

ICCU - OPAC Solr - Architettura

Inera S.r.l.

March 23, 2012

Indice

1	ICCU OPAC Solr Architettura	5
1.1	Componenti hardware	5
1.1.1	Tipologia 1	5
1.1.2	Tipologia 2	6
1.1.3	Tipologia 3	6
1.2	Componenti software	6
1.2.1	Dettaglio versioni software	7
1.3	Visione d'insieme	8
1.4	Architettura frontend	8
1.4.1	Requisiti risorse frontend	9
1.4.2	Tipologie di accesso	9
1.4.3	HA	9
1.4.4	LB	9
1.4.5	Flusso richieste HTTP	10
1.5	Architettura Backend	12
1.5.1	Requisiti risorse backend	15
1.5.2	Architettura Solr	15
1.5.3	Aggiornamento	16
2	Configurazione	21
2.1	Riepilogo path applicazioni	21
2.2	Riepilogo file di configurazione	21
2.3	Configurazione di Solr	22
2.4	Schema Solr	22
2.5	SolrMarc	22
2.6	Configurazione applicativa	23
2.7	Configurazione Z39.50	23
2.7.1	Configurazione record syntax SUTRS	23
3	Procedure	25
3.1	Aggiornamento incrementale	25
3.1.1	00.preparazione	25
3.1.2	01.scarico-e-conversione	26
3.1.3	02.disabilitazione-backend2	26
3.1.4	03.cancellazioni	26
3.1.5	04.inserimento-localizzazioni	26
3.1.6	05.inserimento-documenti	26
3.1.7	06 Ottimizzazione	26
3.1.8	07.abilitazione-backend2	26

3.1.9	08.attivazione-replica	26
3.1.10	09.abilitazione-backend1	27
4	Attività di manutenzione	29
4.1	Avvio e terminazione OPAC	29
4.1.1	Avvio	29
4.1.2	Terminazione	29
4.2	Esclusione nodo frontend da interrogazione	30
4.3	Avvio e terminazione Solr	30
4.3.1	Avvio solr	30
4.3.2	Terminazione solr	30
4.4	Log attività	30
4.5	Backup	30
4.5.1	backup applicazioni software	31
4.5.2	backup indici Solr	32
4.6	Job	32
5	Riferimenti	33

Capitolo 1

ICCU OPAC Solr Architettura

L'OPAC ICCU consiste di:

- una webapp di frontend che viene servita dal web server Apache e dal servlet container Tomcat;
- un backend di indicizzazione che utilizza Apache Solr.

La definizione dell'architettura hardware e software dell'OPAC ha seguito alcune linee guida:

- *high-availability*, alta disponibilità¹;
- *load-balancing*, bilanciamento di carico²;
- nessun disservizio all'utenza internet durante le operazioni di aggiornamento incrementale o totale degli indici.

1.1 Componenti hardware

L'OPAC consiste di quattro server, due per il frontend e due per il backend. I server sono stati acquistati in tempi diversi, hanno destinazioni d'uso diverse e per questo hanno caratteristiche hardware differenti che però possiamo riassumere in tre tipologie:

- tipologia 1, server di frontend
- tipologia 2, server di frontend
- tipologia 3, server di backend per indicizzazione

1.1.1 Tipologia 1

I server della tipologia 1 sono:

Server	Marca	Modello	Serial number	Data acquisto
frontend1	Dell	PowerEdge 2950	7BPC53J	08/10/2007

Le caratteristiche hardware relative ai server sono:

¹da ora in avanti indicata come *HA*

²da ora in avanti indicata come *LB*

CPU	2 x Intel Xeon E5310 @ 1.60GHz
Memoria	8 x 1GB 240pin Fully Buffered DDR2 SDRAM 667 MHz ³ (1.5 ns) PC2-5300
Hard disk	2 x 73.4 GB ⁴ SAS 15krpm
Configurazione RAID⁵	2 dischi in RAID1
Controller RAID⁵	PERC 5/i

1.1.2 Tipologia 2

I server della tipologia 2 sono:

Server	Marca	Modello	Serial number	Data acquisto
frontend2	HP	ProLiant DL360 G6	CZJ00405KX	01/05/2010

Le caratteristiche hardware relative ai server sono:

CPU	2 x Intel(R) Xeon(R) CPU E5530 @ 2.40GHz
Memoria	4 x 2GB DDR3 1333 MHz ³
Hard disk	2 x 73 GB ⁴ SAS 15krpm
Configurazione RAID⁵	2 dischi in RAID1
Controller RAID⁵	

1.1.3 Tipologia 3

I server della tipologia 3 sono:

Server	Marca	Modello	Serial number	Data acquisto
backend1	HP	ProLiant DL360 G6	CZJ015051P	01/05/2010
backend2	HP	ProLiant DL360 G6	CZJ00405K8	01/05/2010

Le caratteristiche hardware relative ai server sono:

CPU	2 x Intel(R) Xeon(R) CPU E5530 @ 2.40GHz
Memoria	4 x 8GB DDR3 1333 MHz ³
Memoria	3 x 2GB DDR3 1333 MHz ³
Hard disk	3 x 146 GB ⁴ SAS 15krpm
Configurazione RAID⁵	3 dischi in RAID5
Controller RAID⁵	

1.2 Componenti software

Le componenti software utilizzate sono:

- CentOS
- haproxy
- keepalived
- apache
- tomcat

- java
- Solr

1.2.1 Dettaglio versioni software

frontend1

componente	versione
CentOS	5.0 32bit
haproxy	1.3.25
keepalived	1.1.15
apache	2.2.3
tomcat	5.5.23
java	OpenJDK 1.6.0_20
jzkitproxysrver	1.0.0

frontend2

componente	versione
CentOS	6.0 64 bit
haproxy	1.4.18
keepalived	1.2.2
apache	2.2.15
tomcat	6.0.24
java	OpenJDK 1.6.0_17
jzkitproxysrver	1.0.0

backend

componente	versione
CentOS	5.5 64bit
tomcat	5.5.23
java	Oracle Java 1.6.0_07
Solr	1.4.0

1.3 Visione d'insieme

Si riporta nella figura qui di seguito la visione d'insieme delle componenti e di come sono collegate tra di loro.

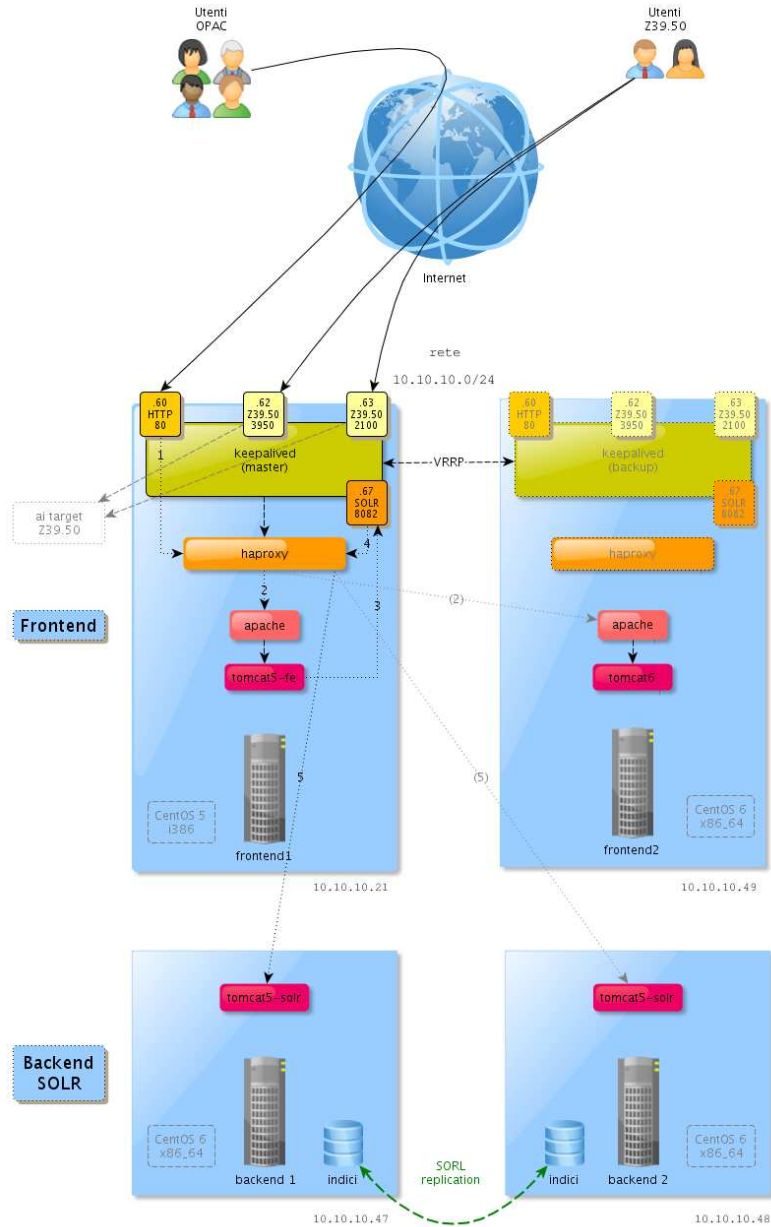


Figure 1.1: ICCU - OPAC Solr - Architettura - Visione d'insieme

1.4 Architettura frontend

Il frontend riceve dagli utenti Internet chiamate che possono essere:

- HTTP⁶

⁶Hyper Text Transfer Protocol

- Z39.50

Ogni servizio fornito dalle macchine di frontend è mappato su un indirizzo IP specifico di rete interna. Il relativo indirizzo esterno non è argomento di trattazione in quanto fa parte della configurazione dei firewall presenti nell'infrastruttura di rete dell'ICCU.

1.4.1 Requisiti risorse frontend

Le risorse richieste dal frontend sono:

Memoria JVM tomcat OPAC	1,5 MB ⁷
--------------------------------	---------------------

1.4.2 Tipologie di accesso

Qui di seguito si riporta una tabella che riassume le tipologie di accesso e i relativi indirizzi IP.

servizio	porta	IP	tipologia accesso
HTTP ⁶	80	10.10.10.60	esterno
Z39.50	3950	10.10.10.62	esterno
Z39.50	2100	10.10.10.63	esterno
Solr	8082	10.10.10.67	esclusivamente interno

Analizzeremo il solo flusso di chiamate HTTP⁶ dirette all'OPAC. Le chiamate Z39.50 verranno discusse in un altro documento.

1.4.3 HA

L'*HA* è affidata a **keepalived**. **keepalived** è configurato in modalità attivo-passivo (*master-backup*).

Il *master* è impersonato dal nodo *frontend1*, il *backup* dal nodo *frontend2*.

Il *master* prende in carico gli indirizzi IP associati a **keepalived** e riportati sopra. *master* e *backup* si controllano a vicenda tramite il protocollo **VRRP** mentre ciascuno controlla (tramite script) che l'istanza di **haproxy** presente in locale sia attiva e funzionante.

Il nodo di *backup* interviene e acquisisce gli indirizzi IP nel caso in cui:

- il *master*, per una causa qualsivoglia, non risulti più raggiungibile o responsivo;
- **haproxy** non risponda correttamente ai check effettuati da **keepalived** tramite script.

1.4.4 LB

Il *LB* è affidato a **haproxy**. Le risorse definite in *load-balancing* su **haproxy** sono:

- gli apache dei nodi *frontend1* e *frontend2*
- i tomcat dei nodi *backend1* e *backend2*

1.4.5 Flusso richieste HTTP

Il flusso delle richieste HTTP⁶ all'OPAC è schematizzato dalla figura qui di seguito:

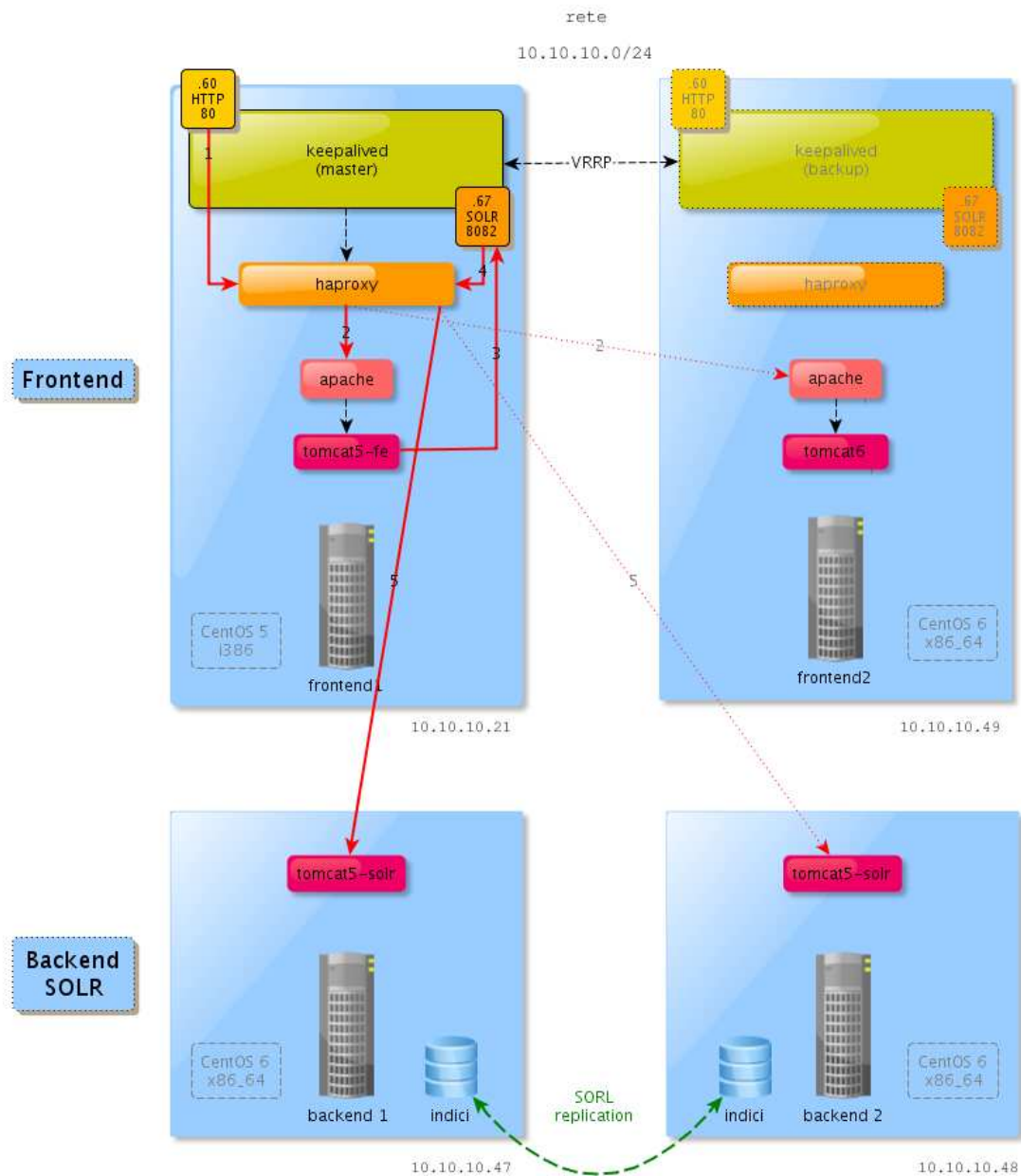


Figure 1.2: Percorso chiamate HTTP a OPAC

e dettagliato da questa sequenza:

- *percorso 1:* le richieste dell'utente internet arrivano sulla porta 80 all'IP (10.10.10.60) ad esse dedicato;
- *percorso 2:* **haproxy** provvede a bilanciare le richieste tra gli **apache** dei due nodi, **frontend1** e **frontend2**; ciascun **apache** punta alla webapp *opacsbn* sul **tomcat** locale; nel caso specifico di figura la richiesta viene inviata al nodo **frontend1**;

- *percorso 3*: il **tomcat** invia le richieste di ricerca all'indirizzo virtuale dedicato a **Solr** (10.10.10.67, porta 8082, in figura attivo sul nodo **frontend1**);
- *percorso 4*: a questo indirizzo risponde **haproxy**;
- *percorso 5*: **haproxy** bilancia le richieste tra i due nodi **Solr**, **backend1** e **backend2**, indirizzando direttamente ai singoli tomcat.

La figura qui di seguito evidenzia un eventuale altro percorso delle chiamate:

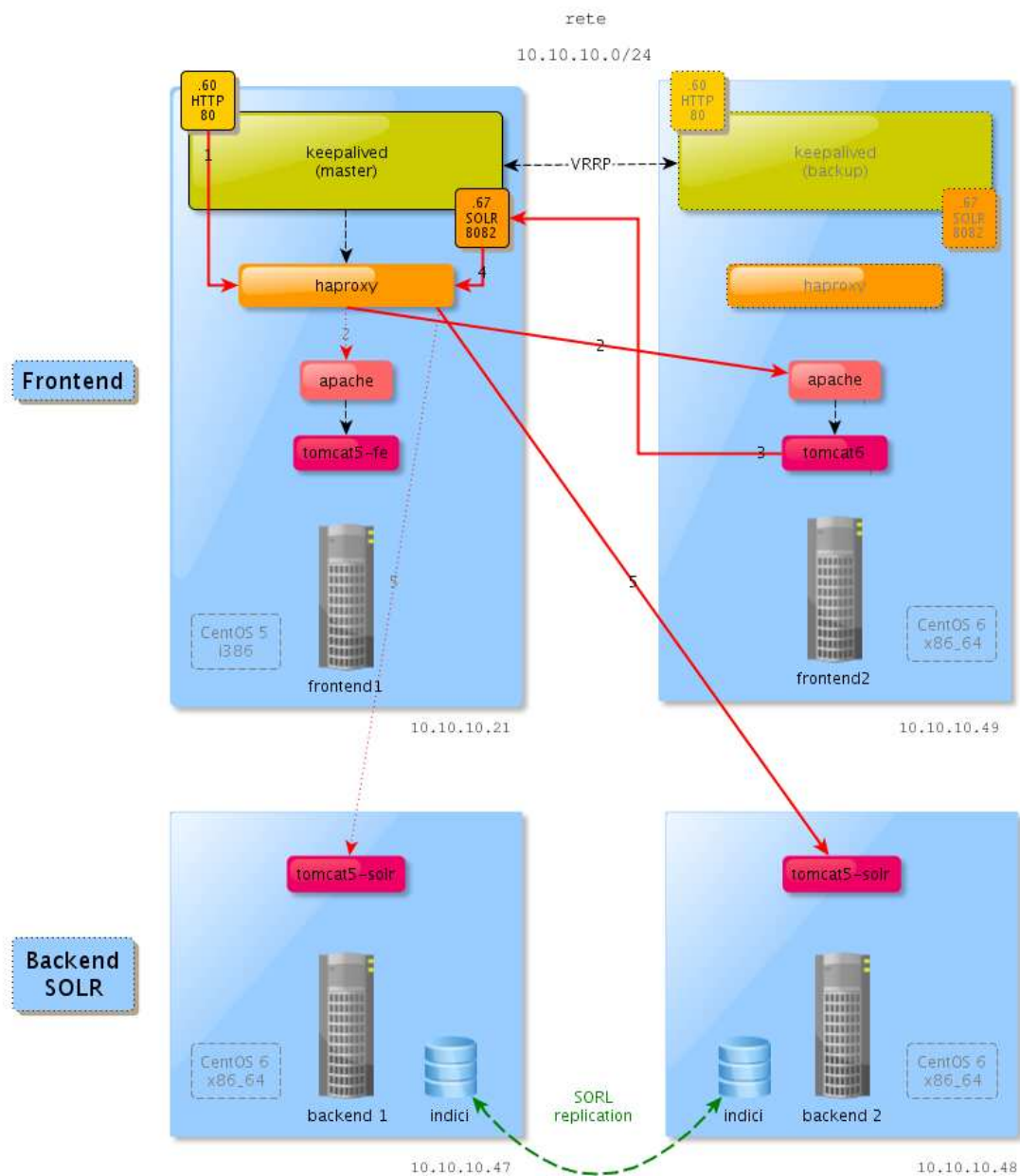


Figure 1.3: Percorso alternativo chiamate HTTP a OPAC

1.5 Architettura Backend

Il backend è composto dai due nodi **backend1** e **backend2**. Entrambi i nodi hanno spazio disco, capacità elaborativa e memoria a sufficienza per sopportare singolarmente il carico (anche di picco) degli utenti dell'OPAC.

Ciascun nodo ospita un tomcat (**tomcat5-solr**) sul quale è installata la webapp **Solr**.

Le figure qui di seguito riportano alcune dei possibili percorsi delle richieste HTTP⁶:

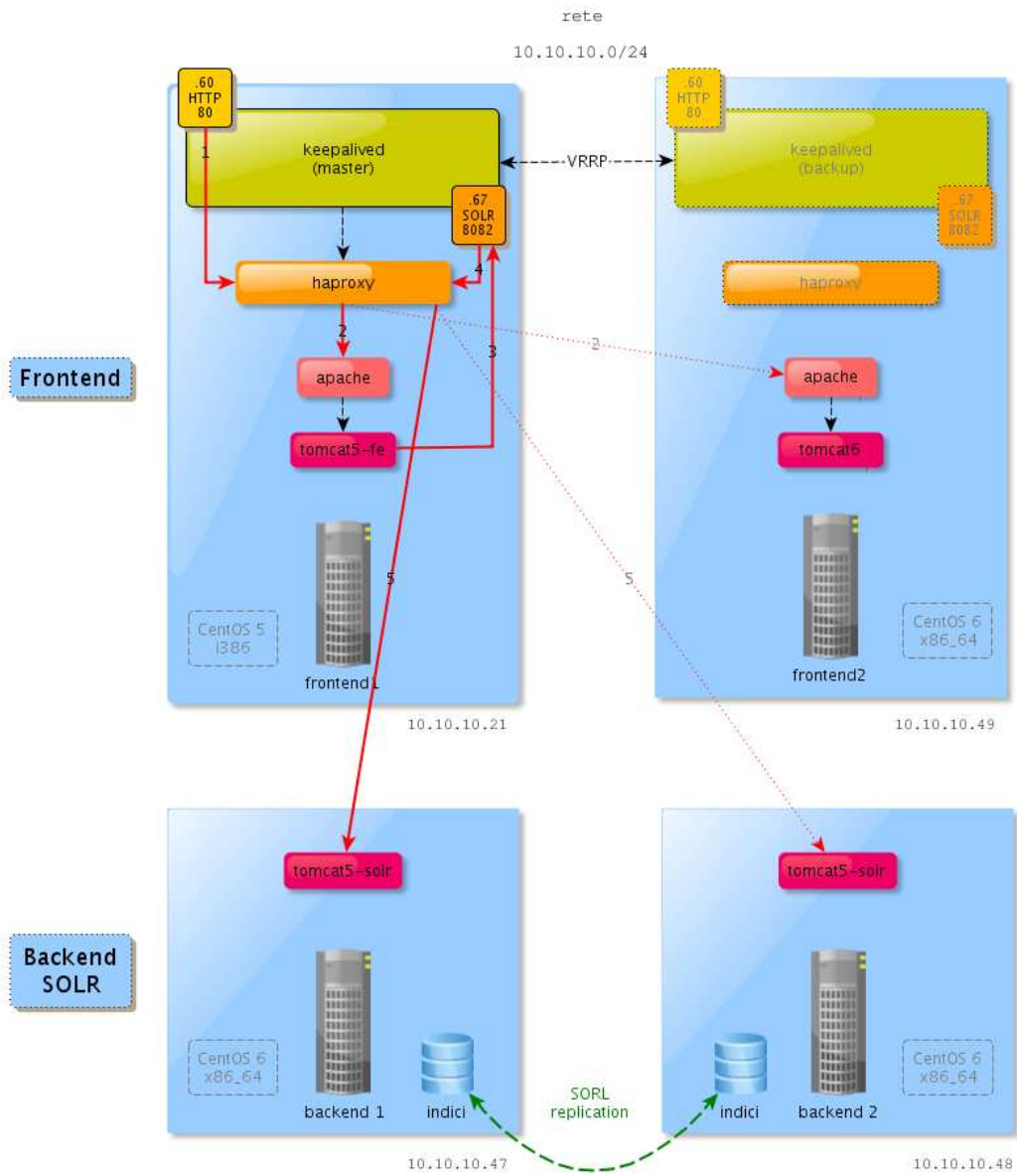


Figure 1.4: Richiesta HTTP - percorso 1

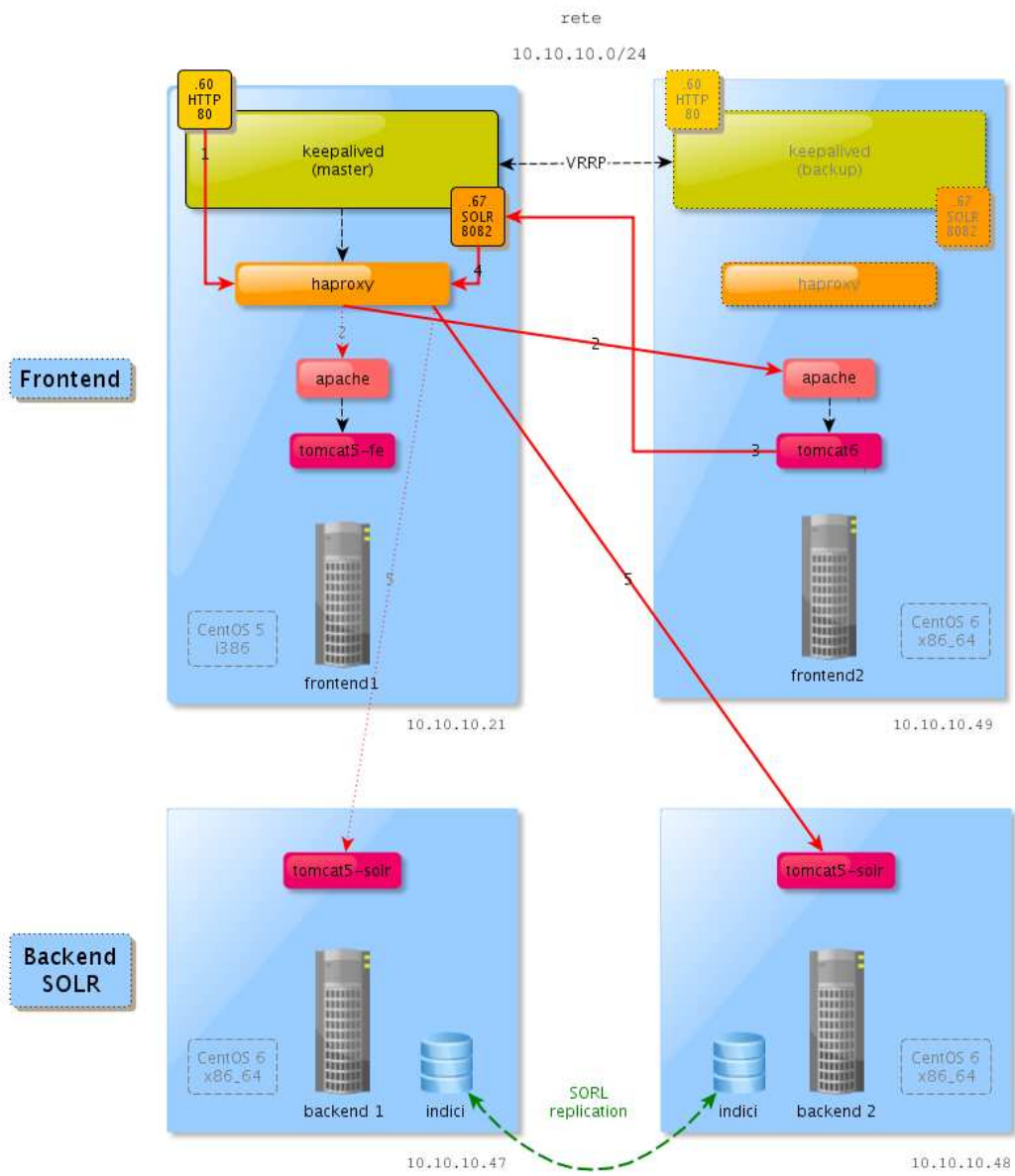


Figure 1.5: Richiesta HTTP - percorso 2

1.5.1 Requisiti risorse backend

Lo spazio disco occupato attualmente dagli indici è:

Numero record	11.946.730
Occupazione spazio disco indici	32 GB ⁴

Le risorse richieste dal backend sono:

Memoria JVM tomcat Solr	24 GB ⁴
Spazio disco per gestione indici Solr ⁸	80 GB ⁴

1.5.2 Architettura Solr

I nodi hanno indici Solr in *replication*. La *replication* è in configurazione **master-slave**, cioè le modifiche effettuate sul master vengono replicate sullo slave. Nonostante la *replication* possa essere effettuata automaticamente, essa è configurata ma non è attiva alla partenza della webapp, la sua attivazione è scatenata dalle procedure di aggiornamento.

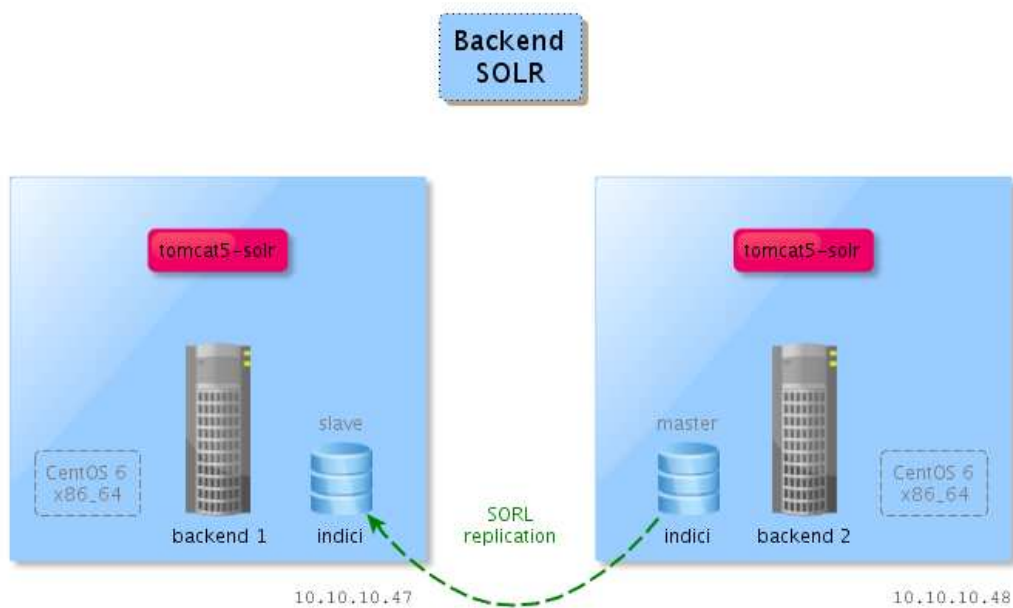


Figure 1.6: Configurazione Solr

⁸Lo spazio disco qui riportato è lo stretto necessario per avere un margine di crescita degli indici (+25%) e poter gestire l'ottimizzazione dei medesimi, operazione che effettua una copia dell'indice e che quindi raddoppia, per il tempo necessario all'operazione, lo spazio disco impiegato

1.5.3 Aggiornamento

La motivazione di questa scelta è dettata dalla necessità di consistenza dei risultati. Si aggiorna quindi il nodo *backup* soltanto al completamento dell'aggiornamento del nodo *master*. Così facendo gli utenti OPAC non riscontreranno discontinuità nell'erogazione del servizio dovute a temporanee cancellazioni o modifiche di record e troveranno sempre dati consistenti.

Richieste Solr

Nella figura qui di seguito vengono riportate come vengono bilanciate le richieste Solr nelle condizioni di normale operatività:

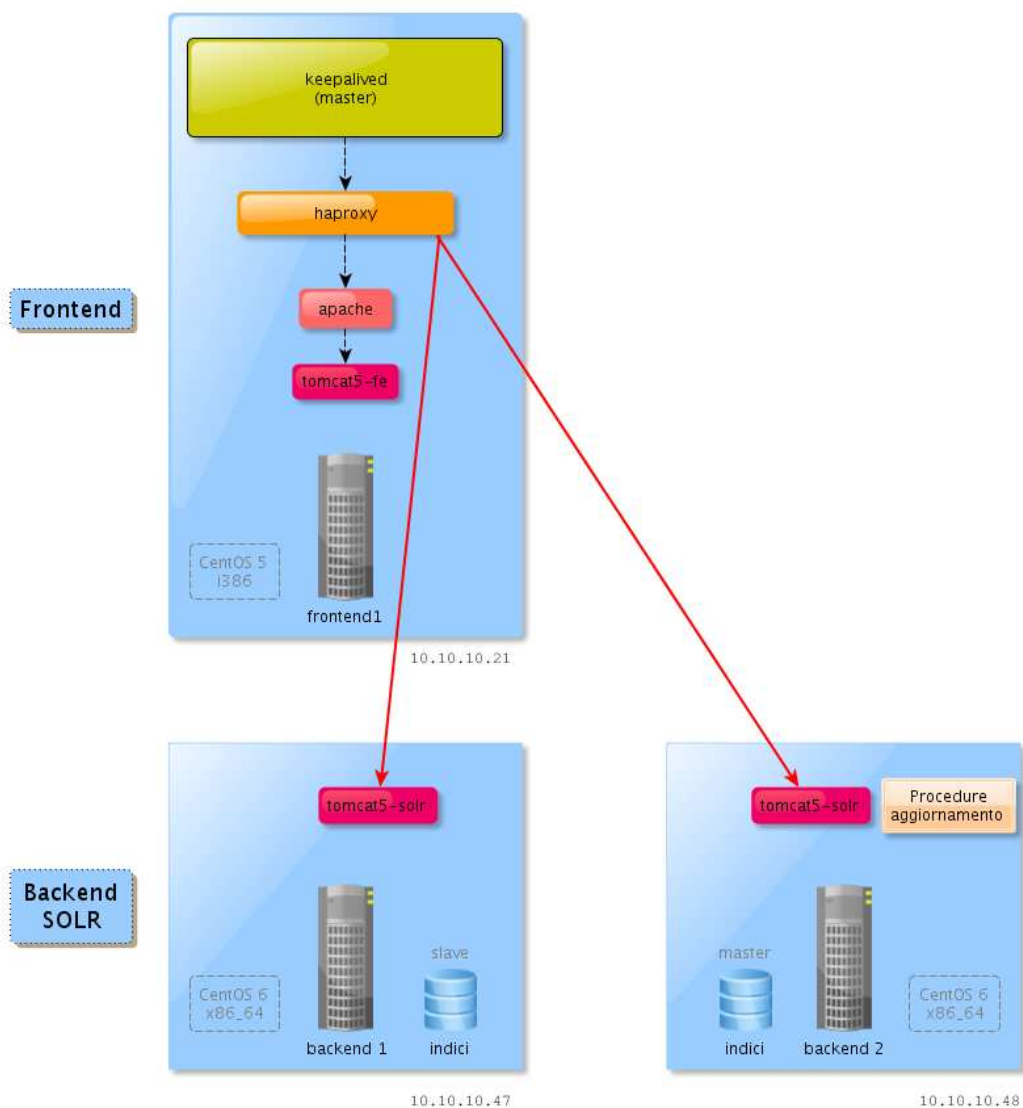


Figure 1.7: Solr - bilanciamento richieste

Fase 1: aggiornamento master

Nella figura qui di seguito viene evidenziato come le procedure di aggiornamento configurino temporaneamente **haproxy** per indirizzare sul nodo *slave* le richieste Solr durante l'aggiornamento degli indici del nodo *master*:

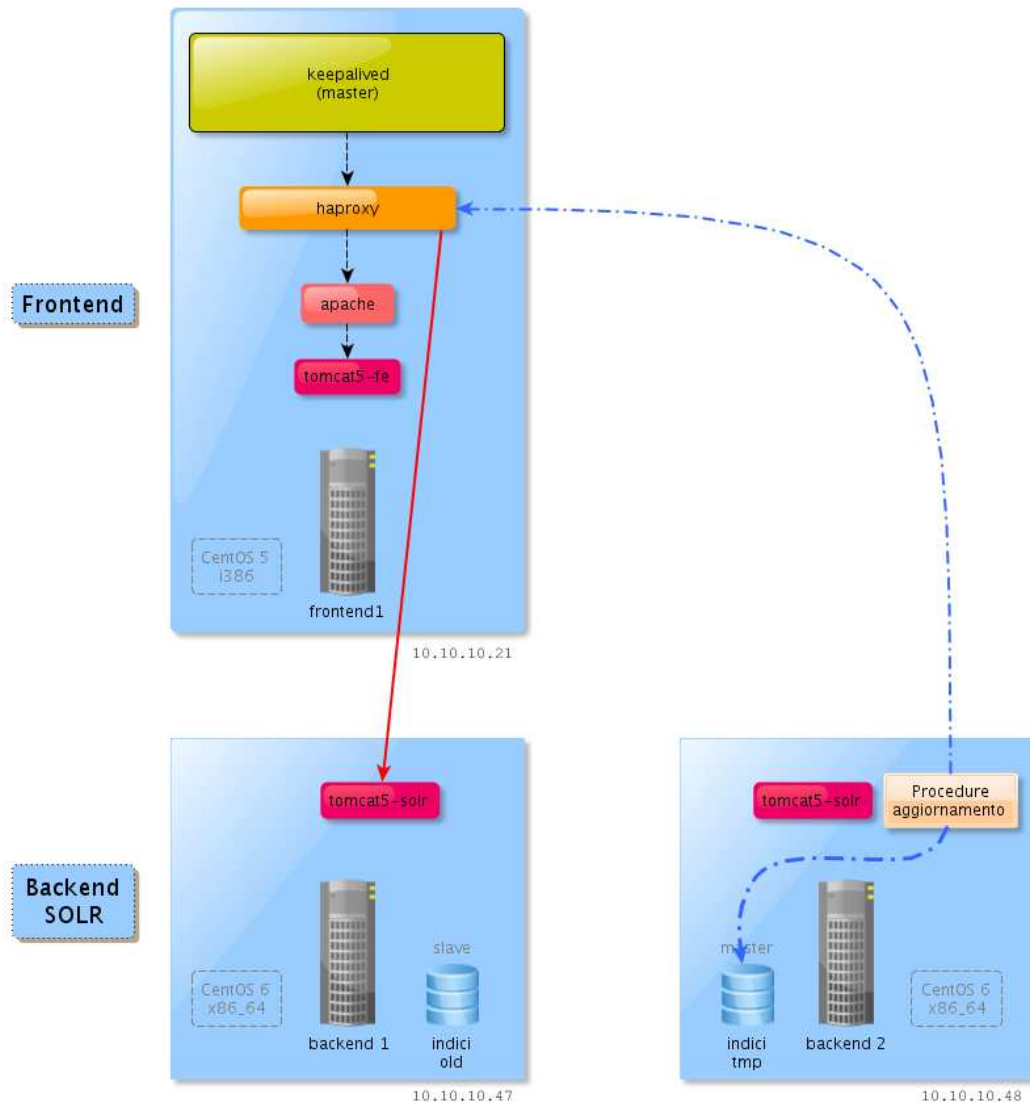


Figure 1.8: Aggiornamento Solr: fase 1 - aggiornamento master

Fase 2: aggiornamento slave

Nella figura qui di seguito viene evidenziato come le procedure di aggiornamento configurino temporaneamente **haproxy** per indirizzare sul nodo *master* le richieste Solr durante l'aggiornamento in replica degli indici del nodo *slave*:

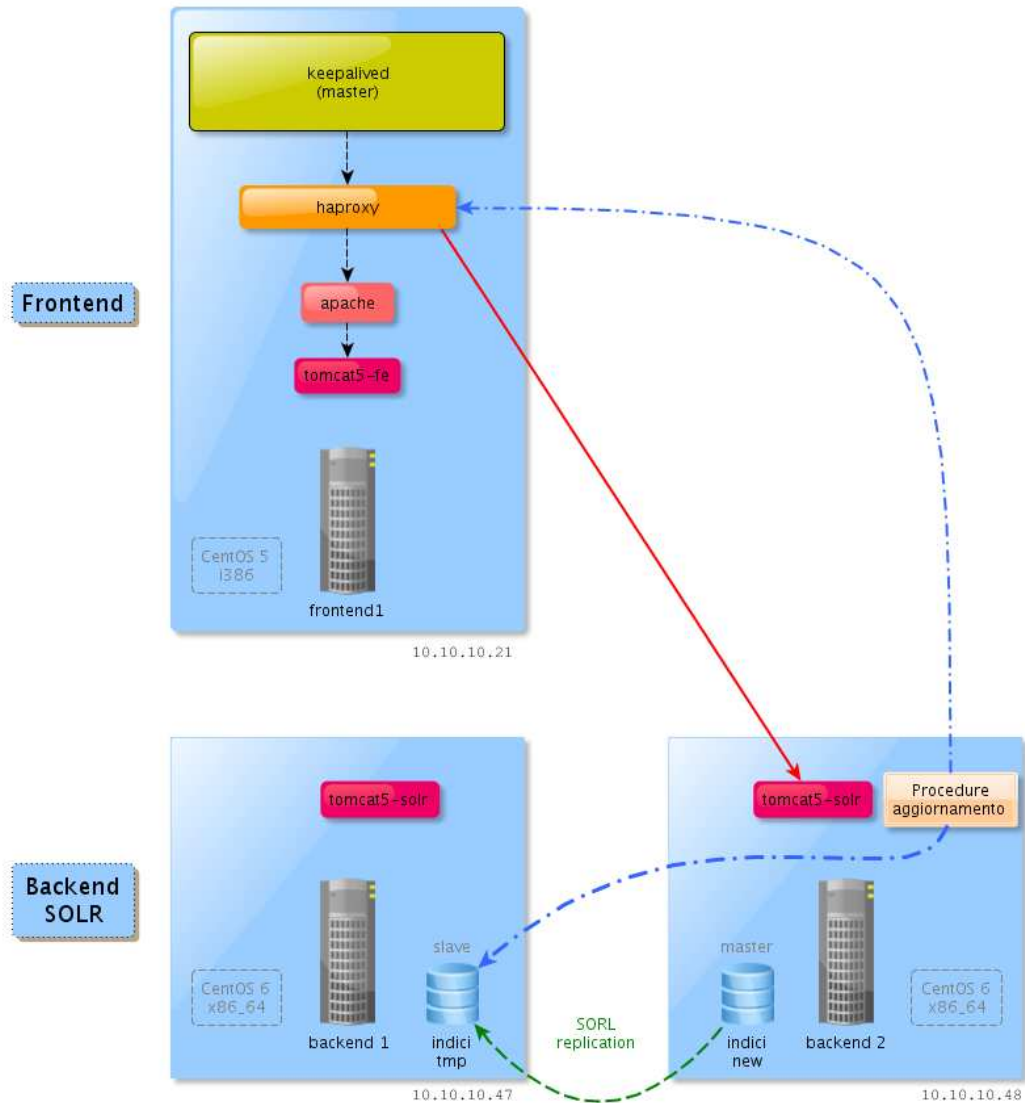


Figure 1.9: Aggiornamento Solr: fase 2 - aggiornamento slave

Fase 3: ripristino normale operatività

Nella figura qui di seguito viene evidenziato come le procedure di aggiornamento configurino **haproxy**, a seguito dell'avvenuto aggiornamento e replica, per bilanciare su entrambi i nodi, *master* e *slave*, le interrogazioni agli indici Solr:

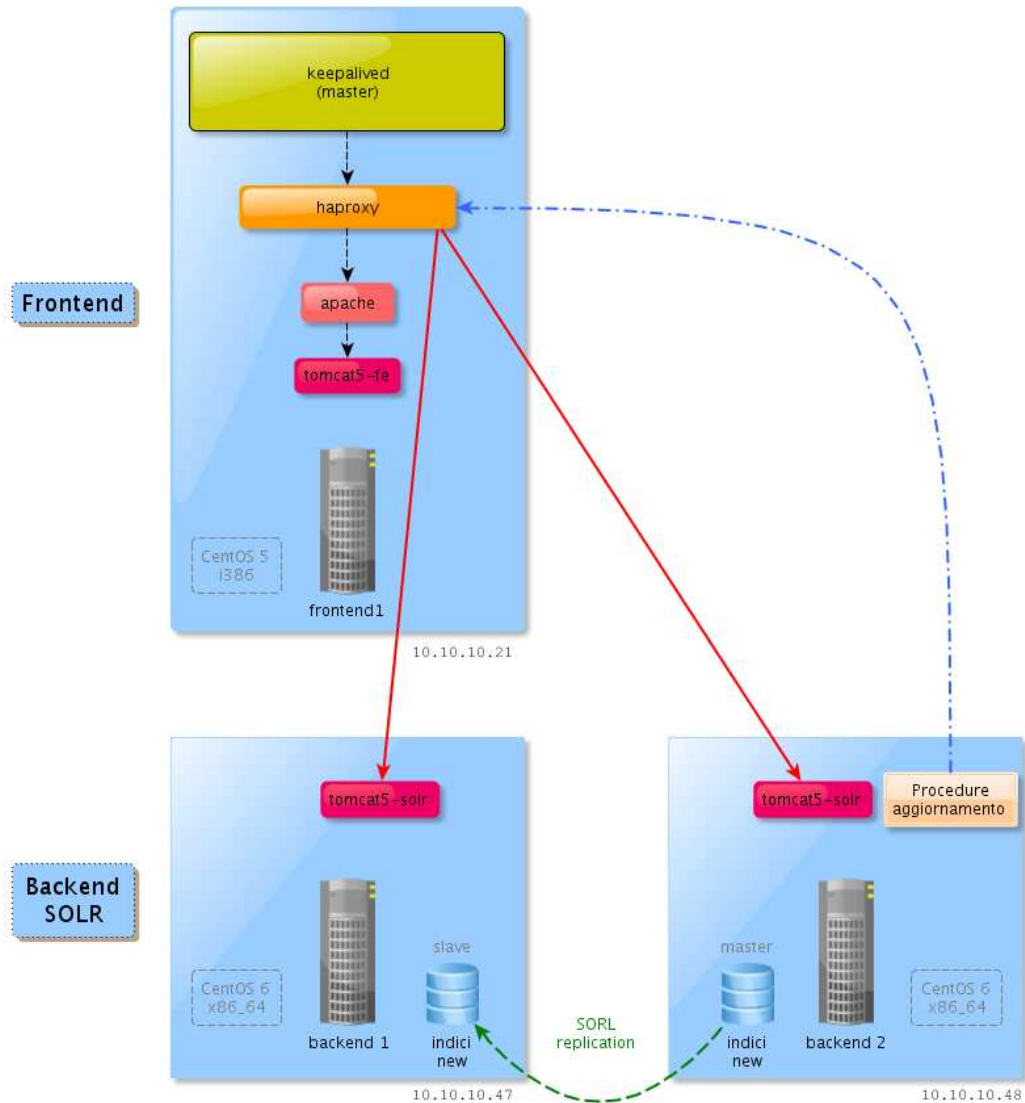


Figure 1.10: Aggiornamento Solr: fase 3 - ripristino normale operatività

Capitolo 2

Configurazione

L'OPAC è configurabile in tutte le sue componenti:

- Solr: è il motore di indicizzazione e ricerca, si possono configurare lo schema dei dati e la configurazione vera e propria del servizio;
- SolrMarc: componente che viene utilizzato durante le procedure di aggiornamento e che carica i dati in formato unimarc su Solr
- JSP interfaccia WEB (OPAC)
- Target Z39.50

2.1 Riepilogo path applicazioni

I path nei quali sono installate le applicazioni sono:

sw	nodo	path
Solr	backend1	/var/lib/tomcat5-solr/webapps/solr/
Solr	backend2	/var/lib/tomcat5-solr/webapps/solr/
OPAC	frontend1	/var/lib/tomcat5-fe/webapps/opacsbn/
OPAC	frontend2	/var/lib/tomcat6/webapps/opacsbn/
target Z39.50	frontend2	/var/lib/jztarget/
SolrMarc	backend1	/var/icc/aggiornamento/proceduresolr/solrmarc
SolrMarc	backend2	/var/icc/aggiornamento/proceduresolr/solrmarc

2.2 Riepilogo file di configurazione

I file di configurazione sono riportati qui di seguito e sono relativi al percorso di installazione dell'applicazione come riportato nella tabella di cui sopra:

sw	configurazione	file
Solr	schema core biblio	WEB-INF/solr/biblio/conf/schema.xml
Solr	configurazione core biblio	WEB-INF/solr/biblio/conf/solrconfig.xml
Solr	stopwords core biblio	WEB-INF/solr/biblio/conf/stopwords.txt
Solr	schema core authority	WEB-INF/solr/authority/conf/schema.xml
Solr	configurazione core authority	WEB-INF/solr/authority/conf/solrconfig.xml
Solr	stopwords core authority	WEB-INF/solr/authority/conf/stopwords.txt
Solr	schema core autocomplete	WEB-INF/solr/autocomplete/conf/schema.xml
Solr	configurazione core autocomplete	WEB-INF/solr/autocomplete/conf/solrconfig.xml
OPAC	configurazione OPAC	WEB-INF/config/opaclib_iccu.xml
target Z39.50	configurazione target	etc/JZKitConfig.xml
SolrMarc	indicizzazione da unimarc	dist/opacsbn_index.properties

2.3 Configurazione di Solr

In questo file si specificano i parametri di configurazione di Solr. Rispetto a quello messo a disposizione da Solr, sono stati aumentati i parametri di default della memoria per gestire un indice molto grande e la directory dove mettere fisicamente gli indici.

Il file è:

```
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/biblio/conf/solrconfig.xml
```

2.4 Schema Solr

Lo schema dei dati in Solr è un file di configurazione XML¹ che definisce come sono organizzati l'indice, i tipi di dato, gli analizzatori e i filtri utilizzati sia in fase di indicizzazione che in fase di query e diversi altri parametri di configurazione.

Schematizzando è possibile individuare le seguenti aree di configurazione:

- definizione delle tipologie di dato e analizzatori di testo
- definizione dello schema dei dati
- parametri di configurazione

Il file è:

```
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/biblio/conf/schema.xml
```

2.5 SolrMarc

SolrMarc è un'applicazione open-source altamente configurabile ed estendibile che permette l'acquisizione dei dati in formato Unimarc negli indici Solr.

Questa applicazione è stata estesa con opportune procedure per creare tutti gli indici secondo quanto richiesto dal committente.

Il file di configurazione utilizzato è :

¹Extensible Markup Language

```
/var/iccu/aggiornamento/proceduresolr/solrmacr/dist/opacsbn_index.properties
```

In questo file si definisce il mapping tra indice Solr e dati Unimarc, utilizzando anche le procedure scritte ad-hoc per l'indicizzazione.

Si rimanda a

<http://code.google.com/p/solrmarc/wiki/IndexProperties>
per la completa descrizione della sintassi di questo file.

2.6 Configurazione applicativa

Il file di configurazione dell'applicazione di frontend (*opacsbn*) è un file xml che risiede nelle macchine di frontend:

frontend1

```
/var/lib/tomcat5-fe/webapps/opacsbn/WEB-INF/config/opaclub_iccu.xml
```

frontend2

```
/var/lib/tomcat6/webapps/opacsbn/WEB-INF/config/opaclub_iccu.xml
```

In tale file sono configurabili:

- mailserver da interrogare (sezione mailserver)
- la configurazione del Solr da interrogare (sezione solr)
- parametri di mapping tra Solr e i canali di ricerca usati nelle form (sezione attrset)
- parametri di default per le connessioni Z39.50 e/o dell'applicativo: da non modificare
- banche dati da interrogare (sezione databases)

2.7 Configurazione Z39.50

Il file di configurazione del target z39.50 è un file xml che risiede nella macchina di frontend2 dove viene descritto il mapping tra punti di accesso e altri parametri Z39.50 e gli indici di Solr.

Il file è:

```
/var/lib/jztarget/etc/JZKitConfig.xml
```

In tale file inoltre si possono configurare le trasformate da utilizzare per convertire il record Unimarc in altre record syntax. Attualmente, le record syntax supportate sono **Unimarc** e **SUTRS**.

2.7.1 Configurazione record syntax SUTRS

La trasformata per la record syntax SUTRS utilizzata è

```
/var/lib/jztarget/etc/config/base/crosswalks/RecordModel/sutrs.xsl
```

e il suo utilizzo è definito come segue nel file di configurazione *JZKitConfig.xml*:

```
<mapping fromSpec="solr" toSpec="sutrs" type="DOM-XSL-TXT" resource="/config/base/crosswalks/RecordMode
```

Capitolo 3

Procedure

Descriviamo le procedure che sono state realizzate per la manutenzione dell'OPAC.

Queste intervengono su:

- aggiornamento dell'OPAC da scarichi incrementali da Indice
- aggiornamento dell'OPAC da scarichi integrali da Indice

3.1 Aggiornamento incrementale

- 00.preparazione
- 01.scarico-e-conversione
- 02.disabilitazione-backend2
- 03.cancellazioni
- 04.inserimento-localizzazioni
- 05.inserimento-documenti
- 06.ottimizzazione
- 07.abilitazione-backend2
- 08.attivazione-replica
- 09.abilitazione-backend1

3.1.1 00.preparazione

Questa procedura provvede a riavviare il Solr della macchina *slave* perché sarà l'unico che verrà interrogato durante le procedure di aggiornamento; per questo motivo si effettua il riavvio per evitare che si manifestino condizioni di blocco dovute all'eccessivo utilizzo o a problemi contingenti di occupazione anomala della memoria.

3.1.2 01.scarico-e-conversione

Questa procedura provvede a scaricare gli aggiornamenti incrementali dal repository di scambio file. Il repository di scambio è accessibile con i seguenti parametri:

IP esterno	193.206.221.17
IP interno	10.10.10.11
username	export
password	export2010
directory	opacPeriodico

3.1.3 02.disabilitazione-backend2

Questa procedura interviene sulla configurazione di **haproxy** di entrambi i nodi di frontend (**frontend1** e **frontend2**) e disabilita l'interrogazione del Solr del nodo **backend2**.

3.1.4 03.cancellazioni

Questa procedura ricava dal file

```
AAAA-MM-GG_TITOLI_CANCELLATI
```

la lista dei record da cancellare dagli indici Solr e provvede ad effettuare la relativa cancellazione.

3.1.5 04.inserimento-localizzazioni

Questa procedura provvede a cancellare tutte le localizzazioni di un record e a inserire le nuove localizzazioni aggiornate.

3.1.6 05.inserimento-documenti

Questa procedura utilizza **SolrMarc** per indicizzare i record partendo dal file unimarc di aggiornamento opportunamente pre-elaborato dalla procedura 01.scarico-e-conversione

3.1.7 06.ottimizzazione

Questa procedura riavvia il tomcat Solr (**tomcat5-solr**) della macchina **backend2**, esegue la *query di autowarm* (script `/var/iccu/autowarm_query.sh`) e invia il comando di ottimizzazione a Solr per ricompattare gli indici. Alla fine della ricompattazione degli indici il tomcat viene riavviato e viene eseguita la *query di autowarm*.

3.1.8 07.abilitazione-backend2

Questa procedura interviene sulla configurazione di haproxy di entrambi i nodi di frontend (**frontend1** e **frontend2**) disabilitando l'interrogazione del Solr del nodo **backend1** e abilitando l'interrogazione del Solr del nodo **backend2**.

3.1.9 08.attivazione-replica

Questa procedura provvede ad attivare la replica degli indici del nodo **backend1** dal nodo **backend2**. Alla fine della replica il tomcat Solr del nodo **backend1** viene riavviato e viene eseguita la *query di autowarm*.

3.1.10 09.abilitazione-backend1

Questa procedura interviene sulla configurazione di **haproxy** di entrambi i nodi di frontend (**frontend1** e **frontend2**) abilitando l'interrogazione del Solr di entrambi i nodi, **backend1** e **backend2**.

Capitolo 4

Attività di manutenzione

Si riportano qui di seguito alcune delle operazioni che servono durante le regolari attività di manutenzione.

4.1 Avvio e terminazione OPAC

4.1.1 Avvio

frontend1

```
service tomcat5-fe start
service httpd      start
```

frontend2

```
service tomcat6 start
service httpd   start
```

4.1.2 Terminazione

frontend1

```
service httpd      stop
service tomcat5-fe stop
```

frontend2

```
service httpd   stop
service tomcat6 stop
```

4.2 Esclusione nodo frontend da interrogazione

Per escludere un nodo di frontend dall'interrogazione si deve terminare l'**apache** relativo. Infatti **haproxy** smette di inviare richieste HTTP¹ all'**apache** che gli risulta non più contattabile o non più responsivo. Per effettuare interventi sull'applicativo di frontend si dovrà disabilitare l'**apache** corrispondente.

```
service httpd      stop
```

4.3 Avvio e terminazione Solr

L'avvio del Solr richiede l'accorgimento di eseguire anche la *query di autowarm* prima di rimetterlo in linea.

4.3.1 Avvio solr

```
service tomcat5-solr start
/var/iccu/autowarm_query.sh
```

4.3.2 Terminazione solr

```
service tomcat5-solr stop
```

4.4 Log attività

Saranno approntati dei meccanismi di log delle attività effettuate dall'utente in fase di catalogazione in modo tale da poter stabilire quale operazioni sono state effettuate.

4.5 Backup

Per il corretto funzionamento e salvaguardia dei dati dell'applicativo sono necessarie delle procedure di backup.

Le procedure possono essere configurate per essere eseguite ad intervalli regolari, ad esempio giornalieri, notturni, ecc...

In particolare vanno approntate le seguenti procedure:

- **backup applicazioni software**
va eseguito il backup del software applicativo in modo da poter ripristinare ogni singola componente a seguito ad esempio di un danno al disco rigido.
- **backup indici Solr**
va eseguito il backup delle directory del filesystem relative all'indice di Solr

¹Hyper Text Transfer Protocol

4.5.1 backup applicazioni software

I percorsi da memorizzare sono:

backup frontend1

```
# OPAC
/etc/sysconfig/tomcat5-fe
/var/lib/tomcat5-fe/webapps/opacsbn

# Old OPAC
/etc/sysconfig/tomcat5-03
/var/lib/tomcat5-03/webapps/opacsbn

# keepalived
/etc/keepalived

# haproxy
/etc/haproxy

# z39.50
/etc/z3950

# utilities
/usr/local/bin
```

backup frontend2

```
# OPAC
/etc/sysconfig/tomcat6
/var/lib/tomcat6/webapps/opacsbn

# keepalived
/etc/keepalived

# haproxy
/etc/haproxy

# z39.50
/var/lib/jztarget

# utilities
/usr/local/bin
```

backup backend1

```
# Solr
/etc/sysconfig/tomcat5-solr
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/solr.xml
```

```
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/biblio/conf
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/authority/conf
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/autocomplete/conf
```

```
# z39.50
/var/lib/jztarget
```

```
# utilities
/usr/local/bin
```

4.5.2 backup indici Solr

I percorsi da memorizzare su entrambe le macchine, **backend1** e **backend2**, sono:

```
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/biblio/data
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/authority/data
/var/lib/tomcat5-solr/webapps/solr/WEB-INF/solr/autocomplete/data
```

Basterebbe memorizzare soltanto un nodo visto che vengono sincronizzati l'uno con l'altro ma visto che ci possono essere frangenti di temporanea inconsistenza, questa sicuramente non si riscontra su entrambi i nodi nello stesso periodo, quindi si consiglia il doppio backup.

4.6 Job

Le seguenti procedure sono organizzate per essere eseguite in tempi differenti, ad intervalli regolari o in momenti particolari (ad es. notturna).

In particolare sono state approntate le seguenti procedure:

- procedura di generazione delle statistiche HTTP¹ (**awstats**) integrata con quanto già presente;
- procedura per il conteggio dei record e visualizzazione della data di aggiornamento

Capitolo 5

Riferimenti

- CentOS <http://www.centos.org/>
- Apache Solr <http://lucene.apache.org/solr/>
- Apache Tomcat <http://tomcat.apache.org/>
- HaProxy <http://haproxy.1wt.eu/>
- keepalived <http://keepalived.com/>
- SolrMarc <http://code.google.com/p/solrmarc/>
- JzProxy <http://developer.k-int.com/downloads/jzproxy/>