno superate dei produttori di hardware e software di base.

Questo, per lo meno, sulla base dell'esperienza sino ad ora accumulata in proposito.

Gli sviluppi futuri

Se il documento ha raggiunto lo scopo, il lettore avrà ora un quadro sufficientemente chiaro di cosa si possa realizzare con il software della 3D Informatica ed in particolare saprà comprendere quale tra gli scenari descritti potrà soddisfare pienamente le proprie esigenze.

Oltre a ciò il lettore avrà anche compreso come il modificarsi del carico di lavoro o delle esigenze di fruizione delle informazioni nell'arco del tempo non trovino in eXtraWay un limite bensì un valido alleato, pronto a modellare le proprie capacità al servizio del più importante degli obiettivi che accomunano 3D Informatica e coloro che ne hanno scelto la tecnologia: fornire un servizio solido e performante che unisca un'ampia fruibilità delle informazioni alle tecniche più adatte per garantire la massima sicurezza dei dati stessi.

Questo è quanto abbiamo voluto fare sino ad ora ma certo possiamo fare molto di più. Per venire ulteriormente incontro alle esigenze della clientela ci sono progetti che riteniamo degli di essere presi in esame, anche se parte di essi non vedrà la luce se non tra diverso tempo.

Tra essi ci preme sottolineare:

Hot Backup

Tra le esigenze che ci sono state spesso esposte c'è quella di poter compiere backup senza causare una effettiva interruzione dei servizi, garantendo quanto meno la consultazione delle informazioni contenute nei Data Base.



Una simile modalità consiste in un congelamento effettuato direttamente dallo strato più basso, ovvero il DataBase Server, delle sole funzionalità di modifica così da consentire la copia e quindi il salvataggio della fotografia del Data Base in un dato momento senza con questo inibire la consultazione. In uno scenario ideale un parco utenti anche molto vasto è in realtà composto da una minoranza che ha funzioni di redazione ed una maggioranza che invece si limita alla fruizione dei Data Base. Lo scenario proposto consentirebbe quindi di limitare l'operato solo di una minoranza.

Scalabilità Orizzontale del Tier DataBase Server

Questo è tra i progetti più ambiziosi ma, al contempo, uno dei più importanti. Nei diversi scenari che sono stati presentati, specie in quelli complessi, è stato più volte posto in evidenza come la scalabilità verticale sia sempre garantita mentre quella orizzontale si riferisca esclusivamente ai primi due livelli del software in quanto attuali limiti tecnologici impediscono di garantire scalabilità sul livello che insiste sui Data Base. Al fine di superare questo vincolo con il pieno rispetto delle diverse piattaforme su cui eXtraWay può essere installato (ed in considerazione di costituire anche installazioni scalate su hardware di diversa tipologia che si avvalgano di Sistemi Operativi diversi) esiste un progetto di realizzazione di un servizio di sincronizzazione e locking delle basi dati da installare parallelamente ad eXtraWay su tutti i server coinvolti nel cluster. A tale servizio verrebbe demandato il compito di garantire un accesso sempre corretto e, là dove necessario, esclusivo unendo le esigenze del Back Office con la grande flessibilità del Front Office. Ad esso può affiancarsi, ulteriormente, un intervento sull'Arbiter, che comunque sarebbe una componente indispensabile in un simile scenario, perché le attività redazionali vengano distribuite equamente su tutti i server per ottenere la miglior distribuzione di carico ovvero concentrate su un singolo server del cluster, il principale di essi, per ottimizzare proprio il compito del servizio di sincronizzazione. Una volta realizzato questo servizio le possibilità di scalatura delle configurazioni eXtraWay diverrebbe, di fatto, senza limite.

Servizio di Content Server

Le applicazioni documentali quali quelle che ci si propone di realizzare con eXtraWay fanno ampio uso di documenti informatici e digitali da considerarsi come allegati dei singoli record degli archivi. Essi possono essere di dimensioni contenute come di ampie dimensioni così come possono essere in numero esiguo come in quantità ragguardevoli. L'esperienza insegna che solitamente essi rappresentano una parte imponente dell'intero Data Base, per dimensioni ed aspetti quantitativi. Parimenti di questi file si può dire che:

- Nella stragrande maggioranza dei casi non possono e non devono essere mai cambiati, al limite sottoposti a versioning, operazione che ne deve comunque garantire la reperibilità in originale. Essi possono quindi essere sottoposti a Backup meno frequenti in quanto destinati ad essere invariabili ma vengono ad appartenere sempre e comunque ai Backup di tutto l'archivio.
- A seconda della natura di tali files le applicazioni possono aver necessità di sfruttarne delle trasformazioni, basti pensare ad un file .PDF non modificabile come versione pubblica di un documento effettivamente registrato in formato Microsoft Word oppure un'immagine ad alta risoluzione della quale si voglia mostrare una versione ridotta (thumb nail) ovvero una versione non sfruttabile perché modificata con un WaterMark. Ciascuna di queste conversioni dev'essere pianificata a monte perché essa possa essere sfruttata applicativamente.

Ciascuno di questi casi viene attualmente gestito da strumenti esistenti che vanno dalla politica di distribuzione degli allegati su disco, alla gestione delle loro trasformazioni con procedure off-line alla realizzazione di complesse modalità di Backup per ottenere risultati parziali. Con la realizzazione di un simile strumento esso può essere sottoposto a proprie politiche di Backup e replicazione, può compiere autonomamente ed a run-time le conversioni richieste rendendole estremamente dinamiche ed adattabili quindi al variare delle esigenze applicative senza dover compiere attività di manutenzione/trasformazione del Data Base. A sua volta esso può rappresentare un ulteriore grado di scalabilità orizzontale del tier inerente i DataBase con un ulteriore beneficio sia prestazionale che architetturale.

Glossario, Acronimi e Note

Questo documento descrive il significato dei termini più frequentemente utilizzati nella documentazione inerente il server e la piattaforma documentale **eXtraWay**. La conoscenza dei significati di questi termini può essere fondamentale per la corretta comprensione della documentazione.

DocWay	Applicazione di Protocollo Informatico e Gestione Documentale realizzata facendo uso della Piattaforma eXtraWay
eXtraWay	Può riferirsi sia al Server eXtraWay quanto alla Piattaforma. Più frequentemente riferito alla Piattaforma
Master	Processo principale che genera processi figli. Non assolve ad alcun altro compito se non la stesura del registro delle attività svolte. Controlla le licenze e genera processi <i>Slave</i> in corrispondenza di nuove connessioni
Piattaforma eXtraWay	Descrive il complesso di moduli binari, componenti Java e files di configurazione che compongono tutto il <i>sistema</i> eXtraWay. Con tale <i>sistema</i> è possibile realizzare qualsiasi applicazione documentale o far uso di applicazioni documentali già elaborate e distribuite come prodotti quali DocWay
PID	Process IDentificator. Numero univoco per mezzo del quale ciascun processo, che si tratti del <i>Master</i> o di uno <i>Slave</i> , viene identificato. Assume una particolare importanza quando si analizzano i logs.



Platform Independent	Dicesi di codice, dati, moduli eseguibili compilati o in linguaggio interpretato che possono essere sfruttati, letti, eseguiti, indipendentemente da sistema operativo ospitante
Server eXtraWay	Cuore della Piattaforma eXtraWay. XML Native DataBase Engine
Slave	Processo figlio del <i>Master</i> che dialoga, in rapporto “uno ad uno”, con un client e risponde alle sue richieste. Al termine delle sue attività, quando la comunicazione socket col client si interrompe, esso si chiude
xw	Il nome “semplice” che viene solitamente attribuito al Server eXtraWay

1)

Normalmente si fa uso di *Internet Information Server* sulle piattaforme Windows e di *Apache Web Server* sulle piattaforme Linux/Unix ma la genericità di questo componente non pone seri limiti alla sua sostituzione con prodotti equivalenti

2)

Sono state fatte anche altre esperienze con strumenti simili quali *JBoss* o *Resin*. La natura di questo strato software è sufficientemente versatile per poter ritenere che il porting verso altri *Java Container* sia un processo di evoluzione non critico

3)

Il server eXtraWay e tutti i moduli a corredo sono stati realizzati per diverse distribuzioni tra le quali, ovviamente, quelle più diffuse sul mercato. L'applicazione su una distribuzione nuova consiste nella ricompilazione del codice ed un eventuale tuning

4)

Sperimentale

5)

E' in fase di realizzazione un'installazione dimostrativa per Linux

6)

Questa è l'unica directory il cui nome non è obbligatorio. In installazioni molto datate assume alle volte il nome prg ma in generale può avere un qualsiasi altro nome purché risulti evidente il significato del suo contenuto e sia collocata nel posto giusto

7)

E' buona norma che più archivi non condividano la stessa directory d'origine

8)

4859 per default

9)

seguendo metodiche diverse da sistema operativo Windows a sistema operativo Linux/Unix

10)

tramite `/etc/init.d` o agendo sui **run level** a seconda del sistema operativo

11)

4859

12)

La cosa dovrebbe essere evidente sia eseguendo manualmente il server sia eseguendolo sotto forma di servizio. Se però si accede ad una postazione di lavoro tramite Remote Desktop l'icona del server qualora avviato come servizio non sarà visibile. Per le attività di verifica o di registrazione sarà necessario eseguirlo manualmente

13)

Se il Server eXtraWay viene eseguito su una porta socket diversa dallo standard, oltre all'ID univoco del processo verrà mostrato anche il numero di porta sulla quale esso opera

14)

15)

default: extraway

16)

ovvero degli eseguibili

17)

Nelle installazioni Linux/Unix il default è `/opt/it-3di/extrawa/xw/bin`

18)

default:4859

19)

default:../logs/xwctl.log

20)

default:/usr/local/lib

21)

default:10 ovvero massima priorità

22)

Il server esegue, se presente nell'installazione, anche il modulo xwls, ovvero eXtraWay Log Server. Può avviare anche altri moduli ma in tal caso essi sono fuori dal controllo dello script di avvio/arresto

23)

eXtraWay Log Server

24)

Lo status viene mostrato, automaticamente, dopo l'esecuzione di ciascuno dei precedenti comandi

25)

files con estensione .so

26)

Si noti che il concetto di connessioni illimitate corrisponde alla cifra 998, non ha senso andare oltre e sovente un valore di 20/30 è più che adeguato

27)

La versione Windows mostra un message box che conferma la corretta registrazione

28)

Si noti che solo per la piattaforma Windows qualora il server sia installato ed eseguito come Servizio l'interruzione ed il riavvio



devono avvenire manualmente tramite il Service Control Manager

29)

In particolare, sulle piattaforme Linux/Unix, il variare dell'IP Address primario della macchina causa la perdita della registrazione della stessa

30)

.dll per Windows, .so per Linux/Unix

31)

Il modulo ha estensione .exe nel caso degli eseguibili Windows, .dll nel caso di librerie dinamiche Windows e .so nel caso di librerie dinamiche di altre piattaforme

32)

Solo piattaforme Windows

33)

fondamentalmente selezioni ma anche semplici attività di modifica

34)

siti web, file system

35)

due o tre sequenze di 3 digit separate da punti

36)

Nella *Processing Instruction* che si trova in coda ad essi

37)

default, l'host locale

38)

default 4859

39)

Ogni archivio si identifica per la presenza di un siffatto file. Esso è nella radice della directory in cui si trova l'archivio e con esso devono trovarsi tutti gli altri files dello stesso, da intendersi come file di mappa, titoli ed indici. Da questo punto in successive sub-directories ci si aspetta che trovino posto tutti i dati e gli allegati che, per ipotesi, potrebbero comunque essere collocati altrove.

40)

Link in realizzazione

41)

File Conversion Agent

42)

File Conversion Service

43)

Alcuni sistemi operativi prevedono già comportamenti simili per i files temporanei ma, per poter essere compatibili con qualsiasi piattaforma il server eXtraWay è autonomo in quest'attività

44)

L'uso del modulo hisetup le imposta copiandole da quelle dell'utente che compie l'installazione