

Toward a stand alone distributed management system

Enrico Vianello, Matteo Manzali, Bruno Santeramo, Luca Tomassetti

Generalization of SuperB distributed production system as stand alone, general purpose infrastructure to accomplish small and medium VO requirements

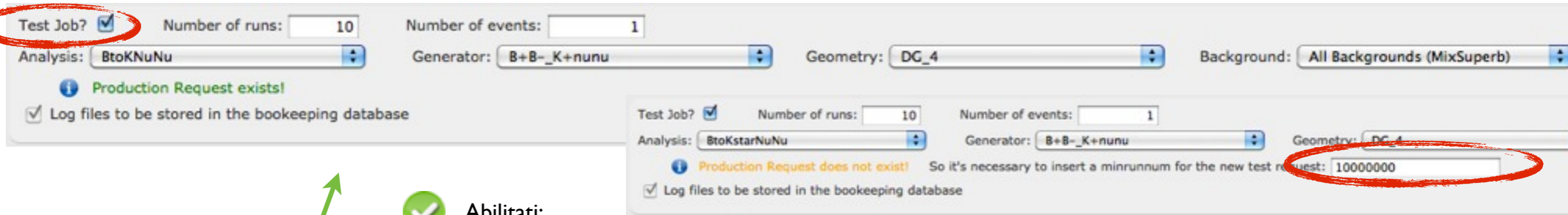
oggi parliamo di:

- SVILUPPO WebUI

 - EXPERT INIT

 - stato del lavoro: usabilità, minrunnum e calcolo del range di runnum di un bulk

- TIMELINE SVILUPPO WebUI



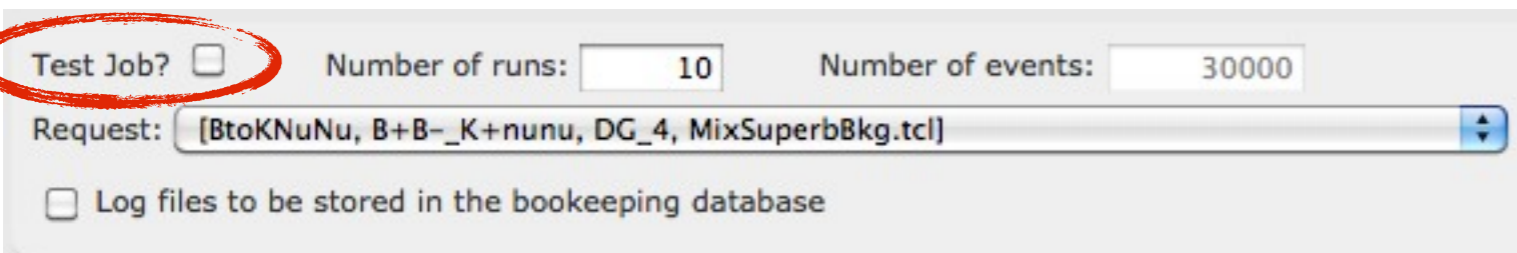
La produzione viene selezionata tra quelle non chiuse e non di test



- ✔ Abilitati:
 - number of runs
 - number of events (il numero di eventi di un job di test non dipende dal campo events_per_job della corrispondente request - stessa request può avere job con event_num diversi)
 - custom request : se la request selezionata usando i selettori dei singoli parametri non esiste viene creata nuova *
- ✘ Non abilitati:
 - log : viene impostato a TRUE di default se si tratta di un job di test

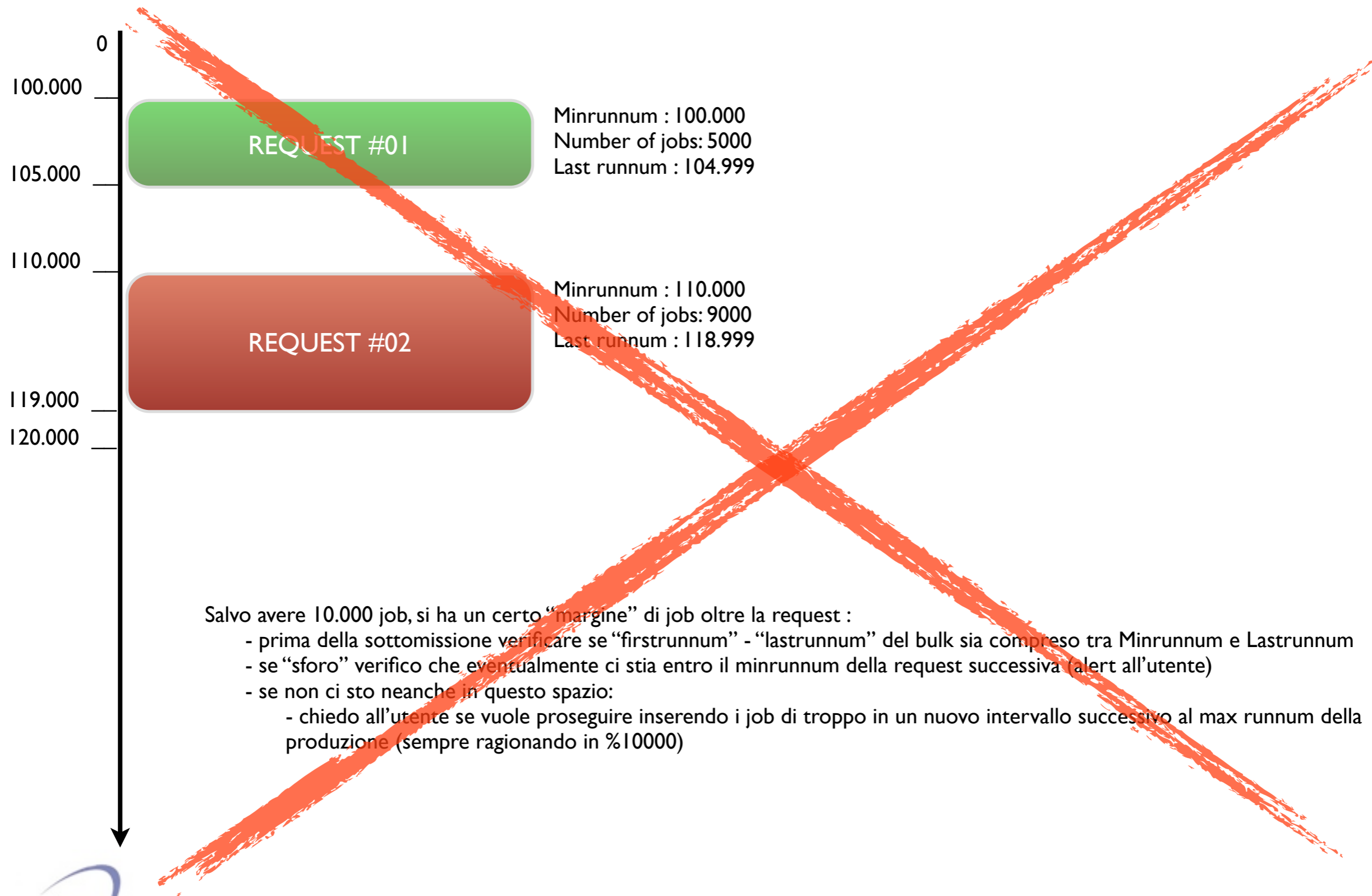
il job verrà lanciato per la produzione di test associata

- ✔ Abilitati:
 - number of runs
 - log
- ✘ Non abilitati:
 - number of events (il numero di eventi di un job di test dipende dal campo events_per_job della sua request)
 - custom request : impossibilità di creare nuove request al volo ma va selezionata una di quelle esistenti (!open)



* con valori di default :
 'req_events' => 1,
 'events_per_job' => 1,
 'number_of_jobs' => 1,
 'priority' => 1,
 'exp_event_time' => 1,
 'max_wct' => 12000,
 'progress' => "NONE"

Come calcolo il minrunnum di una request : AUTOMATICO



Come calcolo il **minrunnum** di una request :

SOLUZIONE LIBERA (a discrezione dell'utente)

PERCHE' ?

Un utente consapevole "distanza" le request in modo appropriato

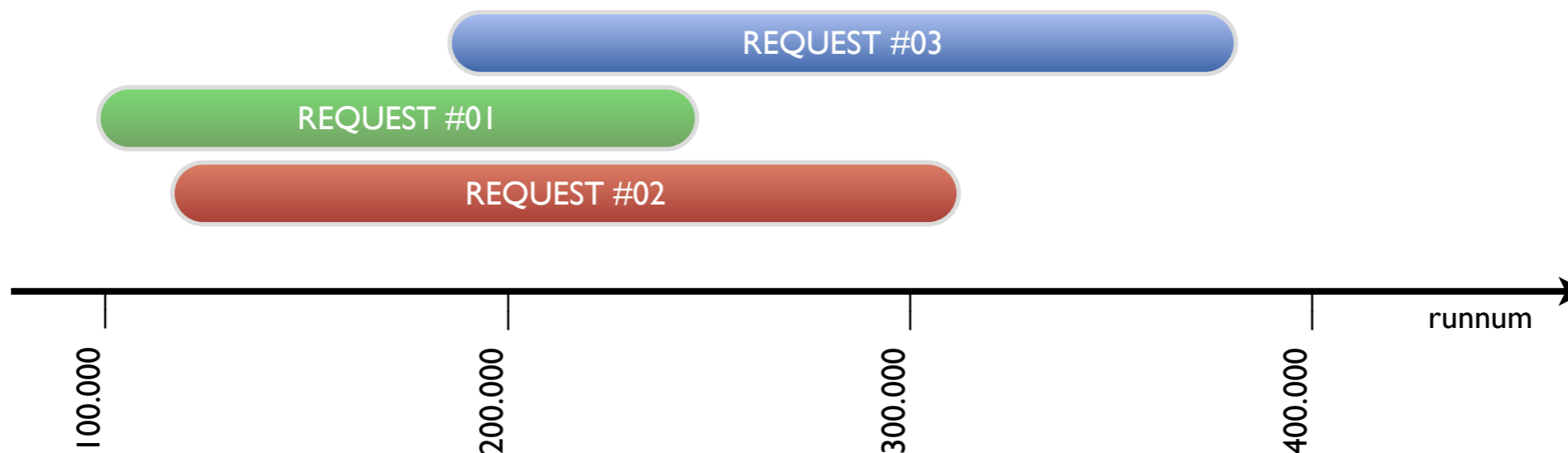
Può essere necessario che 2 request condividano lo stesso minrunnum (quando hanno la stessa ANALISI)

Job di request diverse "mescolati" (in sbk3 non era un problema)

Calcolo del range runnum per la sottomissione di un bulk di job per una request non agevole

Il runnum successivo all'ultimo job per quella request potrebbe non essere libero

La prima posizione libera?



1 ESEGUO LA QUERY:

```
SELECT P.minrunnum, MAX(J.runnum) AS lastrunnum
FROM Fast_Prod_Request AS P, Fast_Job AS J
WHERE P.prod_series = '2010_September_311'
      AND P.prod_series = J.prod_series
      AND P.request_name = J.request_name
GROUP BY P.prod_series, P.request_name
ORDER BY P.minrunnum, lastrunnum
```



- 1) 1007016815 - 1007035924
- 2) 1007016825 - 1007038124
- 3) 1007017150 - 1007037924
- 4) 1007017950 - 1007038094
- 5) 1007018050 - 1007038054
- 6) 1007018450 - 1007038034
- 7) 1007024080 - 1007037984
- 8) 1007035925 - 1007037874
- 9) 1007110215 - 1007120159
- 10) 1007110415 - 1007128154
- 11) 1007110915 - 1007119409
- 12) 1007111015 - 1007119159
- 13) 1007111420 - 1007128204
- 14) 1007111620 - 1007128304
- 15) 1007112770 - 1007127969
- 16) 1007125190 - 1007127929
- 17) 1007205000 - 1007213599
- 18) 1007205700 - 1007213639
- 19) 1007207000 - 1007213249
- 20) 1007207100 - 1007213499
- 21) 1007402116 - 1007403130
- 22) 1007402216 - 1007403135
- 23) 1007402316 - 1007403125
- 24) 1007402416 - 1007402515
- 25) 1007402516 - 1007403145
- 26) 1007402616 - 1007403120
- 27) 1007402716 - 1007402815
- 28) 1007402816 - 1007403150
- 29) 1007402916 - 1007403015
- 30) 1007403016 - 1007403115
- 31) 1007403136 - 1007403140
- 32) 1007403151 - 1007403155
- 33) 1007500827 - 1007500926
- 34) 1007500927 - 1007501251
- 35) 1007501027 - 1007501126
- 36) 1007501127 - 1007501131
- 37) 1007501132 - 1007501246
- 38) 1007501252 - 1007501256
- 39) 1007600911 - 1007601365
- 40) 1007600961 - 1007601065
- 41) 1007601066 - 1007601170
- 42) 1007601221 - 1007601375
- 43) 1007601231 - 1007601340

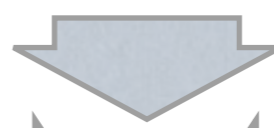
2

minimize()
(molti range sono sovrapposti = li unisco)



- 1) 1007016815 - 1007038124
- 2) 1007110215 - 1007128304
- 3) 1007205000 - 1007213639
- 4) 1007402116 - 1007403155
- 5) 1007500827 - 1007501256
- 6) 1007600911 - 1007601375

minrunnum
della request
del bulk



3

numjobs
del bulk

nextFirst()

trovo il runnum
minimo del bulk
di njobs



1 ESEGUO LA QUERY:

```
SELECT P.minrunnum, MAX(J.runnum) AS lastrunnum
FROM Fast_Prod_Request AS P, Fast_Job AS J
WHERE P.prod_series = '2010_September_311'
AND P.prod_series = J.prod_series
AND P.request_name = J.request_name
GROUP BY P.prod_series, P.request_name
ORDER BY P.minrunnum, lastrunnum
```



- 1) 1007016815 - 1007035924
- 2) 1007016825 - 1007038124
- 3) 1007017150 - 1007037924
- 4) 1007017950 - 1007038094
- 5) 1007018050 - 1007038054
- 6) 1007018450 - 1007038034
- 7) 1007024080 - 1007037984
- 8) 1007035925 - 1007037874
- 9) 1007110215 - 1007120159
- 10) 1007110415 - 1007128154
- 11) 1007110915 - 1007119409
- 12) 1007111015 - 1007119159
- 13) 1007111420 - 1007128204
- 14) 1007111620 - 1007128304
- 15) 1007112770 - 1007127969
- 16) 1007125190 - 1007127929
- 17) 1007205000 - 1007213599
- 18) 1007205700 - 1007213639
- 19) 1007207000 - 1007213249
- 20) 1007207100 - 1007213499
- 21) 1007402116 - 1007403130
- 22) 1007402216 - 1007403135
- 23) 1007402316 - 1007403125
- 24) 1007402416 - 1007402515
- 25) 1007402516 - 1007403145
- 26) 1007402616 - 1007403120
- 27) 1007402716 - 1007402815
- 28) 1007402816 - 1007403150
- 29) 1007402916 - 1007403015
- 30) 1007403016 - 1007403115
- 31) 1007403136 - 1007403140
- 32) 1007403151 - 1007403155

2

minimize()
(molti range sono sovrapposti = li unisco)



- 1) 1007016815 - 1007038124
- 2) 1007110215 - 1007128304
- 3) 1007205000 - 1007213639
- 4) 1007402116 - 1007403155
- 5) 1007500827 - 1007501256**
- 6) 1007600911 - 1007601375



minrunnum
della request
del bulk
1007500827

3

numjobs
del bulk
10

nextFirst()

JOBS DATA - SUMMARY						
Set	Number of Jobs	Site	Type	First RUNNUM	Last RUNNUM	Number of Events (ea.)
1	10	INFN-MILANO	grid	1007501257	1007501266	30000
2	10	INFN-TORINO	grid	1007501267	1007501276	30000
3	10	INFN-MILANO	grid	1007501277	1007501286	30000
4	10	INFN-TORINO	grid	1007501287	1007501296	30000

33) **1007500827 - 1007500926**

- 34) 1007500927 - 1007501251
- 35) 1007501027 - 1007501126
- 36) 1007501127 - 1007501131
- 37) 1007501132 - 1007501246
- 38) 1007501252 - 1007501256
- 39) 1007600911 - 1007601365
- 40) 1007600961 - 1007601065
- 41) 1007601066 - 1007601170
- 42) 1007601221 - 1007601375
- 43) 1007601231 - 1007601340

DUE STRADE:



SUBMIT CON SCRIPT BASH

- meno orientato al generico
- complessità maggiore
- + affidabile

? capire quanto ne vale la pena in termini di tempo

SUBMIT CON FILE CFG PER SCRIPT *Severus*

- + già modularizzato (si adatta meglio a un progetto generico)
- + complessità minore
- non ancora del tutto operativo

DUE STRADE:



SUBMIT CON SCRIPT BASH

- meno orientato al generico
- complessità maggiore
- + affidabile

? capire quanto ne vale la pena in termini di tempo

SUBMIT CON FILE CFG PER SCRIPT *Severus*

- + già modularizzato (si adatta meglio a un progetto generico)
- + complessità minore
- non ancora del tutto operativo

Proposta prossimi passi:

- ➔ accantonare la submission con script bash e preparazione del codice per creare i file di configurazione di *severus*.
- ➔ test di operatività.
- ➔ creazione sessione di full in modo statico.
- ➔ test operatività full.
- ➔ shift init.
- ➔ session management.

TO-DO - “compiti per le vacanze”

Expert Init *...in progress*

in corso

- Sviluppo di una JobSubmission class con funzionalità di:
 - generazione summary OK
 - generazione script bash OK (da testare)
 - generazione file conf python

Submission Monitor e Shift Init

data da destinarsi

Sub. Monitor: da valutare una soluzione in stile “gangamon” (lista submission con numero di subjobs - click sui subjobs: carica elenco job)

studio

22/lug/2011 – 27/lug/2011

ESAME ARCHITETTURA

webui

27/lug/2011 – 29/lug/2011

MEETING
OGGI

webui

04/ago/2011 – 05/ago/2011

webui

15/ago/2011 – 26/ago/2011

24/lug/2011

31/lug/2011

07/ago/2011

14/ago/2011

21/ago/2011

WEBUI



meeting 21 luglio 2011