Calcolo HTC – perche' e come

TOMMASO BOCCALI – INFN PISA

outline

- Il calcolo sperimentale “seriale” (lo ha definito cosi’ Silvia…)
- Le esigenze
- Le soluzioni
- Le evoluzioni

Calcolo scientifico “seriale”

- Il calcolo scientifico di distingue storicamente (M.Livny, 1983) in
  - Calcolo ad alte prestazioni (HPC)
  - Calcolo ad alto Throughput (HTC)
    - Semi intraducibile: “alta capacita’ di trasmissione”
  - Diffenza e’ quella che c’e’ fra un torpedone e una Ferrari
    - Se devi traspostare una persona dal punto A al punto B, il modo piu’ veloce e’ la Ferrari
    - Se devi traspostare 200 persone fra il punto A e il punto B, il torpedone vince

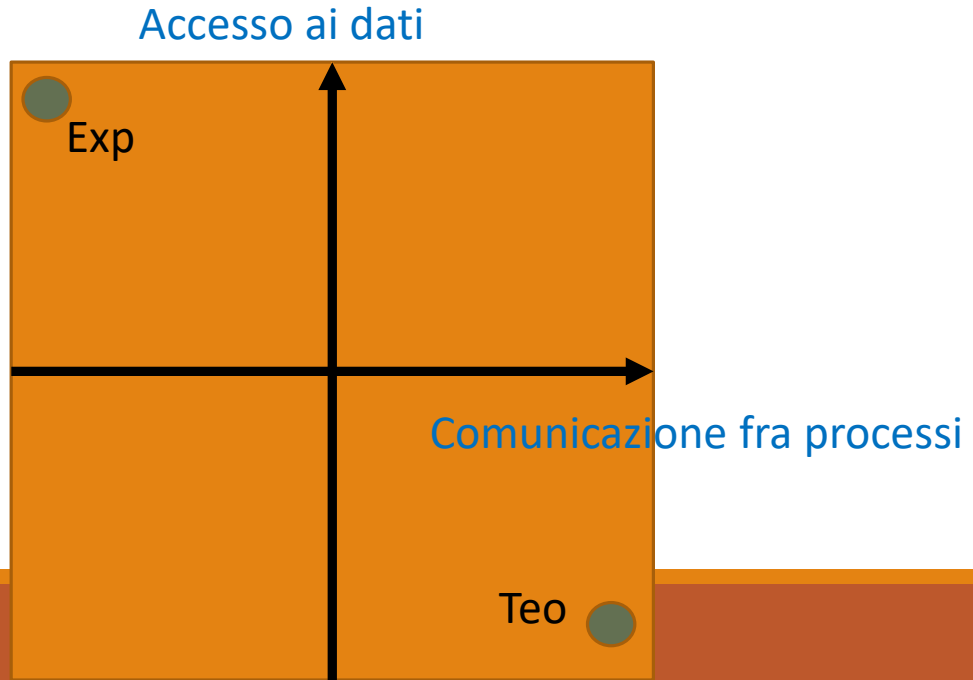
Quindi il tipo di calcolo scientifico …

- In prima approssimazione si distingue fra sistemi
  - In grado di affrontare un singolo problema estremamente complesso (HPC)
  - Affrontare tantissimi problemi mediamente complessi, e scorrelati fra loro (HTC)
    - Esempio tipico della generazione di 1 milione di eventi simulati mediante tecnica MonteCarlo: il processing dei singoli eventi e' indipendente, e non e' necessario che un processore sappia come sta andando l'evento che un altro sta processando
  - … ma non e' finita qui …

Accesso / quantita’ di dati

- Un’altra discriminante forte e’ la necessita / capacita’ di accedere a grandi moli di dati
  - Task informatici che simulano processi fisici raramente hanno bisogno di accedere a grandi moli di dati
    - Pensate a calcolare Pi con metodo MonteCarlo, l’unico dato in input e’ un (singolo?) numero casuale
  - Task che invece analizzano dati sperimentali hanno bisogno di accesso … almeno a tali dati, che possono essere molto grandi

- Tipici processi della Fisica Teorica: HPC, pochi dati
- Tipici processi della fisica Sperimentale: HTC, pochi dati
- (con tante tante eccezioni)



Calcolo seriale spiegato in una frase

- “Ripetere N volte processi semplici, cambiando ogni volta le condizioni di configurazione e I dati di input”
  - Con N milioni se non miliardi

- Il problema diventa quindi come gestire una tale quantita’ di processi, tenerne traccia, gestire fallimenti, ottimizzare il comportamento globale del sistema….

Purtroppo e’ (molto) piu’ complicato di cosi’

- Aggiungete a questo il concetto di calcolo distribuito: invece di prendere le risorse e metterle in un singolo posto, sparetele sul pianeta
  - In almeno 200 siti diversi, in tutti I continenti
  - Con variabilita’ estrema di qualita’ delle risorse
  - Con una rete geografica a volte piu’ simile a un modem che ad un moderno link internet
  - Con tecnologie estremamente differenti
  - Con un meccanismo di accesso uniforme

  - … e avete l’idea della GRID …

Pre GRID

- Calcolo scientifico principalmente su Mainframes
- LEP: una grossa macchina per Esperimento
  - ALEPH (199X): Shift50 = 320 CernUnit. Circa un iPhone 7 !
  - CERN 1995- : passaggio da sistemi proprietari e costosi a farm Linux su PC desktop!

CDC 3400 COMPUTER

The CDC 3400 computer was installed at CERN during the October overhaul as a means of providing some additional on-site computing facilities. It has proved to be a useful additional facility and further it has shown itself to be exceedingly reliable. The 3400 has been installed and maintained by Control Data at no charge to CERN; the duration of its stay at CERN has so far been reviewed with Control Data each fortnight. In order to keep a machine of this type CERN has now sent a letter of intent to CDC to rent a 3000 series computer on site after the 6600 computer is running reliably with SIPROS.

66/268/3/DD/NS/CRS/ng

ALEPH Computing Equipment at CERN

Year	Brand	Processors	CPU (CERN Units)
1984–1990	ALWS VAX Stations	110	60 (1989) – 336 (1994)
–1994	IBM+Siemens VM	2+2	12+13
1988–1990	CRAY	4	32
1994	ALOHA Digital Unix	15	324
1989	FALCON DEC VMS	12	6 (1989) – 27 (1994)
1994–1998	SHIFT 9 SGI	8	136
1996	SHIFT50 DEC Alpha	4	320

Nuove esigenze!

Questo modo di lavorare non puo' funzionare per gli esperimenti LHC!

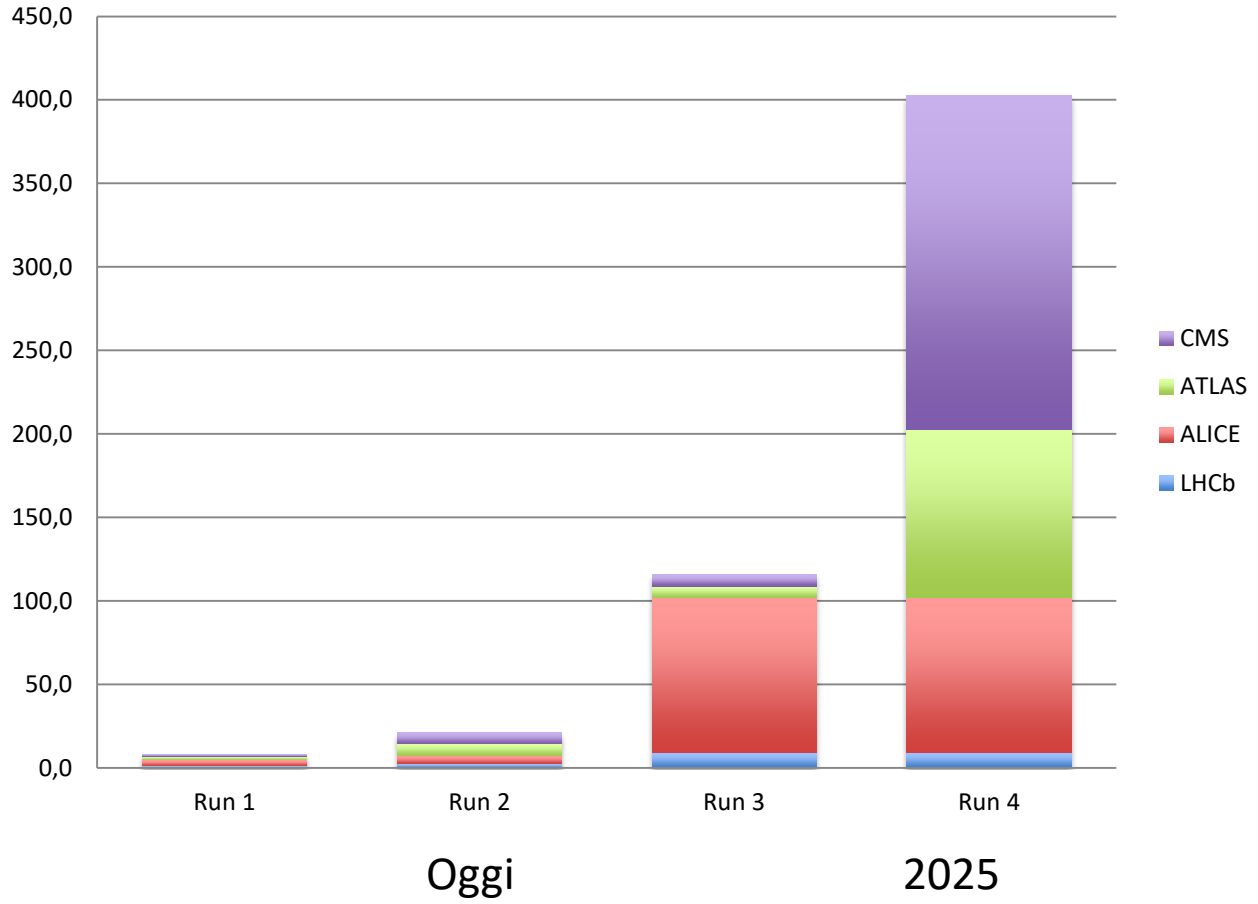
Vediamo perche':

X4: 4 esperimenti!

	ALEPH 1995 	CDF 2004 	CMS 2008 	CMS 2016 
Dimensione dei dati raccolti	**1 TB = 1000 GB** **(1500 CD)**	**1 PB = 1000 TB** **x1000** **(1,5 milioni di CD)**	**~10 PB** **x10** **(15 milioni di CD)**	**~250 PB** **x25**
Capacita' di calcolo (Si2k)	**<<100 k** **(100 CPU attuali)**	**1.4 M** **X50** **(1500 CPU attuali)**	**>25 M** **X20** **(25000 CPU attuali)**	**...** **X10** **(250000 CPU attuali)**

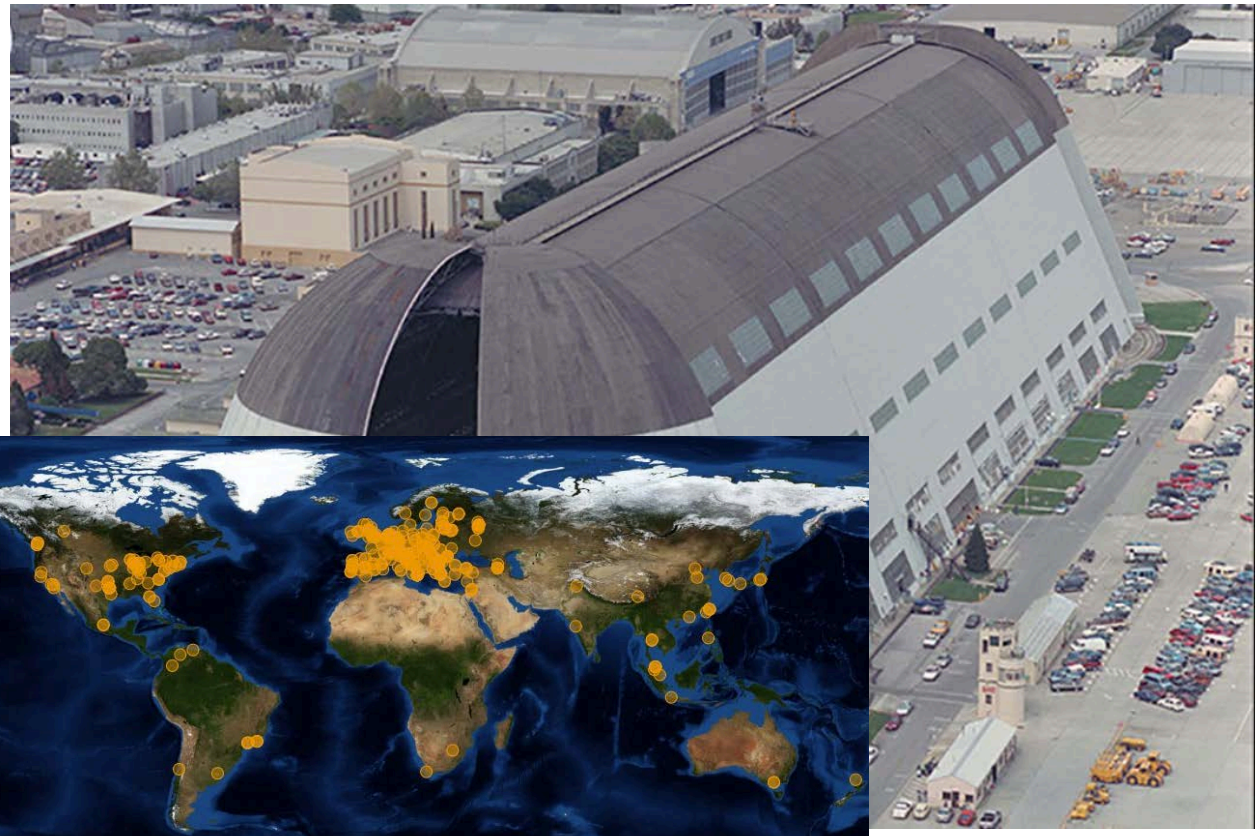
E non e’ finita …

- Lasciamo stare le scale assolute, ma altro fattore 10-100 aspettato per le risorse richieste dagli esperimenti di alte energie nel prossime decennio …
- Dove le metto queste risorse??



Soluzioni ?

1. Sembra che gli hangar degli shuttle siano in affitto … ne prendo uno e ci metto un milione di PC e un miliardo di hard disk
   - Data redundancy? Terremoto? Attacco terroristico? Colpo di Stato?
2. Distribuisco le risorse in giro per il mondo
   - Mi complico certamente la vita, ma almeno ho meno single point of failures
3. (uso servizi commerciali: Amazon etc
   - Perche' no, pero' 10-20 anni fa quando si partiva con la definizione dei modellli di calcolo, la possibilita' non c'era. Ora si)



Google Cloud

Azure

amazon web services

rackspace
the #1 managed cloud company

Notare che …

- Fattore di forma oggi ottenibile al meglio e' ~1000 processing cores per rack
- Questo oggetto consuma ~ 25 kW (~60kEur l'anno compreso il raffreddamento)
- Questo oggetto pesa tra 1 e 1.5 T
- Totali x 1000000 CPU cores:
  - 25 MW
  - 1500 T
- Certamente non gestibili in un unico sito. Tutto il CERN e' sui 100MW. Pisa CED per referenza e' ~ 0.5 MW

Si e' scelto #2: calcolo HTC distribuito

- Ma come fare? Domande a cui la risposta oggi puo' sembrare banale, ma
  1. Come ottengo username e password sui PC in Indonesia?
  2. Come sposto I miei dati da qui al Paraguay (non un file, 1 milione di files)
  3. Come mando in esecuzione 10 Milioni di jobs senza dover premere 1 Milione di volte il tasto enter?
  4. Come posso essere sicuro che in Pakistan ci sia abbastanza RAM per il mio job?
  5. Che compilatore trovero' in Argentina?

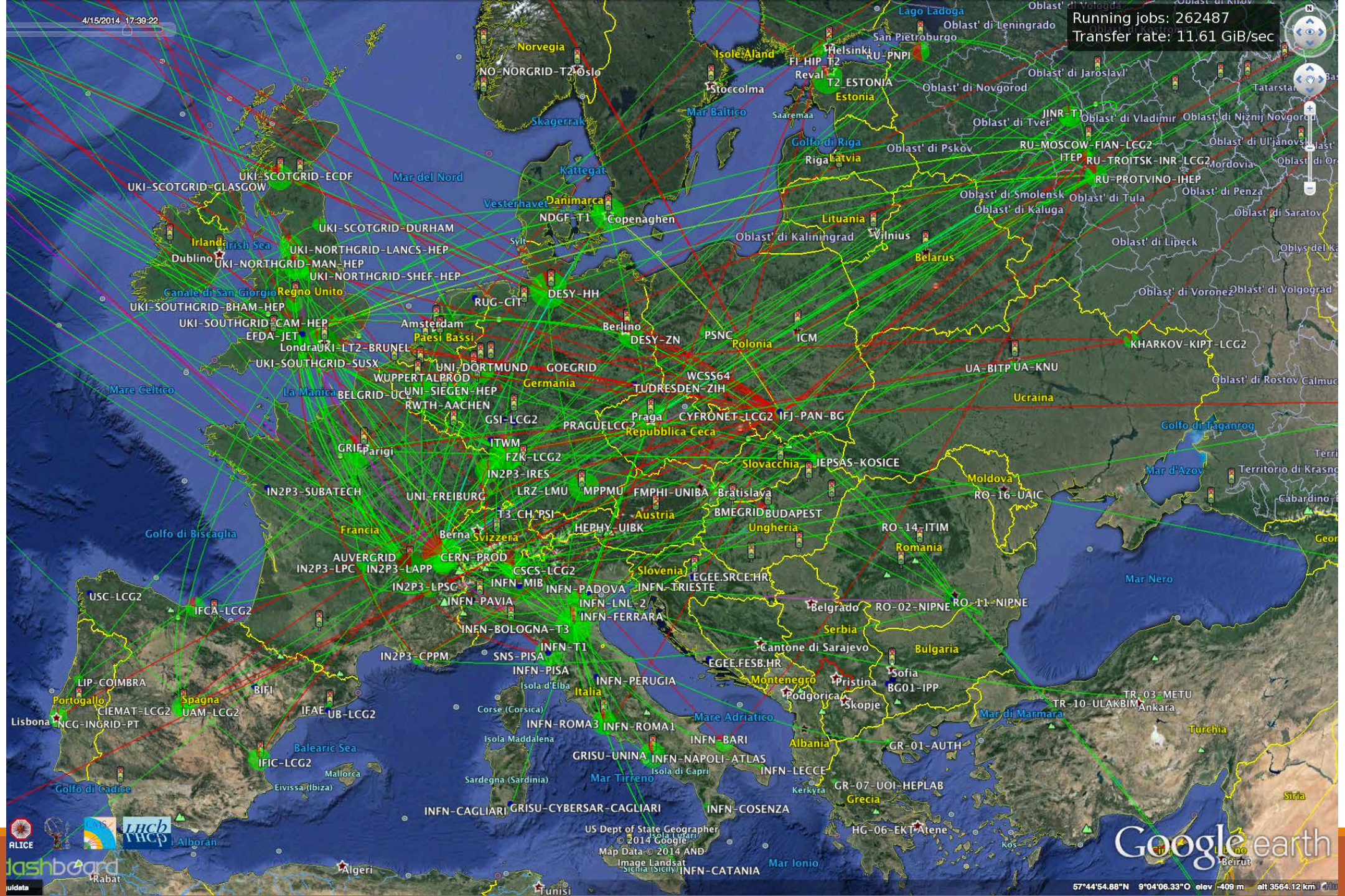
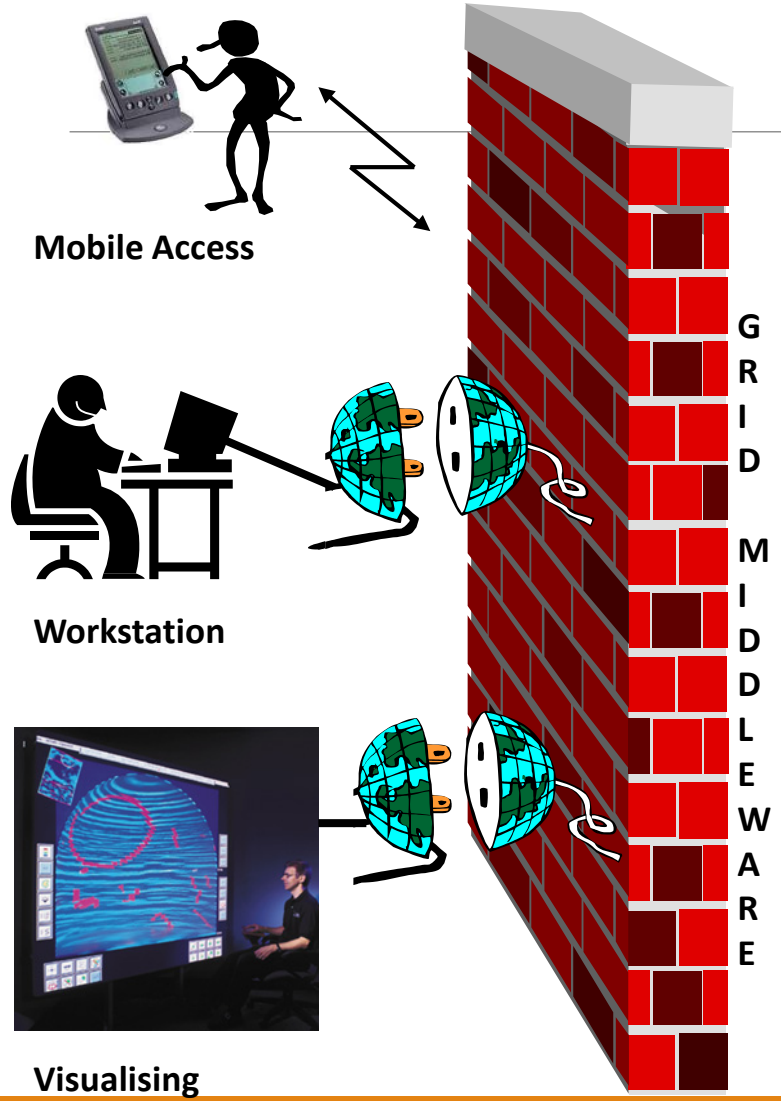
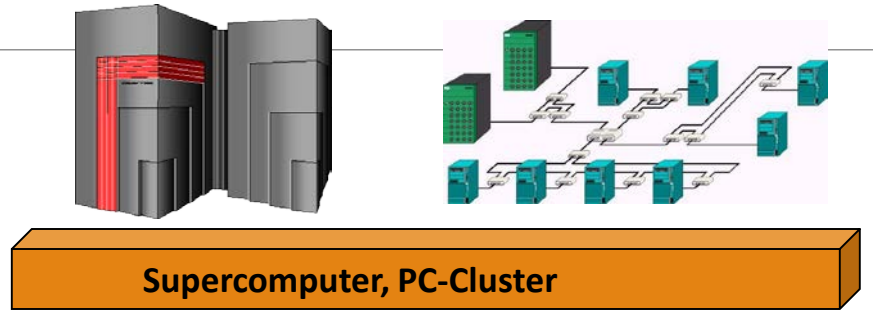
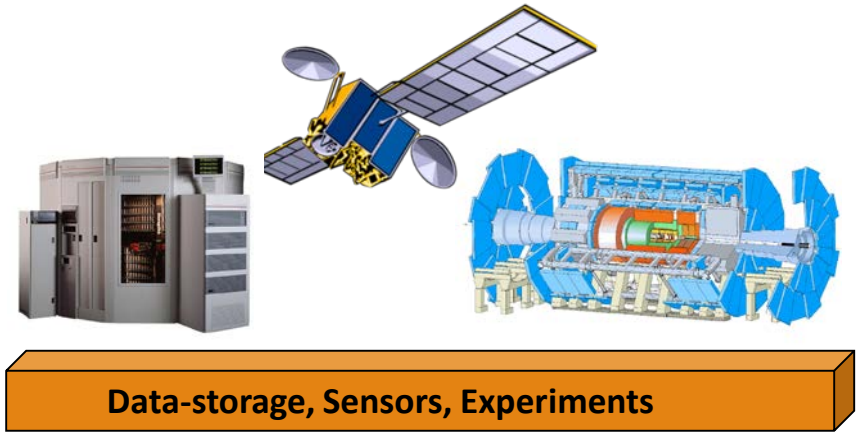

4/15/2014 17:39:22
Running jobs: 262487
Transfer rate: 11.61 GiB/sec
Norvegia
NO-NORGRID-T2-Oslo
Isole Aland
Stoccolma
Helsinki
FI_HIP_T2
RU-PNPI
San Pietroburgo
Lago Ladoga
Oblast' di Leningrado
Reval
T2_ESTONIA
Estonia
Oblast' di Jaroslavl'
Oblast' di Novgorod
Tatarstan
Mar Baltico
Saaremaa
Skagerrak
JINR-T1
Oblast' di Tver'
Oblast' di Vladimir
Oblast' di Nizhnij Novgorod
Golfo di Riga
Riga
Latvia
Oblast' di Pskov
RU-MOSCOW-FIAN-LCG2
ITEP
RU-TROITSK-INR-LCG2
Mordovia
Oblast' di Ul'janovsk
RU-PROTVINO-IHEP
Oblast' di Penza
Oblast' di Tula
Oblast' di Smolensk
Oblast' di Kaluga
Oblast' di Saratov
Kattegat
Mar del Nord
UKI-SCOTGRID-ECDF
UKI-SCOTGRID-GLASGOW
Vesterhavet
Danimarca
NDGF-T1
Copenaghen
Sylt
UKI-SCOTGRID-DURHAM
Irlanda
Irish Sea
Dublino
UKI-NORTHGRID-LANCS-HEP
UKI-NORTHGRID-MAN-HEP
UKI-NORTHGRID-SHEF-HEP
Canale di San Giorgio
Regno Unito
UKI-SOUTHGRID-BHAM-HEP
UKI-SOUTHGRID-CAM-HEP
EFDA-JET
Londra
UKI-LT2-BRUNEL
UKI-SOUTHGRID-SUSX
Amsterdam
Paesi Bassi
RUG-CIT
DESY-HH
Berlino
DESY-ZN
PSNC
Polonia
ICM
Lituania
Oblast' di Kaliningrad
Vilnius
Belarus
Oblast' di Lipeck
Oblast' di Voronez
Oblast' di Volgograd
KHARKOV-KIPT-LCG2
UA-BITP
UA-KNU
Oblast' di Rostov
Calmuc
Ucraina
Mare Celtico
La Manica
BELGRID-UCL
UNI-DORTMUND
GOEGRID
WUPPERTALPROD
Germania
UNI-SIEGEN-HEP
RWTH-AACHEN
WCSS64
TUDRESDEN-ZIH
GSI-LCG2
Praga
CYFRONET-LCG2
IFJ-PAN-BG
PRAGUELCG2
Repubblica Ceca
Golfo di Taganrog
Mar d'Azov
Territorio di Krasno
Cabardino-
GRIF
Parigi
ITWM
FZK-LCG2
Slovacchia
IEPSAS-KOSICE
IN2P3-IRES
Moldova
IN2P3-SUBATECH
UNI-FREIBURG
LRZ-LMU
MPPMU
FMPHI-UNIBA
Bratislava
RO-16-UAIC
T3_CH_PSI
Austria
BMEGRID
BUDAPEST
Golfo di Biscaglia
Francia
HEPHY-UIBK
Ungheria
RO-14-ITIM
Berna
Svizzera
Romania
AUVERGRID
CERN-PROD
IN2P3-LPC
IN2P3-LAPP
CSCS-LCG2
Slovenia
EGEE.SRCE.HR
Mar Nero
IN2P3-LPSC
INFN-MIB
INFN-PADOVA
INFN-TRIESTE
USC-LCG2
INFN-PAVIA
INFN-LNL-2
Belgrado
RO-02-NIPNE
RO-11-NIPNE
IFCA-LCG2
INFN-FERRARA
Serbia
INFN-BOLOGNA-T3
Cantone di Sarajevo
Bulgaria
INFN-T1
IN2P3-CPPM
SNS-PISA
EGEE.FESB.HR
INFN-PISA
Isola d'Elba
INFN-PERUGIA
Montenegro
Pristina
Sofia
LIP-COIMBRA
Italia
Podgorica
BG01-IPP
BIFI
Portogallo
Spagna
CIEMAT-LCG2
UAM-LCG2
IFAE
UB-LCG2
Corse (Corsica)
Skopje
TR-03-METU
TR-10-ULAKBIM
Ankara
Lisbona
NCG-INGRID-PT
INFN-ROMA3
INFN-ROMA1
Mare Adriatico
Mar di Marmara
Turchia
Isola Maddalena
INFN-BARI
Albania
Balearic Sea
GR-01-AUTH
GRISU-UNINA
INFN-NAPOLI-ATLAS
IFIC-LCG2
Mallorca
Isola di Capri
INFN-LECCE
Sardegna (Sardinia)
Mar Tirreno
Kerkyra
GR-07-UOI-HEPLAB
Golfo di Cadice
Eivissa (Ibiza)
Grecia
Siria
INFN-CAGLIARI
GRISU-CYBERSAR-CAGLIARI
INFN-COSENZA
US Dept of State Geographer
© 2014 Google
Map Data © 2014 AND
Image Landsat
HG-06-EKT-Atene
Kos
Libano
Beirut
Alboran
Algeri
Rabat
Tunisi
Mar Ionio
INFN-CATANIA
Google earth
57°44'54.88"N 9°04'06.33"O elev -409 m alt 3564.12 km

Diagramma storico: il MW wall







Il povero fisico non deve e non vuole sapere che c'e' dal lato destro del muro.
Il MiddleWare GRID lo scherma dalla complessita' del sistema, e gli permette di lavorare

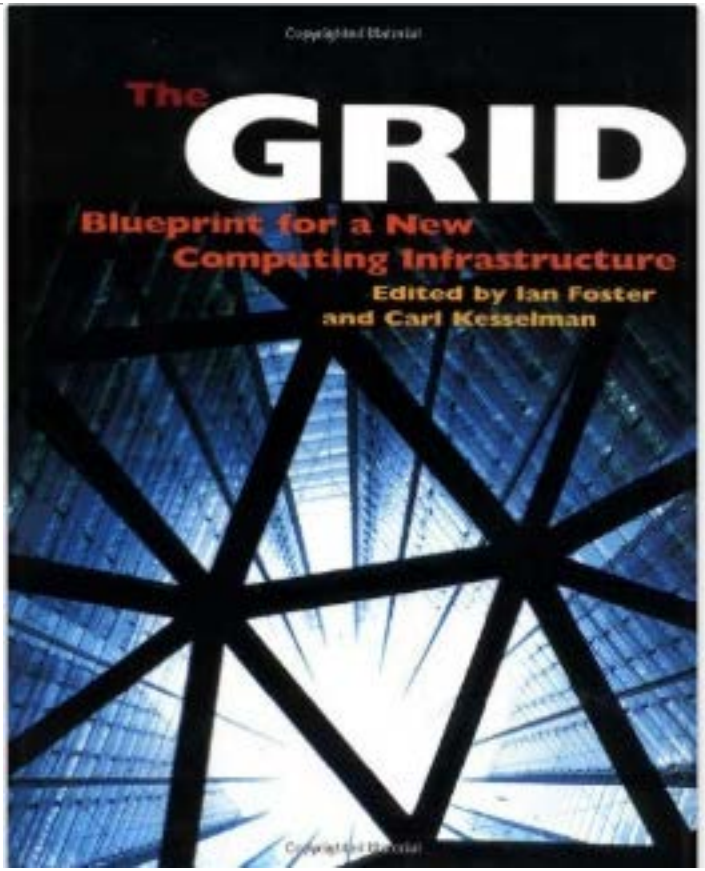
15 anni dopo non e' possibile dire che le cose abbiano funzionato benissimo, ma di certo in qualche modo hanno funzionato…

GRID non e' un acronimo …

- E' solo un riferimento alla Power Grid (la rete elettrica, l'ENEL insomma….)
  - Avere calcolo (scientifico) deve essere facile come attaccare il frullatore ad una presa di corrente
  - Quando lo fare, non vi chiedete
    - Ma la corrente che uso sara' di origine idroelettrica o termoelettrica?
    - Ma passera' per linee a 38 kV o 64 kV?
    - Sara' prodotta in Germania o in Italia?
    - Ma questa presa regge 100 W o 1 kW?
  - … perche' qualcuno vi assicura che e' adatta alle vostre esigenze (220 V, 50Hz, max XX Euro/kWh …)
  - La GRID fa lo stesso (almeno nei sogni): fa da layer fra la complessita' reale del sistema e l'utente

Idea iniziale di informatici

- Questo libro
- Globus Toolkit: middlware basilare per replicare su Grid i comandi piu semplici:
  - Ls → globus-url-copy
  - Exec pippo & → globus-job-run pippo
- Ma da qui in poi grossi(ssimi) progetti EU e US
  - DataGRID, EGEE, OSG, …..
  - Che hanno creato framework molto piu' complessi



Adesso potrei passare 2 ore a descrivere…

- .. I fantastici sistemi gLite, WMS, RB, CE e altre sigle amene
- 5 anni fa lo avrei fatto, peccato che adesso non li usa quasi piu' nessuno
- Si trattava di sistemi molto complessi, in teoria indipendenti da esperimenti, per

1. Utilizzare la grid come un batch system locale
2. Avere accesso remoto a files indipendentemente dalla loro posizione
3. Avere tutta la flessibita' possibile per specificare priorita' di persone / gruppi / sottogruppi / nazionalita' …

In gran parte non piu' rilevante …

Perche'?

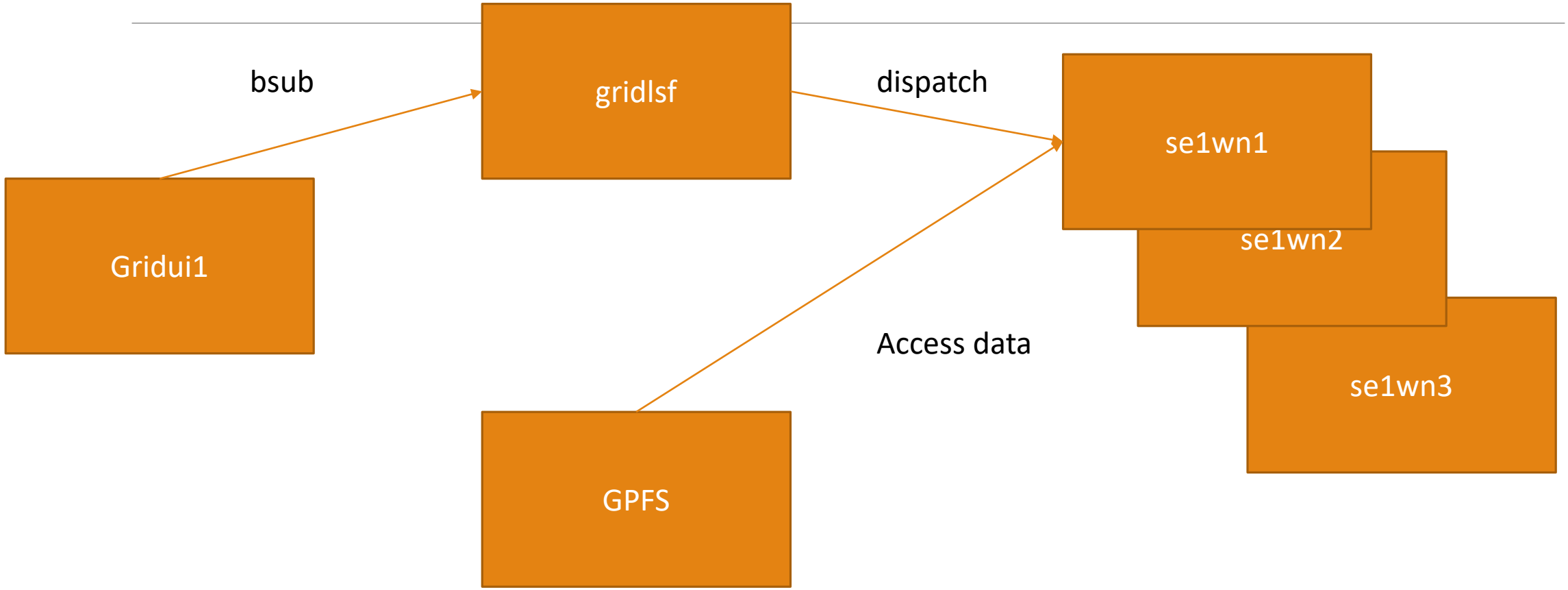
- Tanti motivi ….

1. Approccio a "un grande LSF per tutte le comunita' scientifiche del mondo" appariva ok per gruppi piccoli, ma non abbastanza flessibile per I grossi utenti con necessita' specifiche
2. Servizi centrali di tali dimensioni (centinaia di migliaia di jobs running, milioni in coda) non possono certamente essere singoli
   - Si era finiti su un partizionamento estremo … in pratica un utente aveva a disposizione N sistemi LSF e doveva scegliere su quali mandare … e non c'era bilanciamento fra questi
3. I tool di spostamento dati centrali erano ottimizzati per trasferire files interi, e grossi (grande overhead di handshaking), mentre il calcolo scientifico si stava velocemente spostando verso accessi remoti ottimizzati
4. La non trascurabile voglia degli esperimenti di fare tutto loro, senza usare soluzioni preconfezionate…

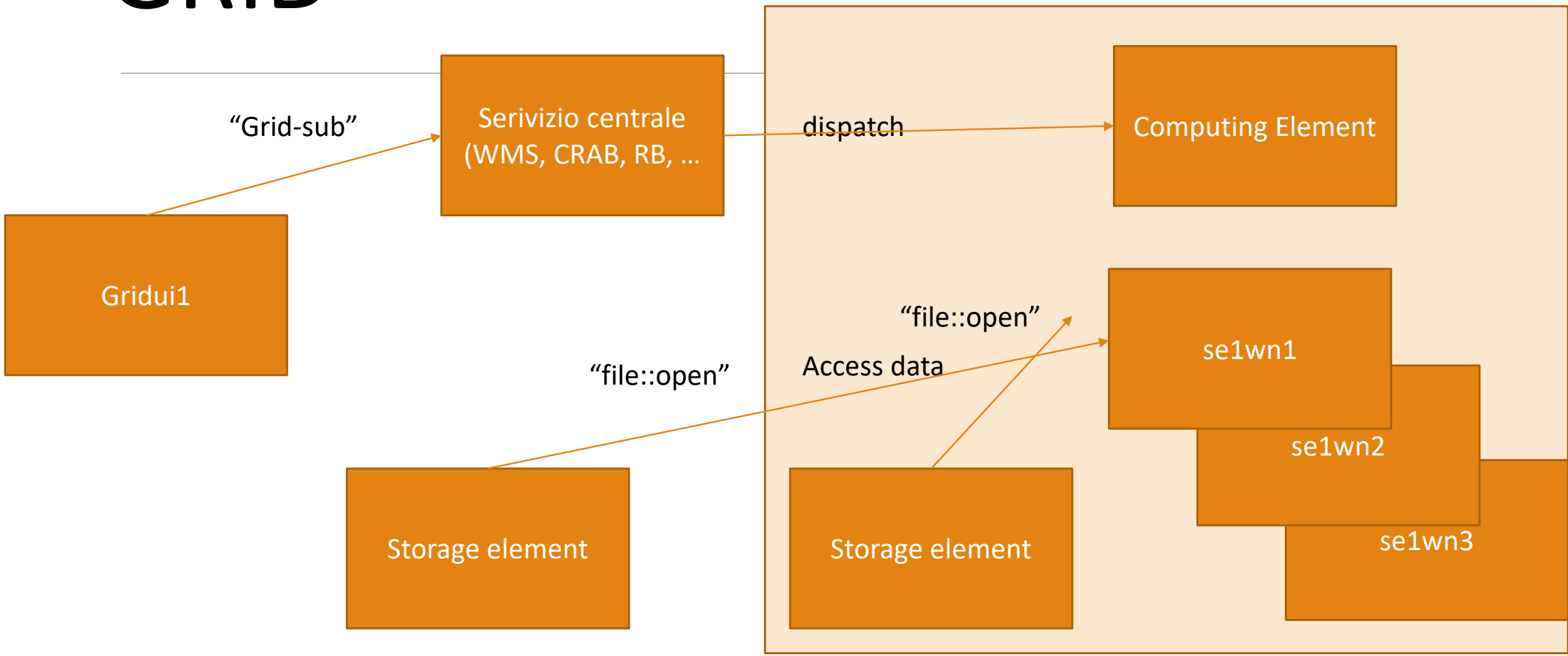
Struttura elementare di un sito GRID

- Non particolarmente diversa da LSF (che continuo ad usare come esempio)
  - Una macchina su cui avete login fisico (ssh), di solito presso il vostro istituto o centralmente (CERN, FNAL, …)
  - Un servizio a cui sottomettete jobs (l'equivalente del Master LSF)
  - Dei nodi di calcolo che ricevono jobs dal Master e li eseguono
  - Dello storage

LSF


bsub
gridlsf
dispatch
se1wn1
se1wn2
se1wn3
Gridui1
Access data
GPFS

GRID


“Grid-sub”
Serivizio centrale
(WMS, CRAB, RB, …
dispatch
Computing Element
Gridui1
“file::open”
Access data
“file::open”
se1wn1
se1wn2
se1wn3
Storage element
Storage element

Da un certo punto di vista…

- La GRID e' un batch system di batch system, che ha due livelli
  - Dal manager centrale si sceglie il sito fra N siti
    - Potete farlo voi o lasciare al sistema la scelta
  - Una vosta sul sito, si sceglie il WorkerNode fra N WorkerNodes

- (questo perche' una struttura ad un livello solo sarebbe equiparabile a un batch system da 1000000 di WorkerNodes sotto: ingestibile … o no?)

Breve tutorial accesso “low level” a GRID

- Low level: senza utilizzare framework specifici di esperimento
  - Se siete di ATLAS, CMS, Belle, non farete mai una cosa di questo tipo; ci sara’ un framework che lo fa per voi
  - Pero’ sempre bene sapere “che succede internamente”
  - Vediamo velocemente I comandi da usare per le cose piu’ basilari… prendendo come riferimento LSF (ovvero un batch system locale)
    - “login” sulla grid
    - “definizione “del job che vogliamo girare (paralleleo con config di LSF)
    - “richieste del job”
    - “accesso” allo storage
  - Se volete un tutorial che spieghi tutti I concetti di GRID, potete utilizzare questo per esempio: TUTORIAL GRID AL CNAF

LSF in una slide

- Sottomettere jobs
  - bsub –q NOME_DELLA_CODA pippo.job
- Controllare I miei jobs
  - bjobs [-l|-a] [NUMERO_DEL_JOB]
- Cancellare un job
  - bkill [NUMERO_DEL_JOB]
- Controllare stato delle code
  - bqueues
- Tutto piu’ o meno spiegato qui

```
-bash-3.2$ bqueues
QUEUE_NAME      PRIO STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN  SUSP
fai               99  Open:Active      -    -    -    -     0     0     0     0
faihmem           99  Open:Active      -    -    -    -     4     0     4     0
fai5              99  Open:Active      -    -    -    -     0     0     0     0
gpu               99  Open:Active      -    -    -    -     1     0     1     0
phi               99  Open:Active      -    -    -    -     0     0     0     0
cert              70  Open:Active      -    -    -    -     1     1     0     0
cmsmcore          60  Open:Active      -    -    -    -  4424  2891  1112     0
local             50  Open:Active    900    -    -    -   151    20   131     0
locallong         50  Open:Active     80    -    -    -     0     0     0     0
theonuc           50  Open:Active      -    -    -    -     7     0     7     0
cms               30  Open:Active   3000    -    -    -    51     0    51     0
atlas             30  Open:Active      -    -    -    -     0     0     0     0
cdf               30  Open:Active      -    -    -    -     0     0     0     0
alice             30  Open:Active      -    -    -    -     0     0     0     0
lhcb              30  Open:Active      -    -    -    -   140     5   135     0
babar             30  Open:Active      -    -    -    -     0     0     0     0
superb            30  Open:Active      -    -    -    -     0     0     0     0
belle             30  Open:Active      -    -    -    -  2216   164  2052     0
calet             30  Open:Active      -    -    -    -   774   174   600     0
magic             30  Open:Active      -    -    -    -     0     0     0     0
theophys          30  Open:Active      -    -    -    -     0     0     0     0
virgo             30  Open:Active      -    -    -    -     0     0     0     0
na48              30  Open:Active      -    -    -    -     0     0     0     0
glast             30  Open:Active      -    -    -    -     0     0     0     0
test              10  Open:Active      -    -    -    -     0     0     0     0
compchem          10  Open:Active      -    -    -    -   258     0   258     0
biomed            10  Open:Active      -    -    -    -    72    72     0     0
grid              10  Open:Active      -    -    -    -     0     0     0     0
diagnosis          5  Open:Active      -    -    -    -     0     0     0     0
```

Il login

- LSF:
  - Dovete fare login su una macchina che veda il batch system (cioe' che accetti I comandi bsub, bqueues etc) – per esempio le gridui a Pisa
  1. SSH PIPPO@GRIDUI1.PI.INFN.IT

- GRID
  1. SSH PIPPO@GRIDUI1.PI.INFN.IT
     1. Ma questo non ci identifica sulla GRID, solo su quella macchina
     2. Serve anche qualcosa a livello globale : il certificato GRID e il PROXY!

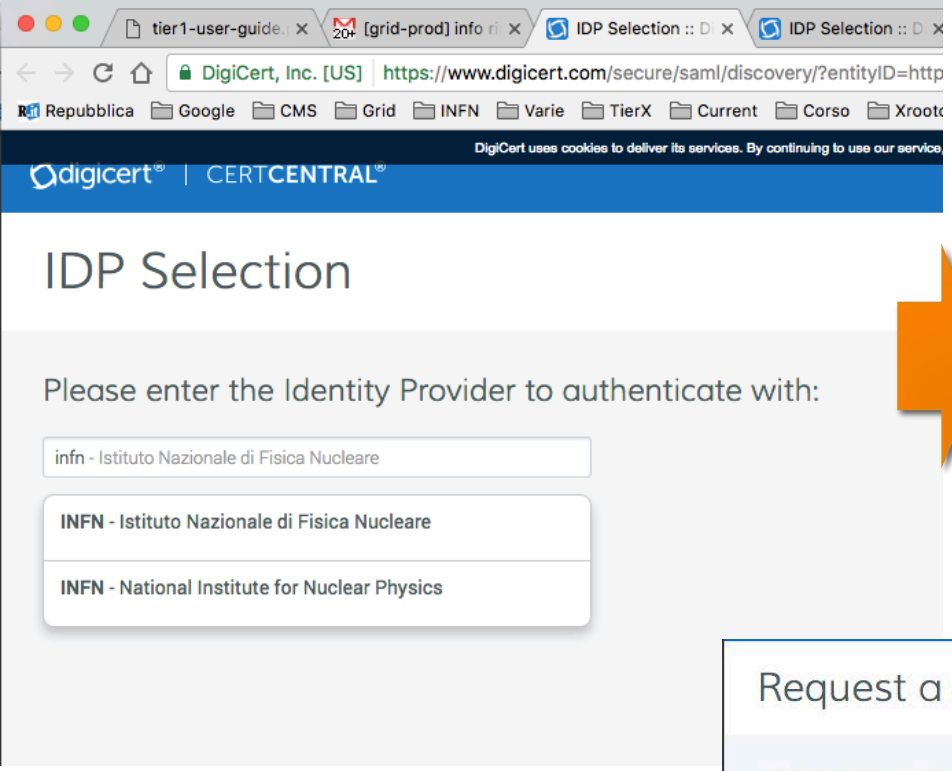
  1. Cosa fanno?
     1. Permettono AUTENTICAZIONE: garantiscono che voi siete voi
     2. Permettono AUTORIZZAZIONE: il "possessore delle risorse" (CMS in questo caso) garantisce che voi potete usare le sue risorse

Come funziona un certificato?

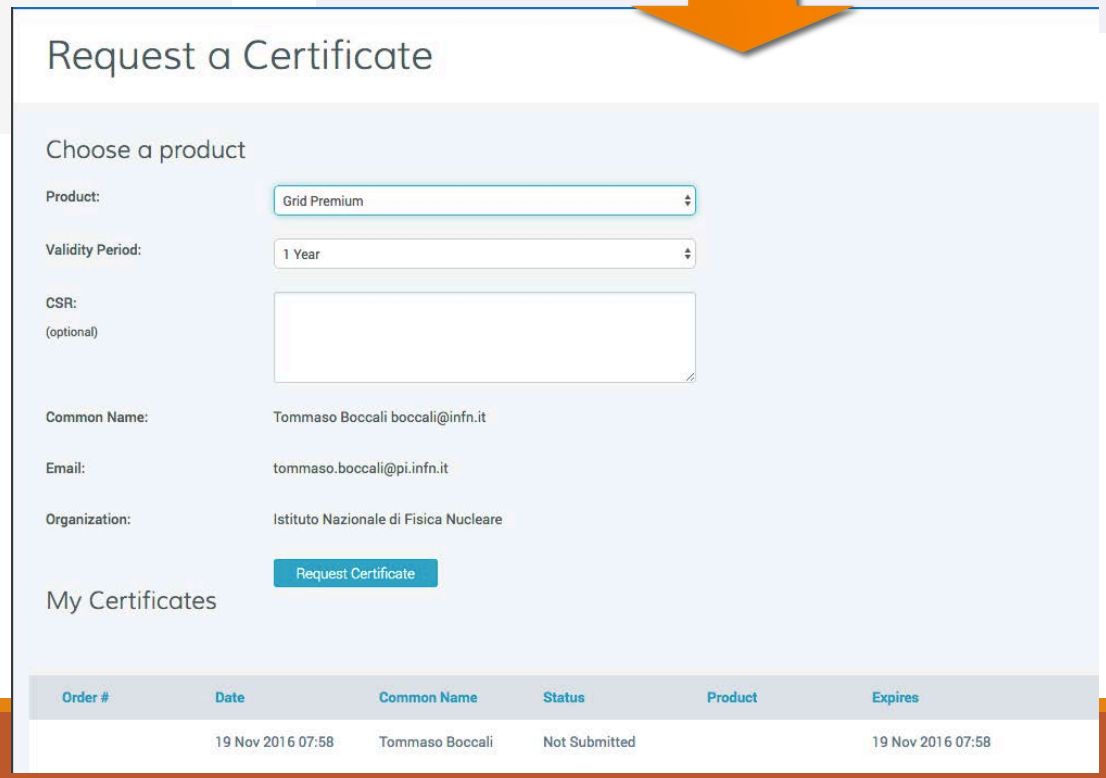
- Le istituzioni che contribuiscono alla GRID si sono accordate per un patto di mutua fiducia:
  - Se INFN dice che io sono Mario Rossi, STFC (UK) ci crede!
- Questo avviene mediante Certification Authorities, che rilasciano a ciascuno dei loro associati delle credenziali segrete (il certificato appunto)
- Questo funziona mediante sistema a chiavi pubbliche/private: in qualunque momento io posso provare alla mia CA che sono io, usando le credenziali che la CA stessa mi ha dato
- Un computer a Boston quando riceve un “login” da parte mia, chiede alla CA INFN: sei sicuro che questo sia Mario Rossi? In caso di risposta affermativa, si fida e basta

Chi mi da’ il certificato?

- Un tempo era una procedura non banalissima:
  - parlare con persone, mostrare carte di identita’, etc etc
- Adesso se avete un login INFN (quello per aprire le missioni, per intenderci…), e’ qustione di un paio di click: via Geant, si usa Digicert per tutto
  - https://www.digicert.com/sso :


tier1-user-guide
[grid-prod] info
IDP Selection :: D
IDP Selection :: D
DigiCert, Inc. [US]
https://www.digicert.com/secure/saml/discovery/?entityID=http
Repubblica
Google
CMS
Grid
INFN
Varie
TierX
Current
Corso
DigiCert uses cookies to deliver its services. By continuing to use our service
digicert® | CERTCENTRAL®
IDP Selection
Please enter the Identity Provider to authenticate with:
infn - Istituto Nazionale di Fisica Nucleare
INFN - Istituto Nazionale di Fisica Nucleare
INFN - National Institute for Nuclear Physics


INFN AAI
INFN Identity Check
Username: boccali
Password: ••••••••
Login
X.509 Certificate
Come ottenere un accesso ad INFN-AAI
Kerberos5 GSS-API
Cambio o Rigenerazione Password - Recupero Username
NON AGGIUNGERE QUESTA PAGINA AI PREFERITI! Dopo il login verrai rediretto a
Richieste di supporto e domande a aai-support@lists.infn.it
This is WAWA (Widely Assorted Web Authenticator) by Dael Maselli, based on a SAML Identity Provider running simpleSAMLphp by Feide


Request a Certificate
Choose a product
Product: Grid Premium
Validity Period: 1 Year
CSR:
(optional)
Common Name: Tommaso Boccali boccali@infn.it
Email: tommaso.boccali@pi.infn.it
Organization: Istituto Nazionale di Fisica Nucleare
Request Certificate
My Certificates
Order # Date Common Name Status Product Expires
19 Nov 2016 07:58 Tommaso Boccali Not Submitted 19 Nov 2016 07:58

Non basta #1

- In questo modo sarete **AUTENTICATI**, ma non **AUTORIZZATI**
  - **Cioe' siete Mario Rossi, ma non e' garantito che siate parte di CMS**
- Per essere autorizzati, serve REGISTRARVI presso l'esperimento a le cui risorse volete accedere

- Per gli esperimenti LHC:
  - https://voms2.cern.ch:8443/voms/<NOMEESPERIMENTO>/register
- Per gli altri consultare la documentazione di esperimento
- Una volta avuto l'ok …

Non basta #2

- Il certificato sara' a quel punto nel browser, va **esportato** (browser dependent) **in formato .p12** (cert.p12) e messo nella cartella .globus nella vostra home sulla macchina che usate per accedere a GRID:

```
cd $HOME
mkdir .globus
cd .globus
openssl pkcs12 -clcerts -nokeys -in cert.p12 -out usercert.pem
openssl pkcs12 -nocerts -in cert.p12 -out userkey.pem
chmod 600 usercert.pem
chmod 400 userkey.pem
```

The files must have the following permissions:

```
-rw------- 1 tentids darkside 1793 Jan 14 14:23 usercert.pem
-r-------- 1 tentids darkside 2002 Jan 14 14:23 userkey.pem
```

Provare ….

```
[boccali@faiwn4 TESTCMS]$ voms-proxy-init -voms cms
Enter GRID pass phrase:
Your identity: /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=tboccali/CN=447815/CN=Tommaso Boccali
Creating temporary proxy ............................. Done
Contacting  lcg-voms2.cern.ch:15002 [/DC=ch/DC=cern/OU=computers/CN=lcg-voms2.cern.ch] "cms" Done
Creating proxy ....................................................... Done

Your proxy is valid until Sat Nov 19 20:13:38 2016
```

- Una volta ottenuto installato il certificato, potete ottenere il proxy

- E verificarlo …

```
[boccali@faiwn4 TESTCMS]$ voms-proxy-info -all
subject   : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=tboccali/CN=447815/CN=Tommaso Boccali/CN=proxy
issuer    : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=tboccali/CN=447815/CN=Tommaso Boccali
identity  : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=tboccali/CN=447815/CN=Tommaso Boccali
type      : proxy
strength  : 1024 bits
path      : /tmp/x509up_u1534
timeleft  : 11:59:43
key usage : Digital Signature, Key Encipherment
=== VO cms extension information ===
VO        : cms
subject   : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=tboccali/CN=447815/CN=Tommaso Boccali
issuer    : /DC=ch/DC=cern/OU=computers/CN=lcg-voms2.cern.ch
attribute : /cms/Role=NULL/Capability=NULL
timeleft  : 11:59:43
uri       : lcg-voms2.cern.ch:15002
```

Ma cosa e' un proxy di preciso?

- Certificato:
  - **Dimostra che siete voi; dura un anno tipicamente**
  - Un processo su una macchina in Brasile per agire a nome vostro (leggere un vostro file, o usare CPU) ne ha bisogno
  - Ma se qualcuno ve lo ruba, poi per un anno puo' fare finta di essere voi → troppo rischioso
    - E' come mettere in un job che finira' chissa' dove il vostro username e password …
- Proxy:
  - Un "**certificato derivato**" dal certificato di cui sopra, che pero' dura solo poche ore
  - E' quello che volete "mandare" in Brasile: se qualcuno ve lo rubasse, il danno e' limitato a poche ore. Poi "scade"

I job …

- (nota: se usiamo I servizi base di GRID, nella pratica gli esperimenti LHC non possono funzionare, perche' usano altre cose – vedere dopo. Per cui qui metto istruzioni generiche)

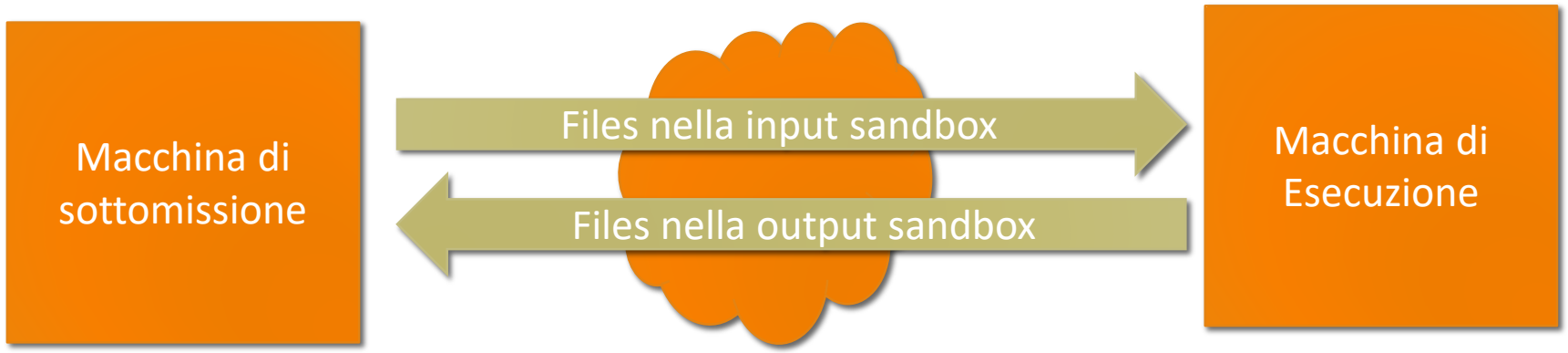
- Il linguaggio JDL (Job Description Language) serve per specificare
  - Cosa girare (eseguibile, parametri, …)
  - Cosa e' necessario perche' funzioni (tipo di sistema operativo, quantita' di RAM, …)
  - Dove girarlo (voglio Pisa, voglio l'Italia, …)
  - Esempio semplice:

- Il "job" e' niente altro che un file JDL

```
Executable="/bin/hostname";
StdOutput="stdout";
StdError="stderr";
OutputSandbox={"stdout", "stderr"};
```

- **Nome dell'eseguibile** (che e' nella directory locale quando sottomettete … vedi dopo)
- **Parametri da linea di comando**
- Su che file (locale sulla macchina dove avverra' l'esecuzione) ridirigere **STDOUT e STDERR**
- Quali files **copiare** dalla macchina di sottomissione a quella di esecuzione prima di cominciare
- Quali files **copiare** dalla macchina di esecuzione a quella di sottomissione dopo l'esecuzione

```
Executable = "cw6v15g3.sh";

Arguments = "Run41117_d85.mac test/v0.9.1/Kch2pinunu";

StdOutput = "run41117.out";

StdError = "run41117.err";

InputSandbox = {"cw6v15g3.sh", "Run41117_d85.mac", "input_files.tgz"};

OutputSandbox = {"run41117.out", "run41117.err"};
```



Altri tag utilizzabili:

- **Requirements = RegExp (”pi.infn.it", other.GlueCEUniqueID);** → gira a Pisa
- **Requirements = (other.GlueCEPolicyMaxCPUTime>60*24);** → gira su code che permettano almeno 24 hore di esecuzione
- **Requirements = (other.GlueCEStateFreeCPUs>2000);** → gira su un sito che abbia almeno 2000 cpu libere (??)

Comandi per interagire con la GRID:

- Sottometti un JOB (da una macchina configurata per voi → chiedere a centro di calcolo)
  - **glite-wms-job-submit –a job.jdl**

```
====================== glite-wms-job-submit Success ======================
=

The job has been successfully submitted to the WMProxy
Your job identifier is:

https://wms005.cnaf.infn.it:9000/3O4tJAjJGt1YLpX7rqEvKA

==========================================================================
```

- Controllare lo stato di un job
  - **glite-wms-job-status <jobID>**

- Una volta finito il job, prendere l’output (la output sandbox)
  - **glite-wms-job-output <jobID>**

Lo storage …

- E’ piu’ complicato dell’utilizzo CPU, purtroppo.
- L’idea e’ che mediante il proxy visto prima, si riesca ad accedere a qualunque file nel mondo a cui si sia autorizzati
- Idea iniziale di GRID: nascondere completamente la complessita’ di
  - Dove sta il file
  - Quale sia il suo nome fisico (/tmp/pippo.root) sullo storage dove e’
  - Quale protocollo io debba utilizzare per accedervi
- Il tutto semplicemente dietro un nome logico: il file per me si chiama **my_file**, GRID per favore scrivilo da qualche parte e fai in modo che poi possa accedervi …

Il nome del file …

- L'idea e' appunto una distinzione fra **LogicalFileName**
  - il nome che io voglio are a quel file – LFN
- E **PhysicalFileName**
  - che comprende: **protocollo** per l'accesso, quale **macchina** contattare, opzioni varie, e **nome del file** su quello storage – PFN o SURL
- La relazione e' 1 a N: lo stesso LFN puo' essere un due repliche, quindi avere associati 2 PFN

- L'LFN nelle intenzioni iniziali serviva per **dare l'impressione** che la GRID fosse un unico singolo filesystem, globalmente accessibile e potenzialmente di dimensioni infinite

Comandi di base

- Creare un file (cr = copy and register)
  - **`lcg-cr [-v] --vo cms file:///home/tom/a.txt -l lfn:test_tommaso`**
  - Il file va automaticamente sul closeSE, a meno che non si specifichi direttamente un SE: -d castorgrid.cern.ch
- Cercare un file (lr = list replicas)
  - **`lcg-lr [-v] --vo cms lfn:test_tommaso`**
- Copiare un file da un posto all'altro (rep = replicate)
  - **`lcg-rep [-v] --vo cms lfn:test_tommaso -d pccmsgrid09.pi.infn.it`**
- Cancellare un file da un posto (del = delete)
  - **`lcg-del [-v ] --vo cms lfn:test_tommaso -s castorgrid.cern.ch`**
- Cancellare tutte le copie di un file
  - **`lcg-del [-v ] --vo cms lfn:test_tommaso -a`**

Perche’ tutto quello che vi ho raccontato fino ad ora e’ abbastanza accademico …

- Vari motivi:
  - Questi sono comandi per gestire singoli jobs / singoli files; sempre piu’ spesso si ha a che fare con migliaia di jobs e milioni di files. Per questo serve un’infrastruttura in mezzo che gestisca il salto di complessita’ e non mandare comandi singoli
  - Come detto prima, in presenza di milioni di jobs/ oggetti questi servizi generici non sono ottimizzati, e I grossi gruppi sperimentali hanno preferito scrivere prodotti proprietari
  - La gestione dei files, soprattutto, si e’ nel frattempo complicata: I files non vengono solo spostati come nei comandi lcg-*, ma vengono cachati, acceduti in remoto, acceduti contemporanemaente da piu’ repliche …
  - … e poi: la GRID non e’ piu’ l’unica risorsa da utilizzare!

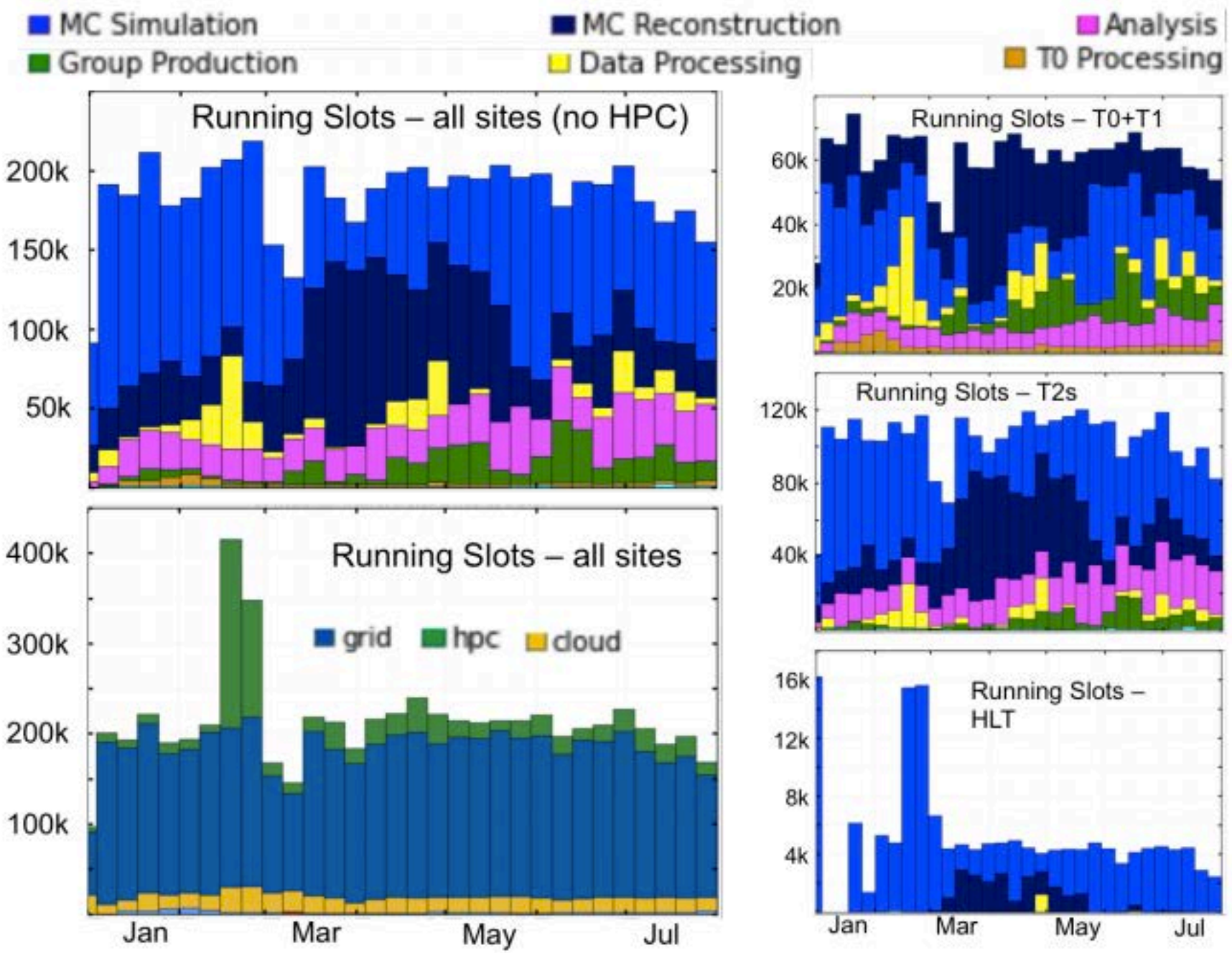
Esempi di grossi framework di sottomissione e accesso alle risorse

- **Dirac**
  - LHCb, Belle, …:
- **Panda:**
  - ATLAS, LSST, …
- **WMAgent, CRAB**
  - CMS, …
- Sono oggetti che integrano nello stesso sistema
  - Gestione dei jobs (dove / come mandarli)
  - Gestione dei files (dove metterli, spostarli, accedervi, ….)
  - Gestione dei workflow
    - Produzione, reprocessing, analisi, qualunque cosa
    - Prioritizzazione fine

- Con questi:
  - Non vedrete mai I comandi descritti qui, tipicamente tutto avviene o tramite webservices (dal browser) e da comandi diversi
  - Servizi entrali di esperimento interagiranno con la GRID al posto vostro
  - Sono oggettivamente troppo experiment dependent per parlarne qui, vediamo l'approccio generale che usano …

Sistemi in azione

ATLAS: 200k jobs running praticamente sempre. Picchi di 400k



Oltre la GRID …

- Anche se il calcolo dei grossi esperimenti oggi segue in massima parte il paradigma GRID, la soluzione evolvera' nel prossimo decennio, per due ordini di motivi

1. GRID e' stata pensata come un'unione di centri di calcolo **piu' o meno stabili e gestiti da personale esperto** negli esperimenti - ma il personale costa da alcune parti piu' delle risorse stesse (US)
2. Esistono nel mondo (sia scientifico sia commerciale) risorse di calcolo estremamente superiori a quelle che utilizziamo noi, e "succede" che ci vengano date disponibili per brevi periodi a prezzi bassi o nulli … pero' trasformarle in un sito GRID non e' sensato, vista la dinamicita'
   - Inoltre, pensate a SETI@Home e alle risorse che e' in grado di mobilitare..

Costi del personale: la Cloud!

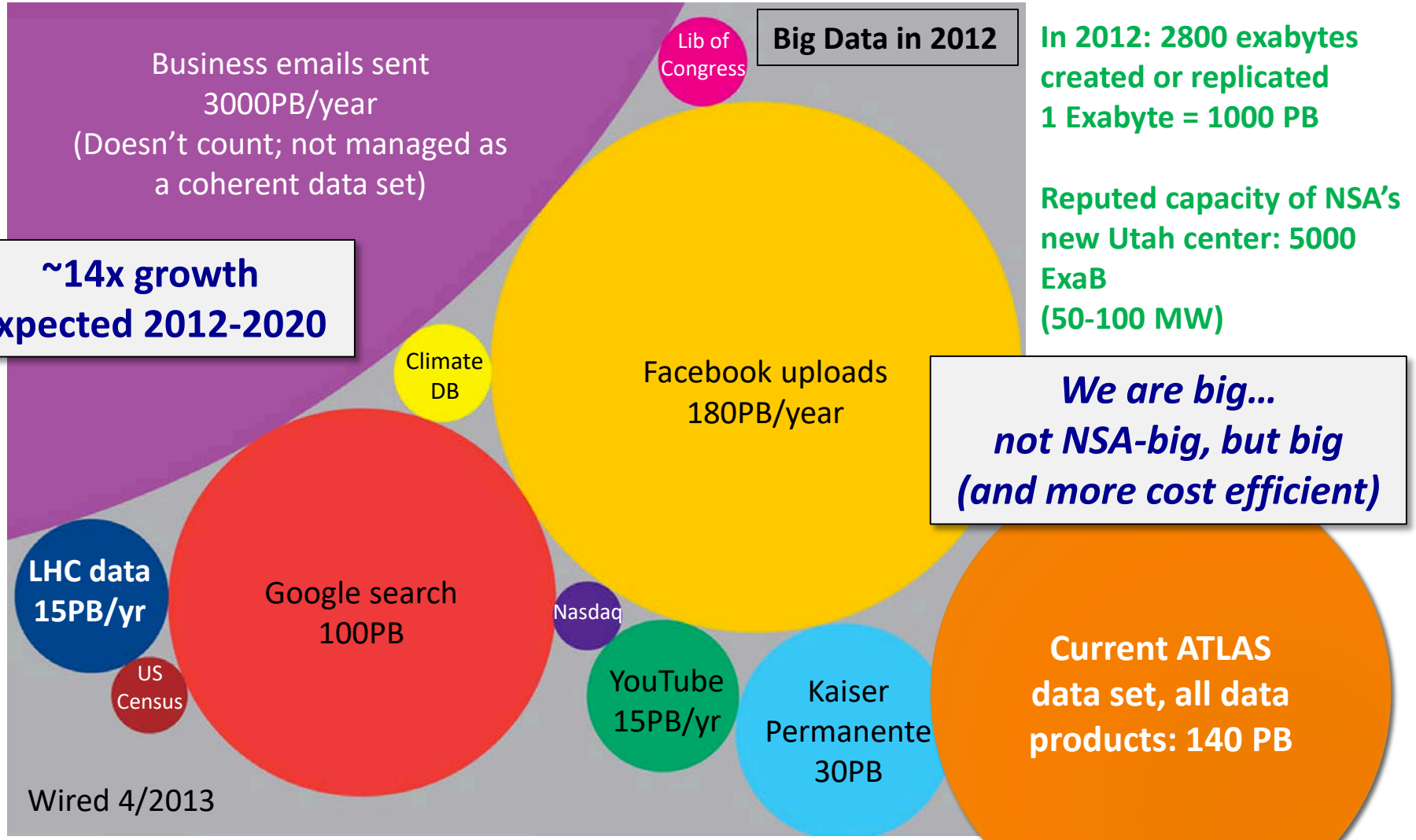
- Abbiamo visto gli esempi di jobs semplici via GRID
- Io mando sostanzialmente in esecuzione uno script, che suppone che ci sia gia' un'installazione del software che mi serve nel sito di esecuzione
  - Qualcuno deve averlo installato, e darmi supporto nel caso di problemi
  - E se per esempio ho un problema dovuto alla versione particolare del compilatore che esiste la'? Ancora bisogno di supporto …
  - E cosi' via!
- Cloud: la sottomissione non e' di uno script, ma di una macchina di calcolo (completa di sistema operativo, software etc) mediante virtualizzazione delle risorse

	GRID	CLOUD
You need	**Who provides it**	**Who provides it**
A computer connected to network, with conditioning and power	**Local site staff**	**Local site staff**
An operating system “compatible” with the application	**Local site staff**, after negotiation with experiments	Comes as a virtual image from the **experiment central infrastructure**
A local installation of the experiment software (and a local area where to store it)	**Local site** staff provides area, **Experiment support** installs software	Downloaded on demand from the **experiment central infrastructure**
Machines for local experiment facilities (voboxes etc)	**Local site staff** provide them.	They are also virtual images / not needed locally
A local storage containing the input data	**Local site staff** needs to have bought storage for the experiment	Data can be accessed remotely
A configuration to be executed	**User**!	**User**!

Il supporto dello staff locale di un sito e’ importante solo per mantenere il sito acceso, non per dare supporto alle applicazioni degli utenti!

Facilemnte, questo si estende anche a siti commerciali

Confronto col resto del mondo…



In 2012: 2800 exabytes created or replicated
1 Exabyte = 1000 PB

Reputed capacity of NSA's new Utah center: 5000 ExaB
(50-100 MW)

We are big…
not NSA-big, but big
(and more cost efficient)

Il dislivello perggiorera' enormemente nel 2020

http://www.wired.com/magazine/2013/04/bigdata/

O anche semplicemente i supercomputer

TOP 10 Sites for November 2016

For more information about the sites and systems in the list, click on the links or view the complete list.

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
3	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
4	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
5	DOE/SC/LBNL/NERSC United States	**Cori** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect Cray Inc.	622,336	14,014.7	27,880.7	3,939

Questi sono I primi 5 supercomputer (noti) a Nov 2016.
Il primo, da solo, e’ 40 volte tutta la potenza di calcolo di LHC!

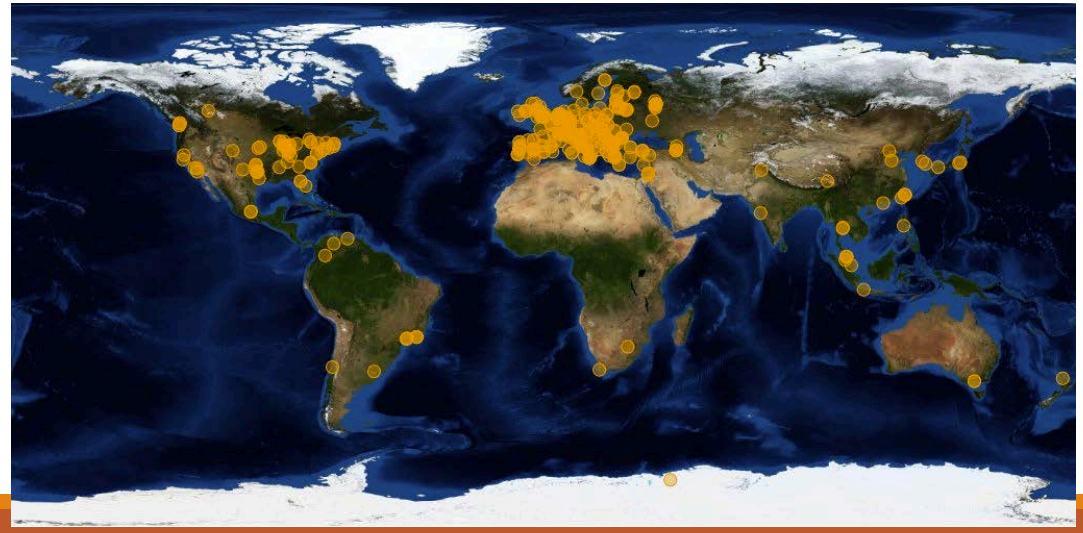
Notare la nuova entry cinese, di cui non si sapeva nulla poco prima: fattore 3x rispetto al precedente vincitore (cinese)

Americani sono un fattore 6 sotto sulla carta!

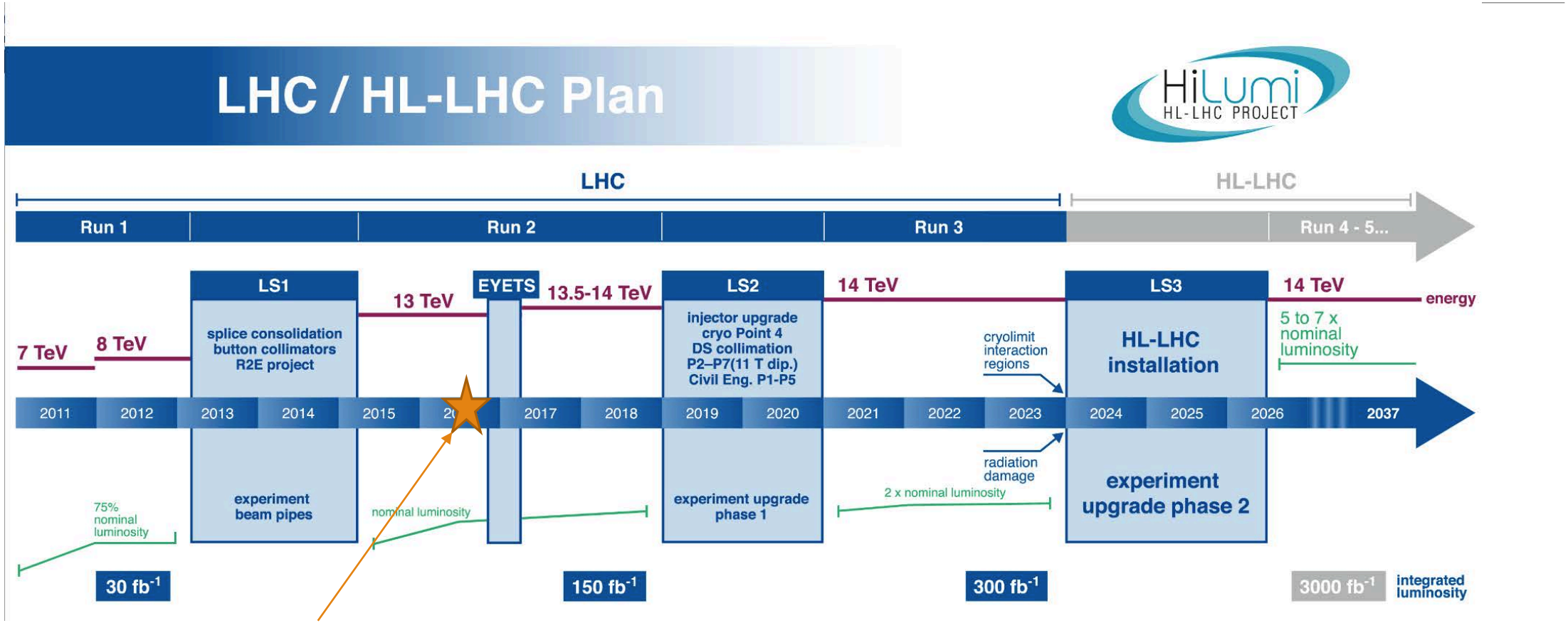
Why is LHC so relevant …

- LHC is not the biggest player in the Scientific Computing
  - A Single high level HPC size can (at least by some metrics) surpass all of it: Sunway TaihuLight, Tianhe-2, Titan, …
  - But is currently the biggest for what concerns distributed Scientific Computing on CPU architerctures
- Expected to maintain a leadership also in the next, even if some competitor may arise
  - SKA, Human Brain Project, …
  - How big is LHC Computing today?

- WLCG-2016: summing up all the experiments
  - CPU = 3.6 MHS06 (~360000 cores)
  - Disk = 310 PB
  - Tape = 380 PB
- ~200 sites registered in WLCG

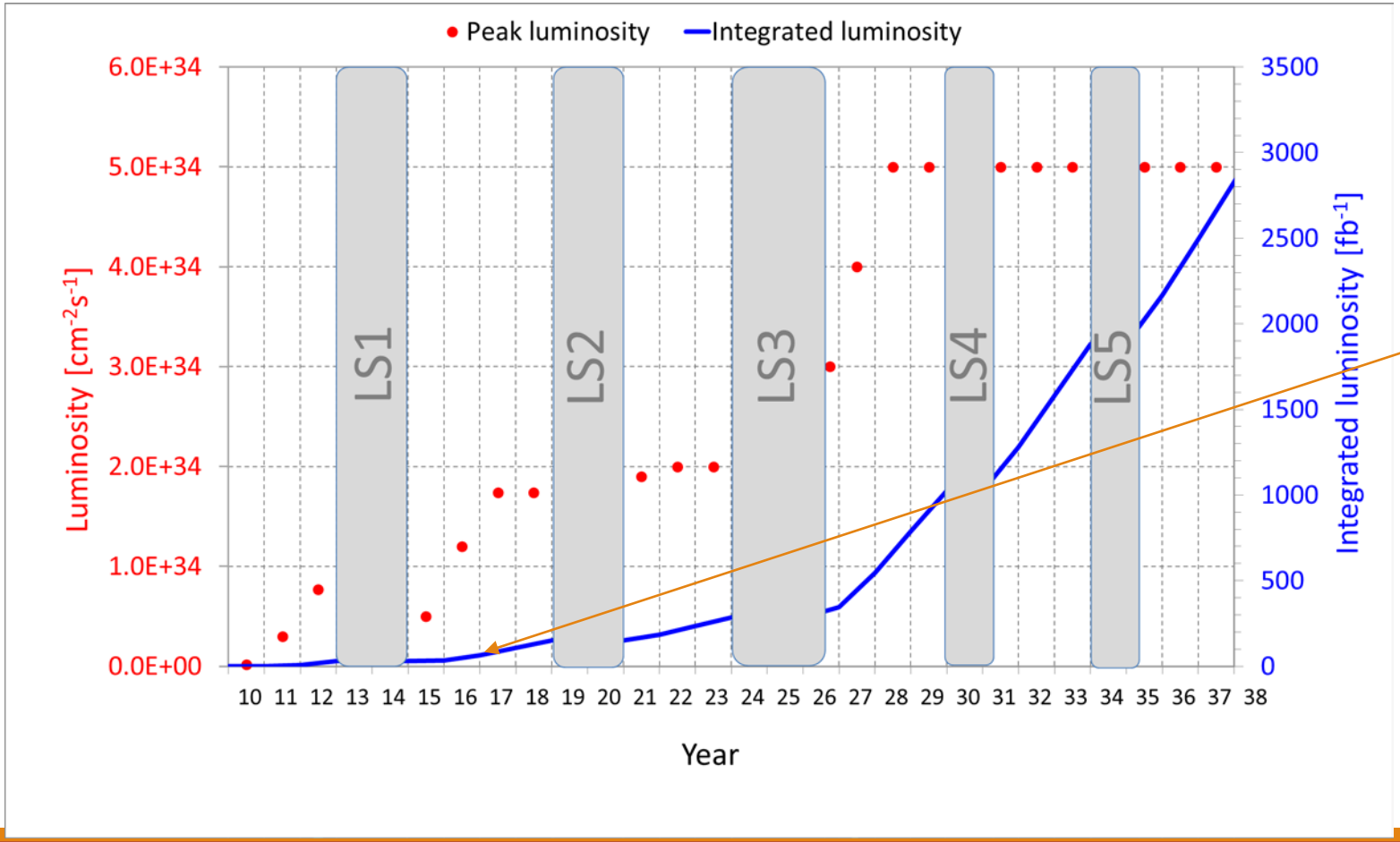


Expected Evolution



We are here

Or even …



We are here, at < 100/fb collected
Expected total yield of LHC data
3000/fb per experiment:

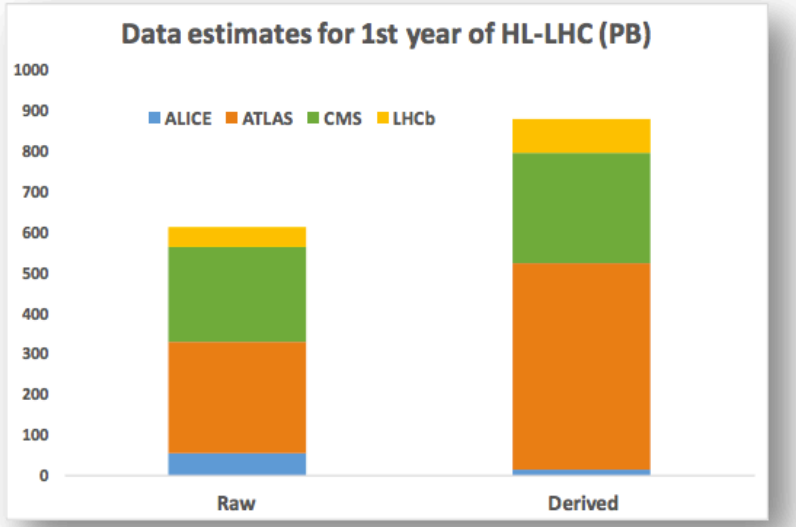
We did the first 3% of LHC!

Estimates of resource needs for HL-LHC

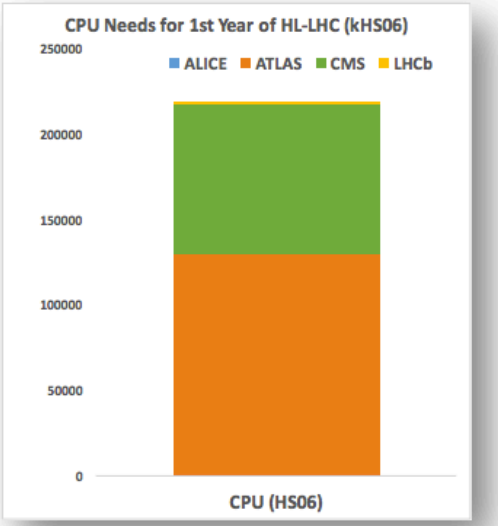
ECFA, a few days ago

Message: expected hardware evolution insufficient to close the gap with Run IV, even if 10 years are a lot

Out of the box, computing in Run IV would be 10x more expensive than today
(then, let me conclude, simply impossible …)



Storage
Raw 2016: 50 PB → 2027: 600 PB
Derived (1 copy): 2016: 80 PB → 2027: 900 PB



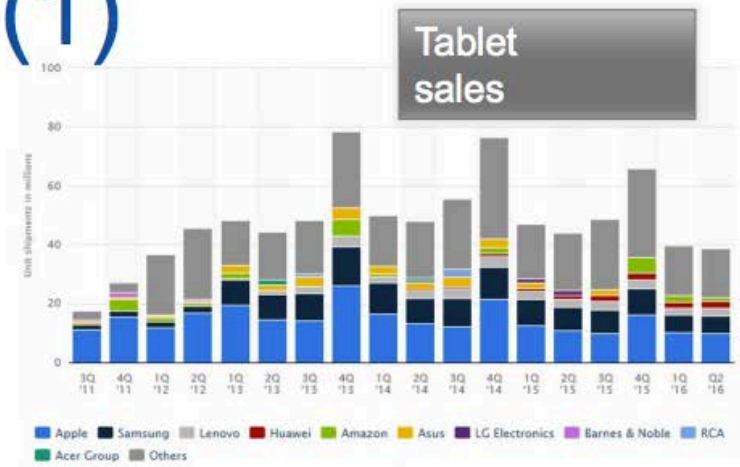
CPU
x60 from 2016

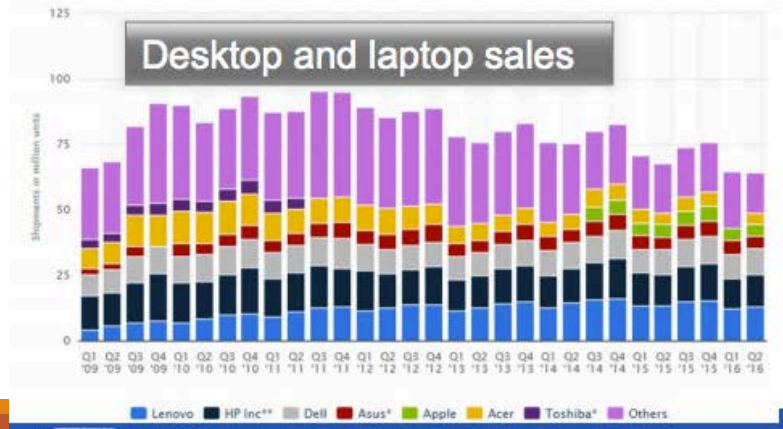
Technology at ~20%/year will bring **x6-10** in 10-11 years

=> x10 above what is realistic to expect from technology with constant cost

What is off-the-shelf? (and hence has the sweet spot for price?)

- 200X: x86 PCs
- 201X: Smartphones, Tablets
- 202X: ???
  - IOT?





Market saturation: no or negative growth rates

Smartphones	0%
Tablets	-12%
Desktops and laptops	-7%
Servers	-3%

The computing world, circa 2020 (a possible scenario)

Smart-stuff
(out of scale…)

Home/Office PCs
(out of scale…)

Commercial Clouds
(out of scale…)

HPC

LHC Owned resources for Online

LHC Owned resources for Offline

External resources!

- Commercial cloud even today outpaces LHC both as installed and as growth rate, by a lot (no one really knows how much!)
  - Amazon: 2 M servers on 2012, who knows how much today
  - M$: they claim 3x as Amazon
  - Google?
  - RackSpace?
  - …
- Number of user/office PCs: 100M/y, so let's say 1Bcores = 3000x WLCG
  - A volunteer (BOINC, etc) approach is the most common
- HPC: Tianhe-2 (the second in the list) has 3.1 Mcores (similar to the ones we use)
  - 10x the whole WLCG
- Difficult (useless?) to compare PFlops and HS06, but with a rough conversion 1 core = 10 Gflops
  - If so, the first 10 HPC centers are equivalent to 200 Pflops = 20Mcores = 80x WLCG
- Smartphones / Tablets = 500M/y, so let's say 1 Bcores = 3000x WLCG

L’esigenza nuova che la GRID non sa coprire e’ il calcolo opportunistico!

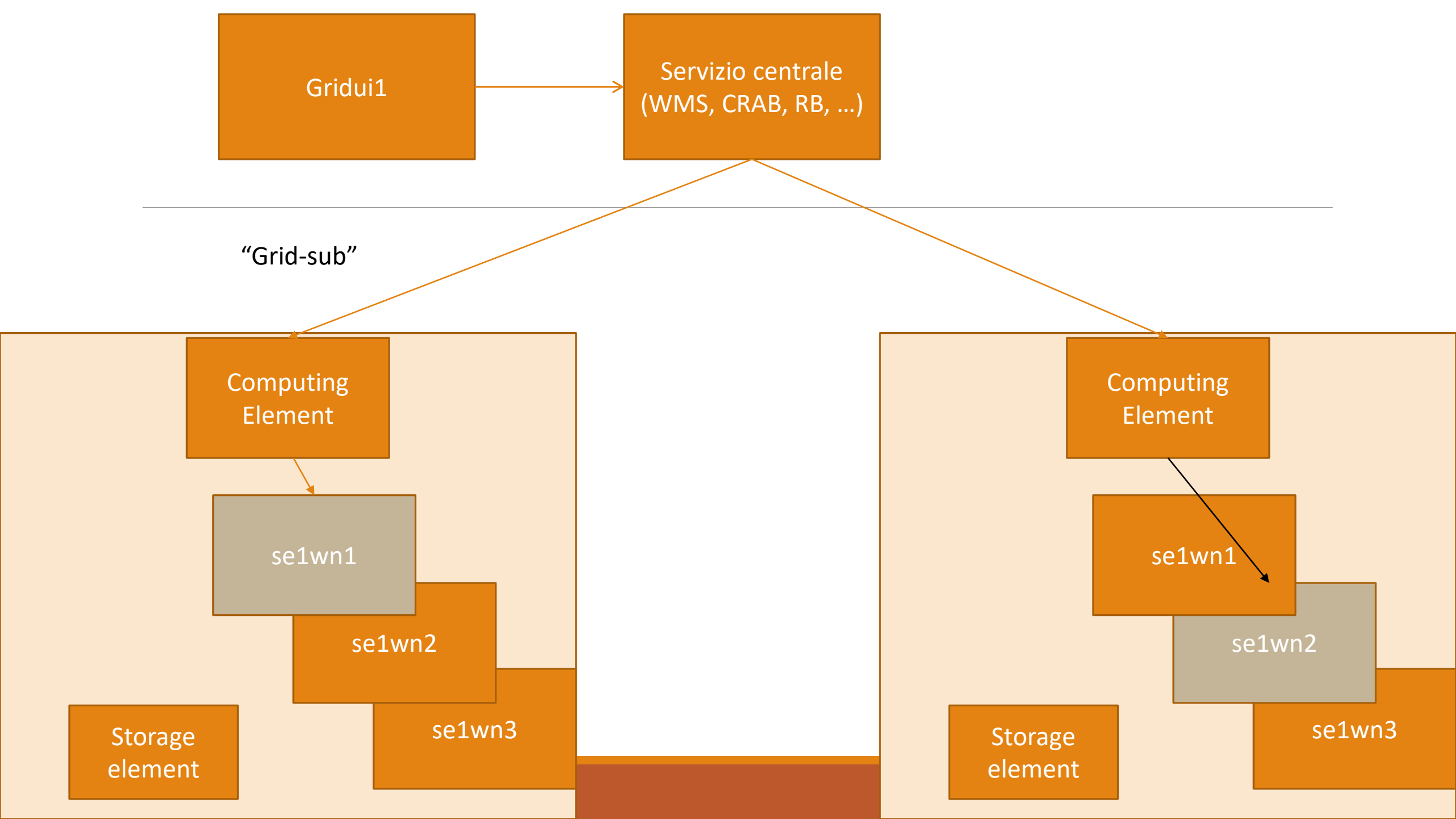
- Computer che sono disponibili per 2 o 3 ore, e poi scompaiono
- Google Cloud che fa un’offerta per 3 giorni al 90% di sconto, pero’ va usata ORA
- Un milione di ragazzini che di notte lasciano acceso il loro computer con BOINC acceso

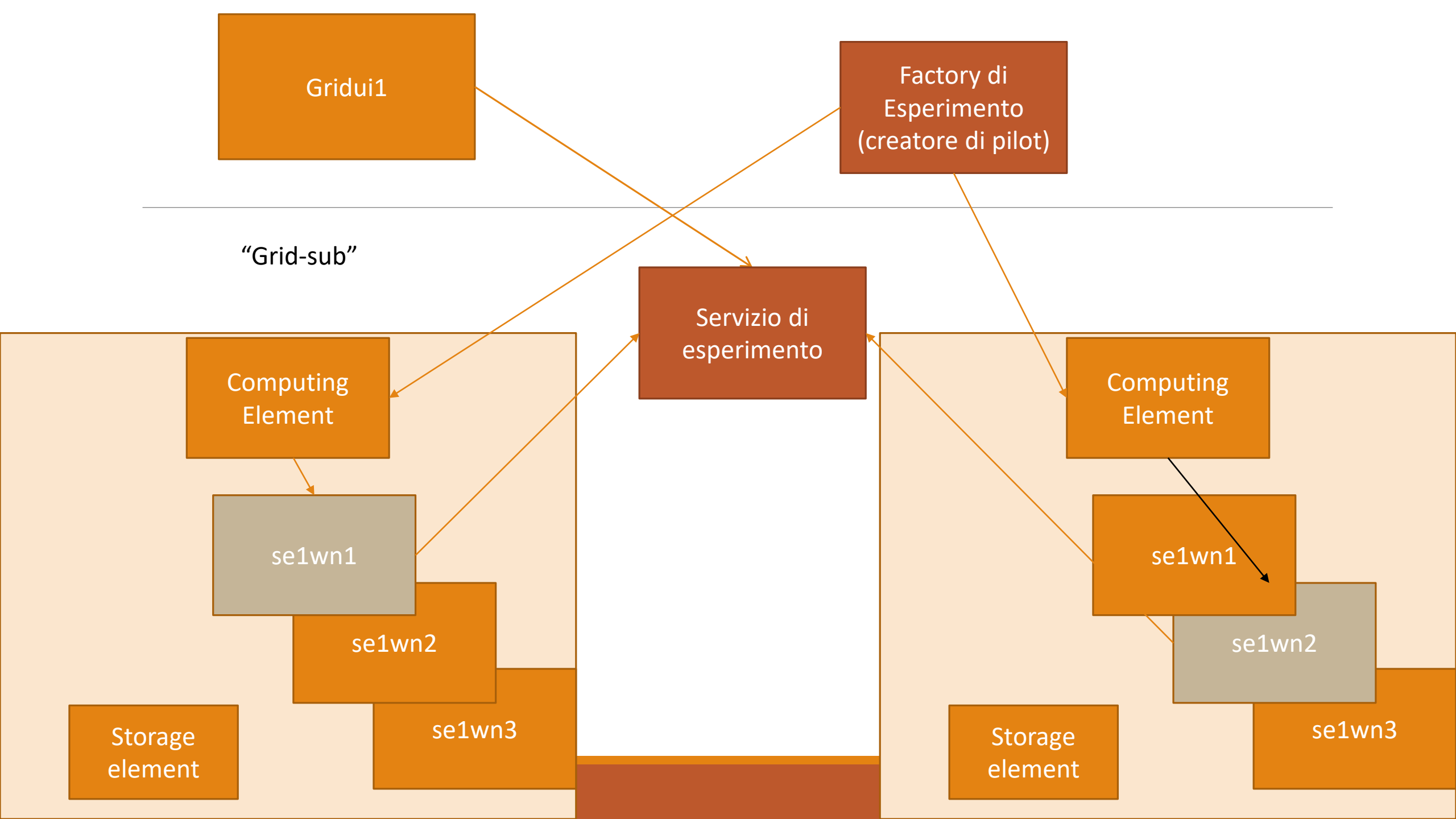
- Pensate alla GRID come
  - “ho comprato delle risorse, le uso per I prossimi 4 anni a tempo pieno”
- vs la Cloud
  - “ora compro 3 ore di calcolo con la carta di credito; domani si vedra’”

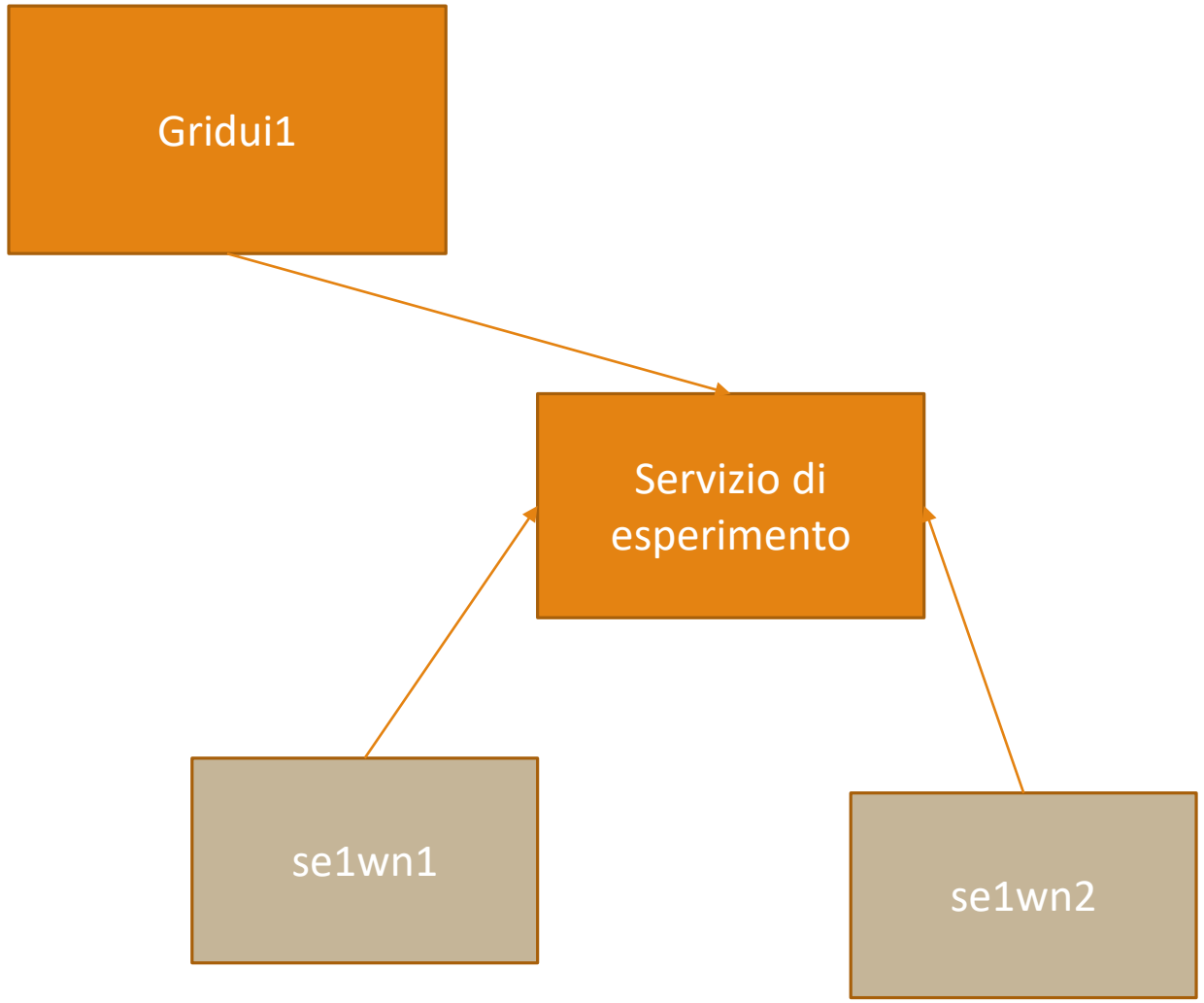
Altra evoluzione: batch system model versus squatting

- Modello standard: sottometto dei jobs, quando sta a me vanno in esecuzione
- Vantaggi:
  - una CPU viene impegnata da me solo se ho effettivamente lavoro da fare
  - Il sistema e' "fair" fra I vari utenti (a meno di bugs e problemi)
- Svantaggi:
  - Se ho fretta, mando I jobs ora e chissa' quando andranno in esecuzione. D'altra parte non potevo mandarli prima perche' non sapevo cosa avrei dovuto fare
  - Se io non mando nulla in coda, e se qualcun'altro manda tantissima roba, quando io sottometto anche se ho priorita' alta dovro' comunque aspettare che I suoi jobs finiscano

- Modello a Pilot
  - Creare un batch system nel batch system
  - In pratica (poi spiego meglio con un esempio), io non sottometto alla GRID un job di calcolo scientifico, ma sottometto in processo che io controllo da remoto
  - "sto creando una overlay network sulla GRID, di cui sono il padrone"


Gridui1
Servizio centrale
(WMS, CRAB, RB, …)
“Grid-sub”
Computing
Element
se1wn1
se1wn2
se1wn3
Storage
element
Computing
Element
se1wn1
se1wn2
se1wn3
Storage
element


Gridui1
Factory di Esperimento (creatore di pilot)
"Grid-sub"
Servizio di esperimento
Computing Element
Computing Element
se1wn1
se1wn2
se1wn3
se1wn1
se1wn2
se1wn3
Storage element
Storage element



Adesso ho creato un **cluster virtuale**, prendendo nodi da tutte le parti dove ho potuto (dove I miei pilots giravano).

Su questo metto un mio batch system, e decido che jobs farci partire.

Vantaggi:

- Se devo far partire dei jobs ORA, non devo aspettare che la mia priorita' sulla GRID sia tale da farli partire, se ho gia' requisito delle macchine
- Se ho 10000 jobs in coda su GRID, ho qualche modo di cambiarne la priorita', ma limitato. Qui faccio tutto quello che voglio

Svantaggi:

- Se non ho nulla da fare, requisisco macchine per nulla (se sono bravo le restituisco in fretta, ma mica e' detto)
- In pratica, e' un approccio molto egoistico al calcolo: arraffo tutto quello che posso,e me lo tengo fino a che e' possibile.
- (in pratica, per ME nessun svantaggio, per gli altri si'…)

Approccio Pilot

- Nella pratica, crea davvero un LSF di centinaia di migliaia di nodi, che appaiono e scompaiono quando il pilot muore / finisce il suo tempo a disposizione
- Su tale LSF, sono il padrone completo: faccio quello che voglio

- Approccio Pilot++: non requisisco CPU, ma direttamente nodi di calcolo interi!
  - Per esempio mandando pilot che richiedono TUTTE le CPU di un certo computer

Cosi’ facendo posso prendere decisioni ottimizzate

- Se so di avere 24 CPU/cores e 48 GB di RAM:
  - Posso utilizzarli dicendo al MIO batch system di mandarci 8 jobs da 2 cores + 1 da 8 cores
  - Se mando 8 jobs da 1 GB di RAM l’uno, so che c’e’ spazio poi per mandarne 8 da 5 GB l’uno
  - Se so che ho jobs che utilizzano la CPU al 100%, e altri che la utilizzano al 50%, posso mandare 12 jobs del primo tipo e 24 del secondo

- Insomma, l’approccio a nodo completo permette di utilizzare al meglio le risorse esistenti, ma e’ ancora piu’ egoistico: requisisco non piu’ un job slot, ma una macchina intera …

Storage – evoluzione

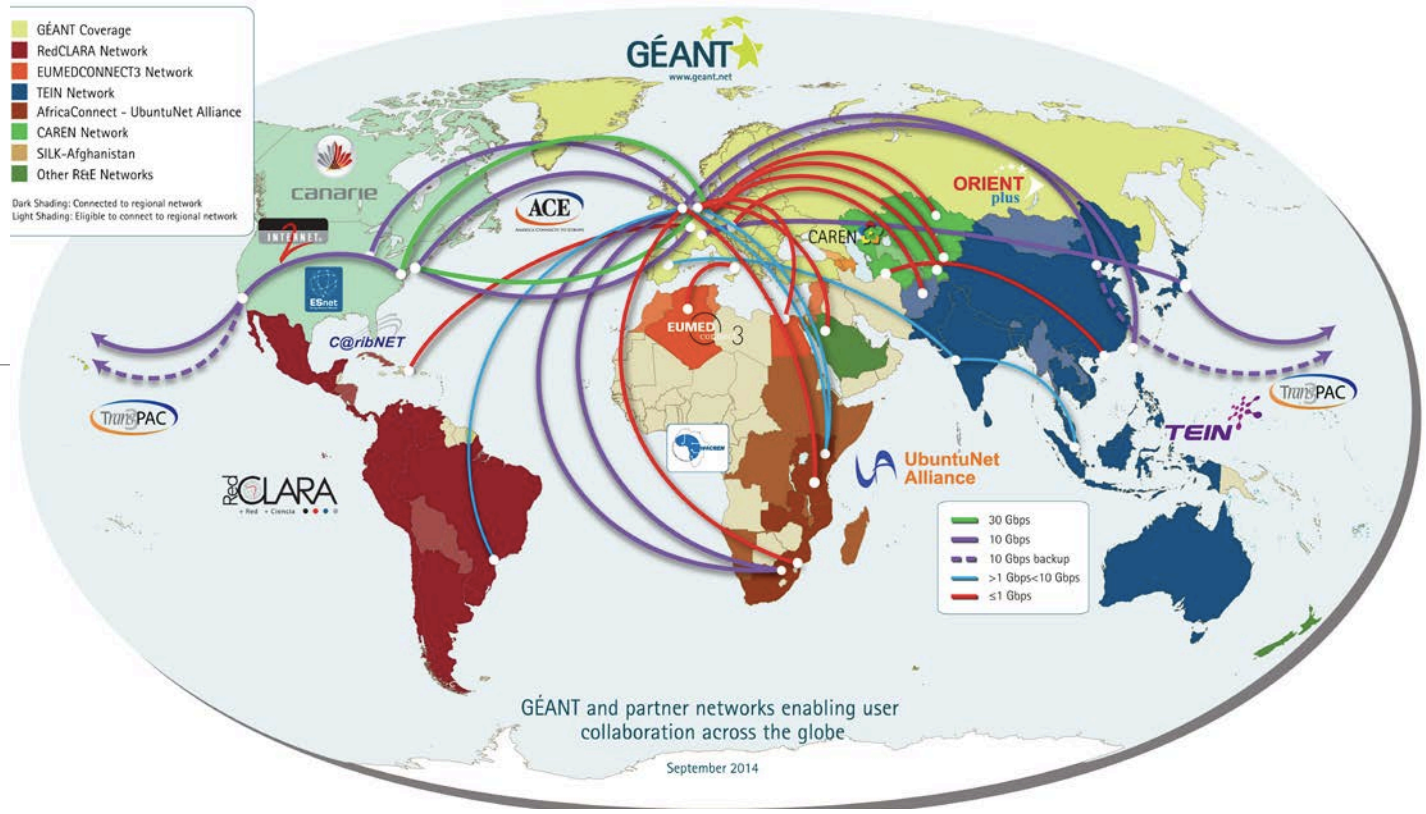
- I comandi GRID sono fatti per spostare in modo abbastanza efficiente files fra storage diversi. Questo e' tutto.
- Questo implica parecchie cose:
  - Visto che spostare I files e' costoso in termini di rete (per esempio), I jobs vengono tipicamente mandati dove sono I files – abbiamo una DataGRID
  - Se voglio mandare un job in un sito che non ha il file
    - O lo metto nella input sandbox
    - O il job stesso se lo copia localmente prima di cominciare
  - … perche' non va bene???

Cloud etc

- Sulla Cloud, e su Seti@HOME equivalenti, c'e' gente che vende / regala CPU
  - Lo storage se in vendita costa molto di piu'
  - Lo storage se in regalo e' sul PC di una persona, che non vuole si utilizzi tutta la sua ADSL per prendere un file di cui magari poi leggo solo I primi 10 MB
- Attivita' commerciali possono offrire in regalo delle risorse CPU (di notte, per esempio), ma di certo non storage
- … serve poter almeno in alcuni casi lavorare con centri diskless!

Evoluzione della rete !

- In questa valutazione entra anche il fatto che le reti geografiche sono stato l'aspetto tecnologico che ha avuto la maggiore evoluzione nell'ultimo decennio
- L'accesso remoto a files (**Netflix**, etc…) ha adesso una capacita' non troppo differente dall'accesso locale
  - Chiaramente la velocita' della luce nelle fibre ottiche e' un limite, e genera latenza, ma c'e' una soluzione …



Sito tipico italiano 10-20 Gbit/s – 100 entro un anno
Sito tipico US 100 Gbit/s – 400 entro l'anno
Connettivita' US-EU: 340 Gbit/s (solo per la ricerca!)

Accesso remoto!

- E’ un modo per reintrodurre in modo soft il vecchio concetto di GRID di **FileSystem unico nel mondo (tutti I files a disposizione di tutti da qualunque posto)**
- I job possono girare dovunque ci sia **CPU libera**, e accederanno remotamente ai files
- Possibilita’ di ottimizzazione della sorgente da utilizzare nel caso il file sia presente in piu’ posizioni
  - Posizione migliore (piu’ veloce)
  - Posizione con rete meno intasata
  - Tutte le posizioni insieme
- Accesso ottimizzato per files ROOT nel caso vengano letti solamente TBranches specifici
  - Per esempio solo muoni e elettroni, mai I jets in un file ricostruito
  - Se questo succede per I primi 1000 eventi e’ improbabile che vengano letti I jets dopo:
    - TTreeCache: tool di ROOT che “impara” nei primi eventi l’access pattern e precarica i dati, in modo da non soffrire per problemi di latenza di rete.
  - Tools che oggi sanno fare questo: **Federazione di Xrootd**, Federazione di Https, OneData
    - Pisa nella federazione Xrootd di CMS

Il Tier2 di CMS

- E’ stato la forza trainante dell’evoluzione del Centro di Calcolo di Pisa almeno dal 2000 al 2012

- Adesso e’ tuttora di gran lunga la maggiore installazione di storage di Pisa (> 1 PB di disco usabile via GPFS); per quanto riguarda le CPU non e’ piu’ vero (cluster teorico almeno comparabile)

Facilities presenti a Pisa

- Storage GPFS visibile via POSIX, SRM, XrootD
  - Quello del Tier2, ~1.2 PB
  - /gpfs/ddn/srm/cms/store
- Storage “scratch” su GPFS, visibile via POSIX
  - ~30 TB
  - /gpfs/ddn/cms/user
  - Una UserInterface esclusiva per analisi interattiva
    - **Cmsanalysis.pi.infn.it**: 64 cores, 512 GB di RAM

Mi fermo qua

- Purtroppo un vero e proprio corso hands-on non ha senso visto che che I modelli operativi dei vari esperimenti non sono piu' allineati sul semplice modello GRID
  - Che pero' vale ancora per piccoli ricercatori "isolati", o gruppi di ricerca che non vogliano investire nella costruzione di tools complessi
- Spero comunque che la panoramica sulle direzioni che il calcolo sta assumendo nell'INFN (e nel calcolo scientifico in generale) serva a rendere il singolo ricercatori consapevole della quantita' di lavoro esistente dietro la richiesta di accesso ad una singola CPU o ad un singolo file!