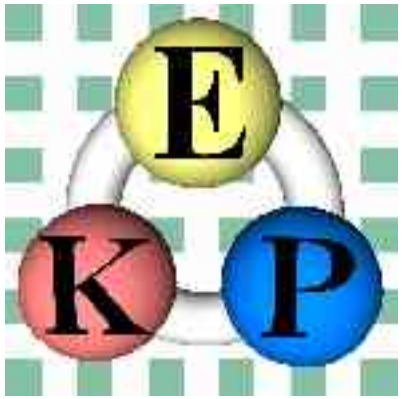
Freie Software und Linux in der Hochenergiephysik

Levin Jungermann, Yves Kemp
Thorsten Scheidle, Marcel Stanitzki



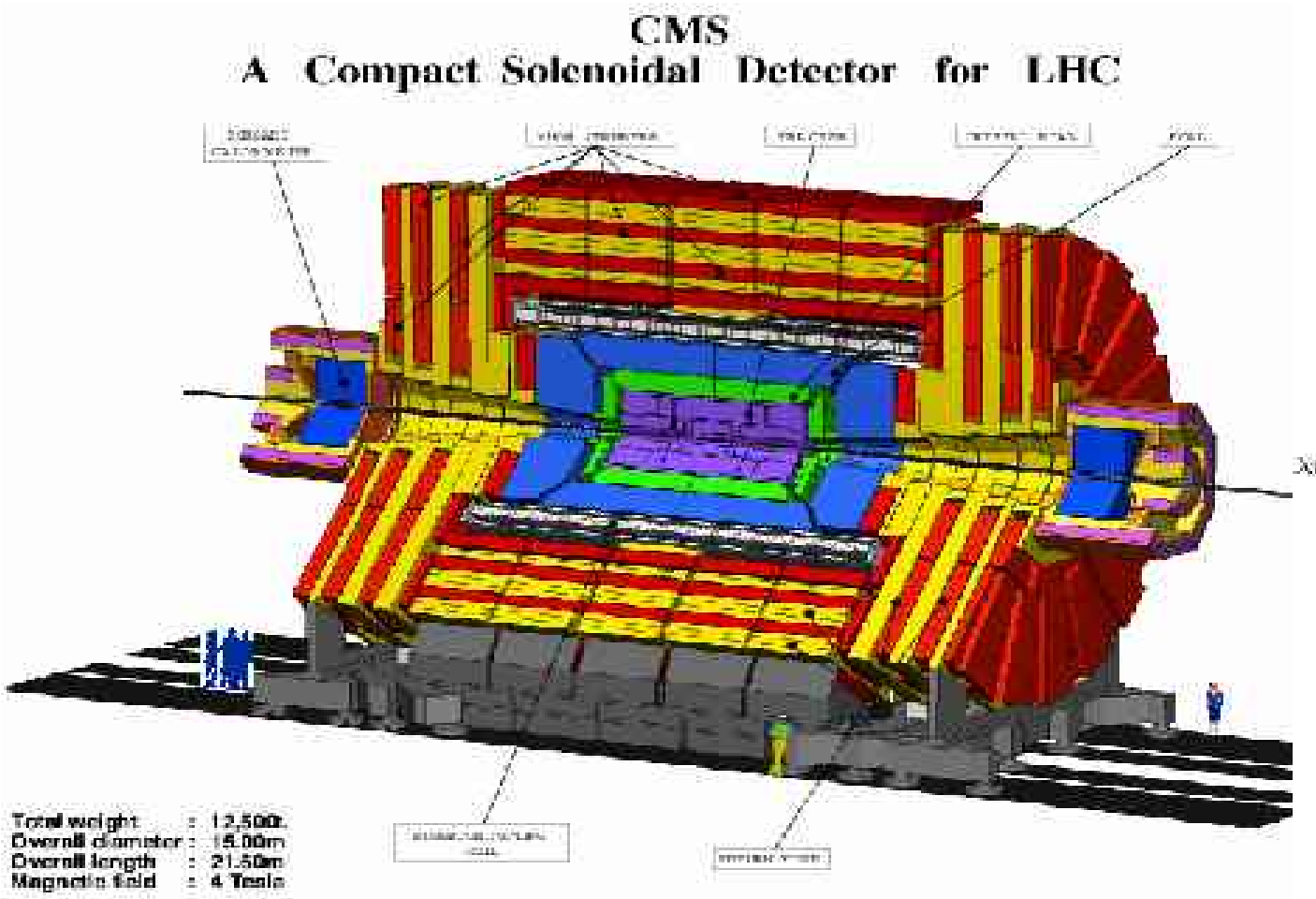
Universität Karlsruhe (TH)
Institut für Experimentelle Kernphysik

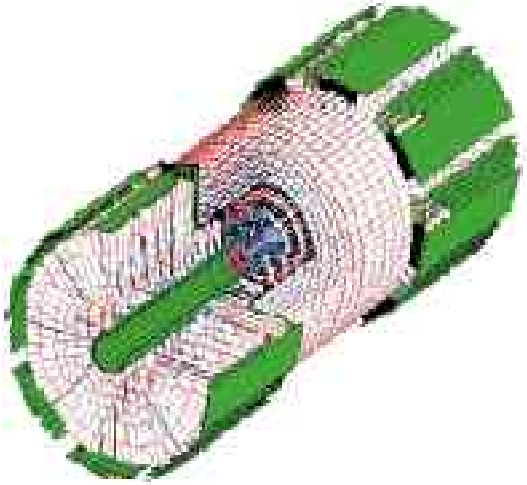
Übersicht

1) Was ist HochEnergiePhysik (HEP)?
   Wie sieht unsere Community aus?
2) Einsatz von Linux in der HEP
3) Einsatz von OpenSource in der HEP
4) Exkurs: GRID-Demo
5) Nutzen von HEP für Linux
   & Open-Source
6) Entwicklungen am EKP

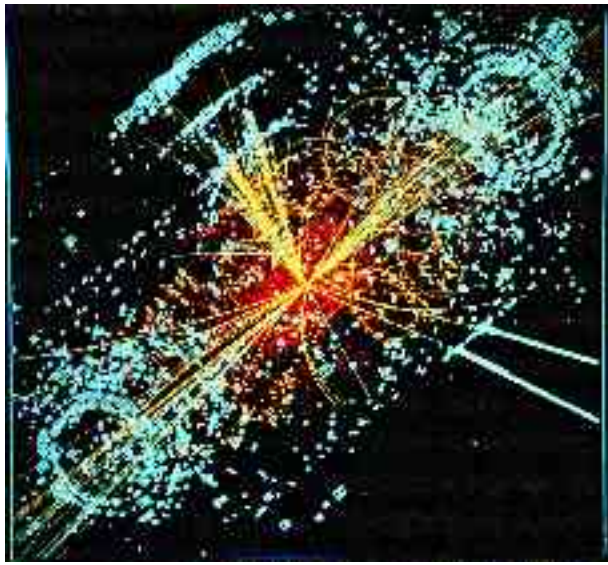
Detektoren

10 Mio Kanäle
>200m^2 Silizium





Ansicht des CMS Silizium-Spurdetektors

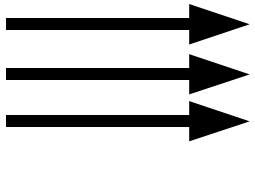


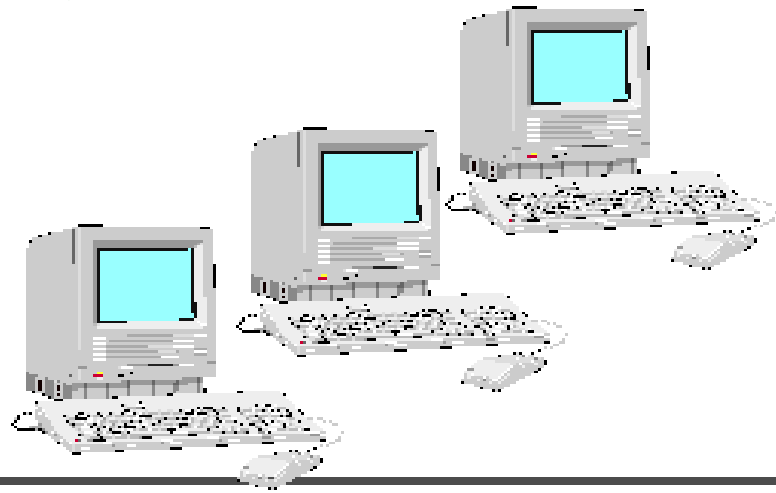
Levin Jungermann, Yves Kemp
Institut für Experimentelle Kernphysik

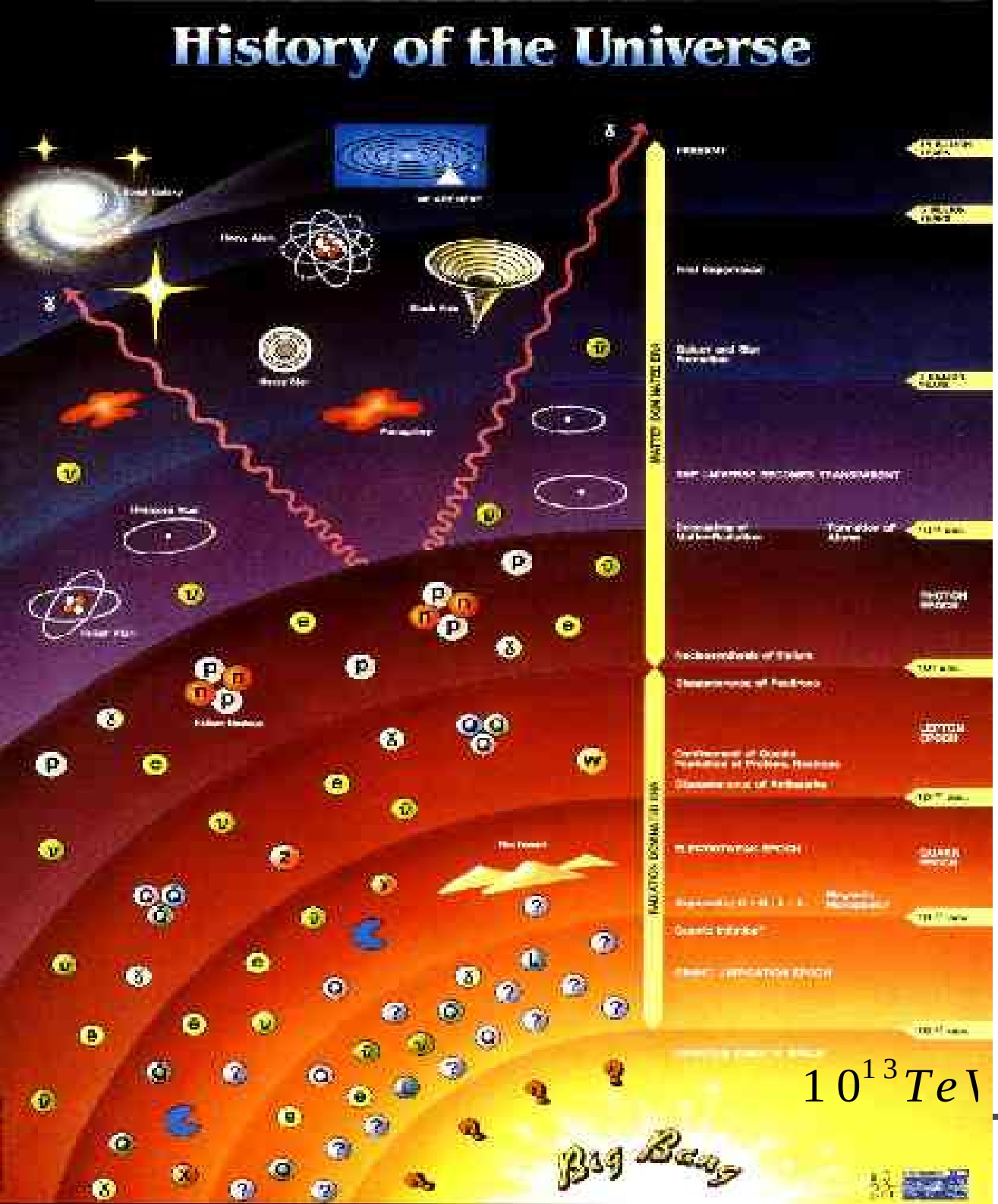
Datenrate und Datenmenge

40 Mhz; 109 Ereignisse/s in 150x106 Sensoren = **1 PetaByte/s**

Online-Datenreduktion
Rekonstruktion und Speicherung
100 Hz ~ **1 PetaByte/Jahr**







heute
0.7 meV

Was War Wann?

Atome
1 eV

Atomkerne

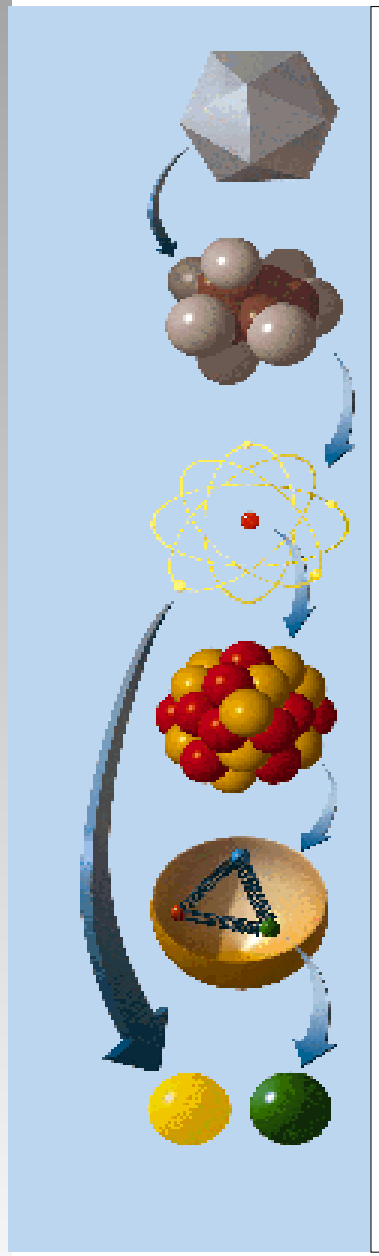
Nukleonen
0.3 MeV

nur noch 1.Generation

alle Teilchen im Gleichgewicht
1 TeV

Levin Jungermann, Yves Kemp
Institut für Experimentelle Kernphysik

Teilchenphysik (HochEnergiePhysik)



$10^{-2}\ m$	Kristall	Röntgenstrahlung
$10^{-9}\ m$	Moleküle	
$10^{-10}\ m$	Atom	Teilchenstrahlung (Labor)
$10^{-14}\ m$	Atomkern	
$10^{-15}\ m$	Proton	Teilchenstrahlung (Beschleuniger)
$<10^{-18}\ m$	Elektron Quark	

Untersuchungen der Materie durch Strahlung:

- atomos (unteilbar)
- viele Atommodelle
- Bohrsches Atommodell
- Atomkern,Elektron
- Proton,Neutron
- Quarkmodell

Teilchen wurden immer als elementar angesehen bis neue Energien tiefere Einblicke erlaubten!

Beschleuniger

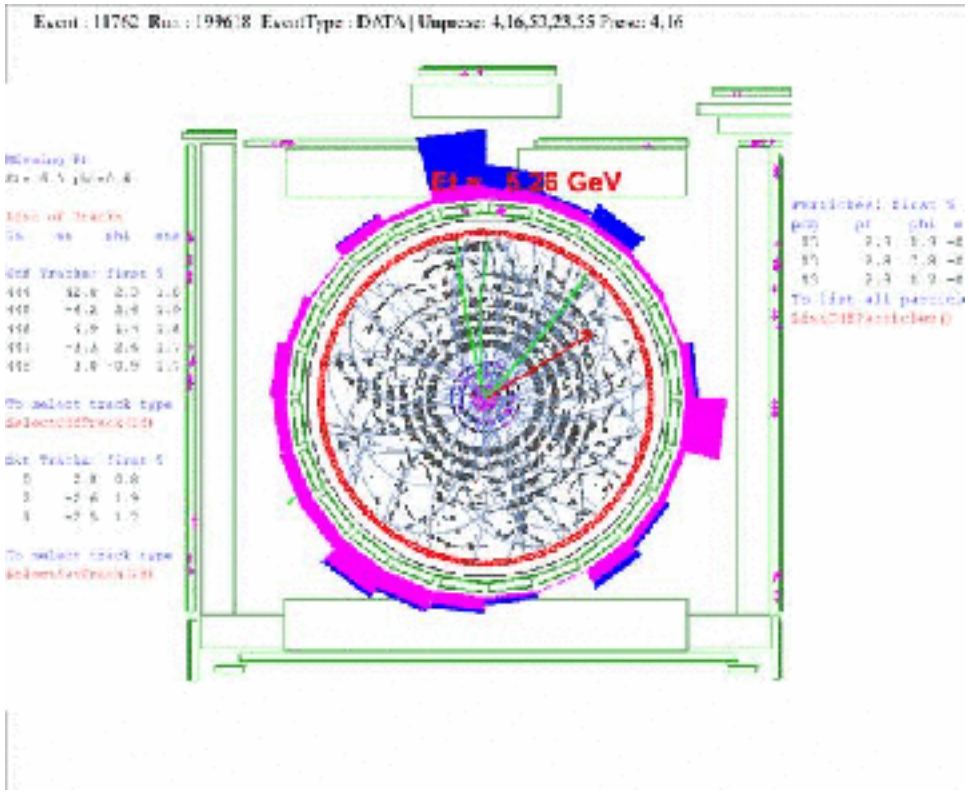
Fermilab (Chicago): TeVatron

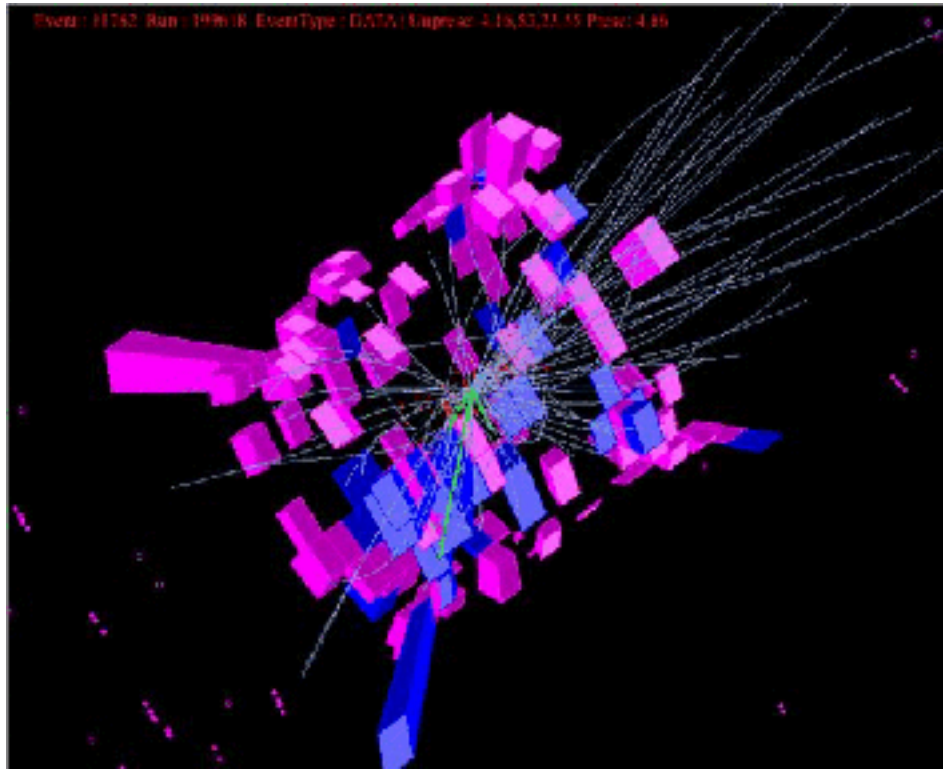
U: 6km E: 1.96 TeV (99,99995% von c)

CERN (Genf): LHC

U: 27km E: 14 TeV

LIVE-Events

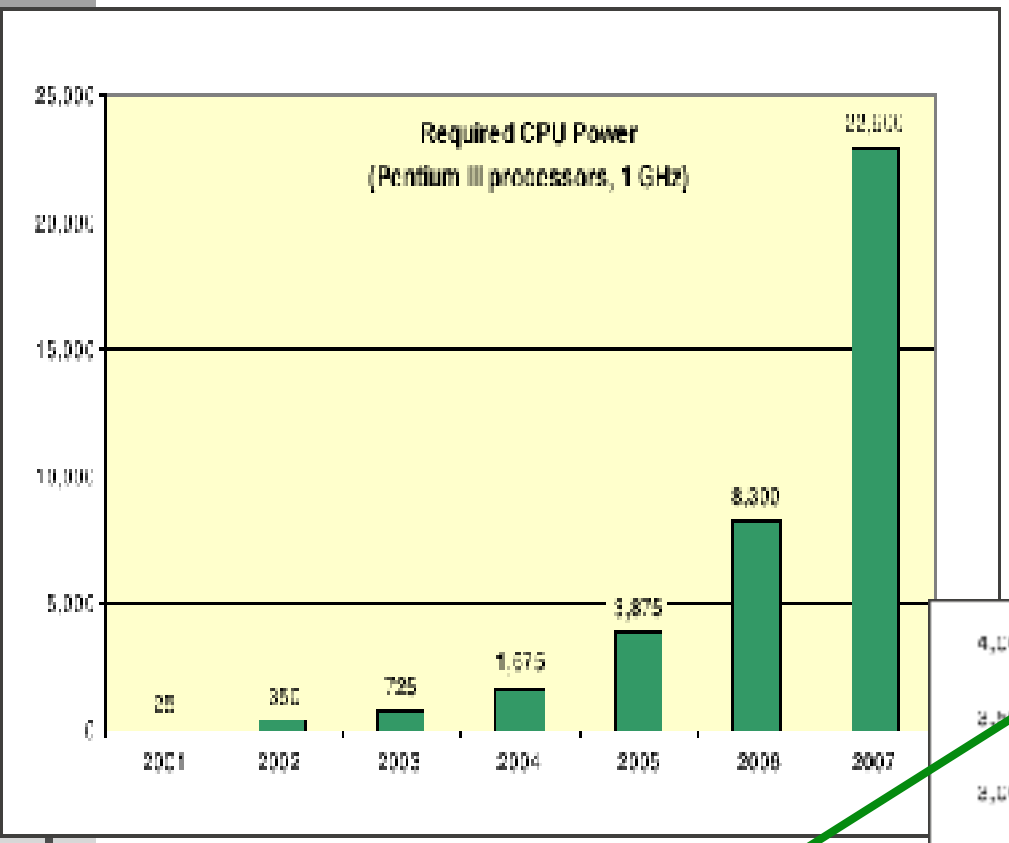


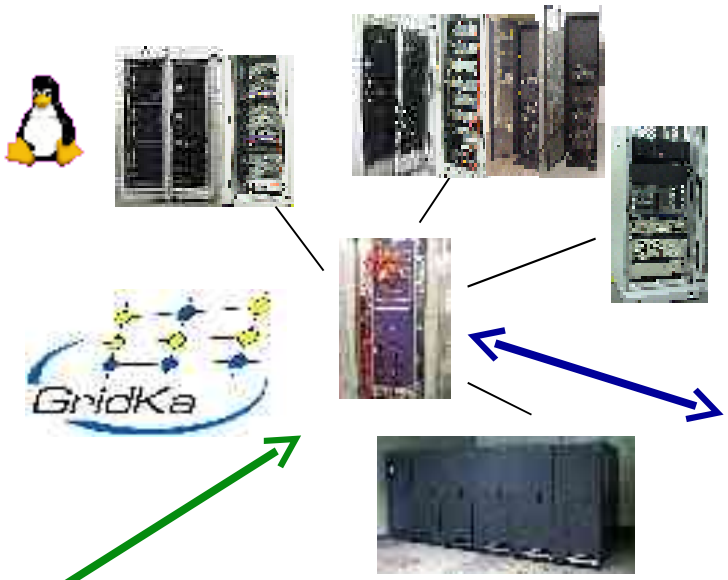


http://www.fnal.gov/pub/now/live_events/cdf_live.html

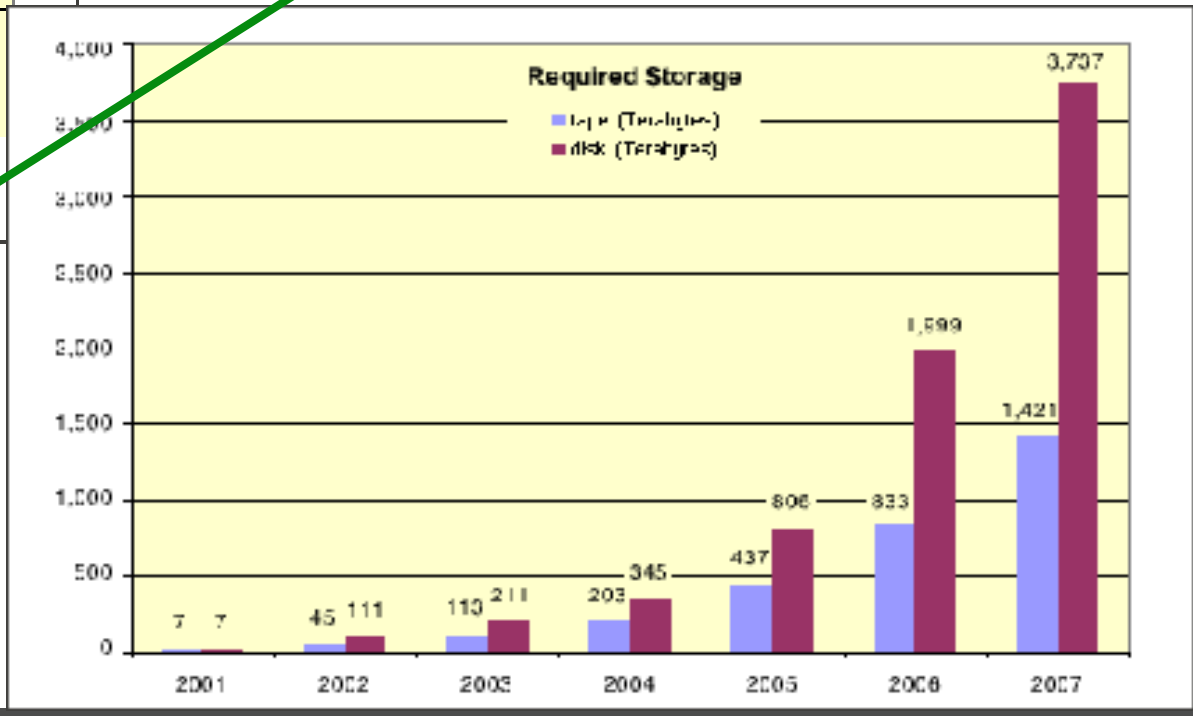
“Alte” Technologie (Design ~1990): 20 MB/s

Erwartete Computerleistung

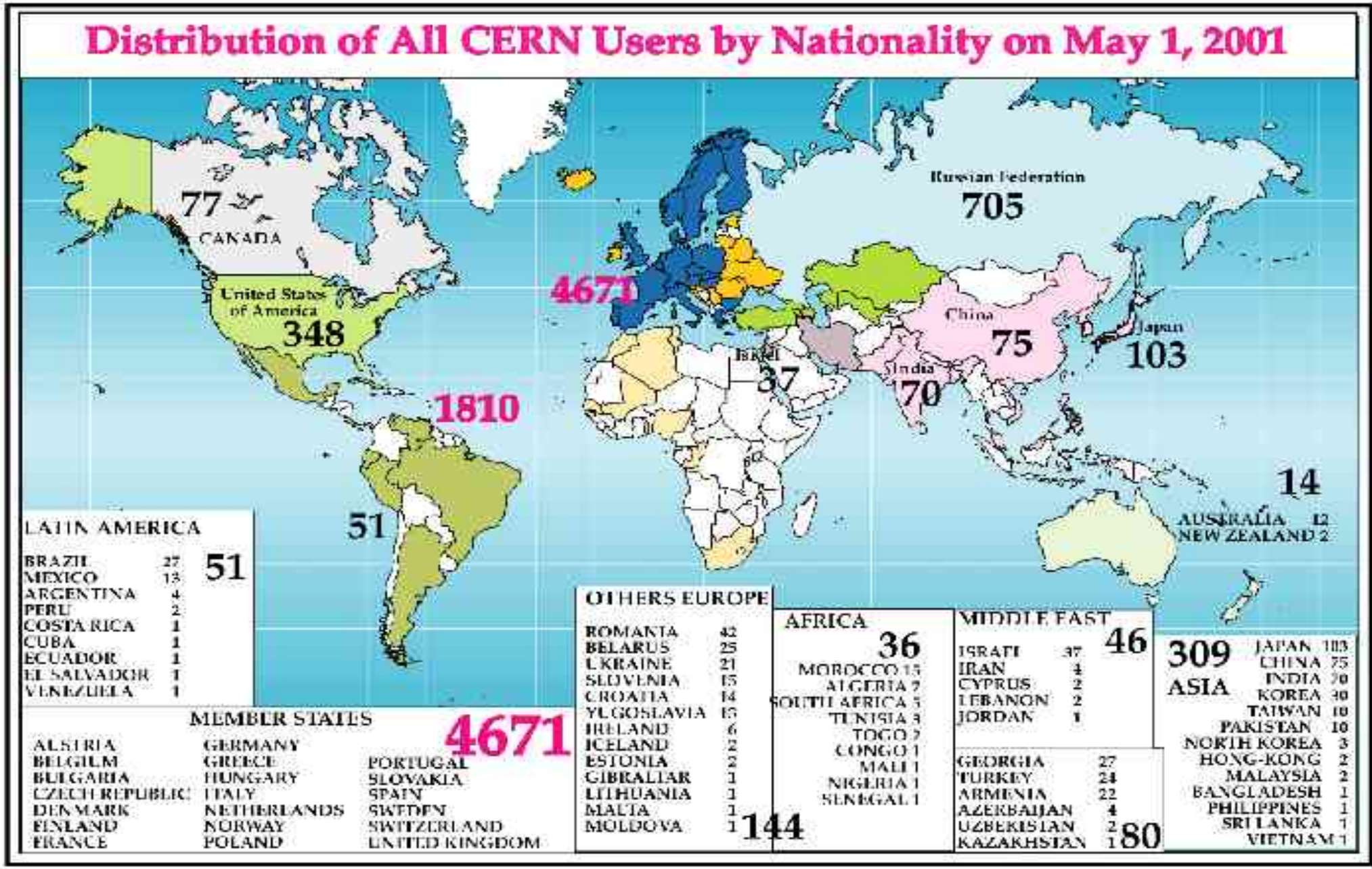




IEKP

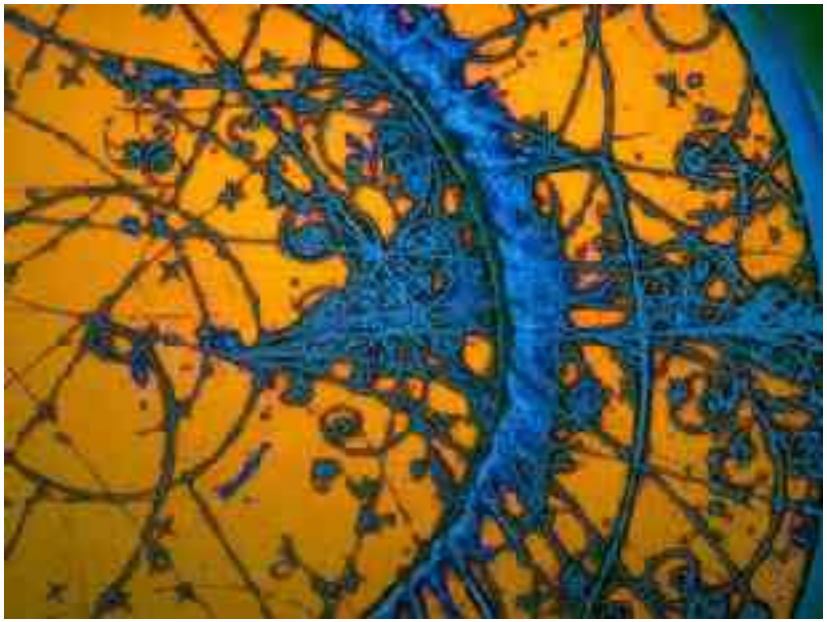


Community


Distribution of All CERN Users by Nationality on May 1, 2001
77
CANADA
United States of America
348
1810
51
4671
Russian Federation
705
China
75
India
70
Japan
103
37
14
AUSTRALIA 12
NEW ZEALAND 2
LATIN AMERICA
BRAZIL 27
MEXICO 13
ARGENTINA 4
PERU 2
COSTA RICA 1
CUBA 1
ECUADOR 1
EL SALVADOR 1
VENEZUELA 1
51
OTHERS EUROPE
ROMANIA 42
BELARUS 25
UKRAINE 21
SLOVENIA 15
CROATIA 14
YUGOSLAVIA 13
IRELAND 6
ICELAND 2
ESTONIA 2
GIBRALTAR 1
LITHUANIA 1
MALTA 1
MOLDOVA 1
144
AFRICA
36
MOROCCO 15
ALGERIA 7
SOUTH AFRICA 5
TUNISIA 3
TOGO 2
CONGO 1
MALI 1
NIGERIA 1
SENEGAL 1
MIDDLE EAST
46
ISRAEL 37
IRAN 4
CYPRUS 2
LEBANON 2
JORDAN 1
GEORGIA 27
TURKEY 24
ARMENIA 22
AZERBAIJAN 4
UZBEKISTAN 2
KAZAKHSTAN 1
80
309
ASIA
JAPAN 103
CHINA 75
INDIA 70
KOREA 40
TAIWAN 10
PAKISTAN 10
NORTH KOREA 3
HONG-KONG 2
MALAYSIA 2
BANGLADESH 1
PHILIPPINES 1
SRI LANKA 1
VIETNAM 1
MEMBER STATES
4671
AUSTRIA
BELGIUM
BULGARIA
CZECH REPUBLIC
DENMARK
FINLAND
FRANCE
GERMANY
GREECE
HUNGARY
ITALY
NETHERLANDS
NORWAY
POLAND
PORTUGAL
SLOVAKIA
SPAIN
SWEDEN
SWITZERLAND
UNITED KINGDOM

2) Linux in der HEP

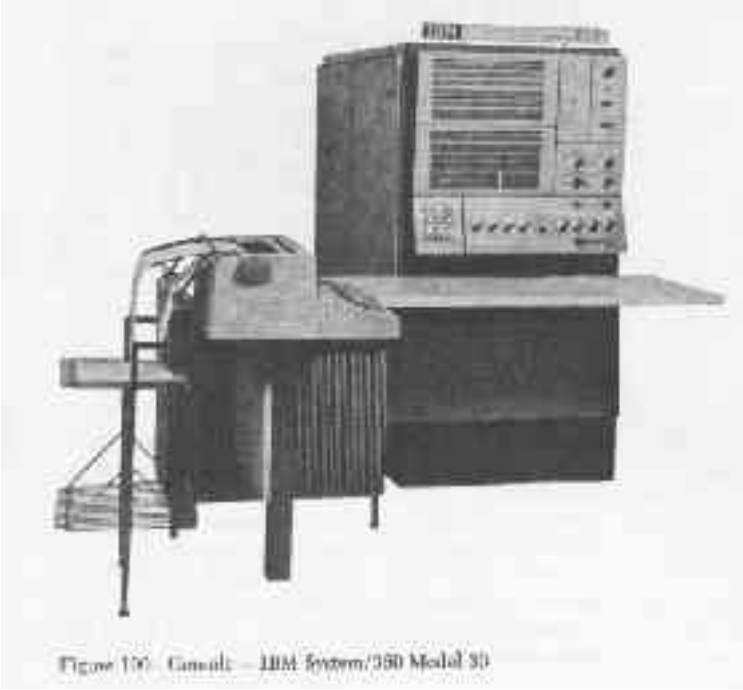
- Geschichtliche Evolution der Betriebsysteme
  - So ähnlich wahrscheinlich auch in anderen Bereichen
  - Immer an die Hardware gebunden
  - Auch an Fortschritt in der Datennahme
- Blasenkammer: “Manuell-Optische” Auswertung (bis in die 1970er Jahre)



70er Jahre: Vielfalt

PDP-12

IBM-360



CDC 6600

- Batchbetrieb
- Fortran-Programme
- Experimente:
  - Auslese z.B. auf 35mm Film
  - Digitalisierung, aber Stich-proben optische Scans, teils am Bildschirm
  - Datenaustausch: Magnetbän-der

80er Jahre: IBM und VAX

- IBM Mainframes sehr verbreitet
  - hauptsächlich zur Datenverarbeitung
  - IBM-OS
  - z.B. Am Forschungszentrum Karlsruhe
- weiterhin VAX (Digital Equipment)
  - hauptsächlich im Online-Bereich
  - Betriebsystem VMS
  - Einzelne Experimente bei Datennahme auf VAX gesetzt
- Datentransport (große Mengen): Noch immer Bänder

VAX 11

90er Jahre: Unix, Workstation

- Ultrix-Systeme von Digital
- UNIX-Systeme
  - DEC/Alpha
  - HP Apollo
  - ...
- Vernetzung, Datentransfer über Netz
- Einsatz von PCs unter DOS/Windows/MacOS nur vereinzelt

Ab ~1995: Linux

- Entwicklung beginnt 1991
- Erster Einsatz in HEP erst ~1995:
  - Hardware (x86) ähnlich performant wie DEC/Alpha
  - Preis/Leistung DEC/Alpha schlechter
- Damals (in Deutschland) SuSE als Distribution
- Weltweit hat sich RedHat eher durchgesetzt
- Entwicklung von Universitäten forciert, großen Zentren vorsichtig
- Weg von Fortran zu C/C++

Warum Linux in der HEP?

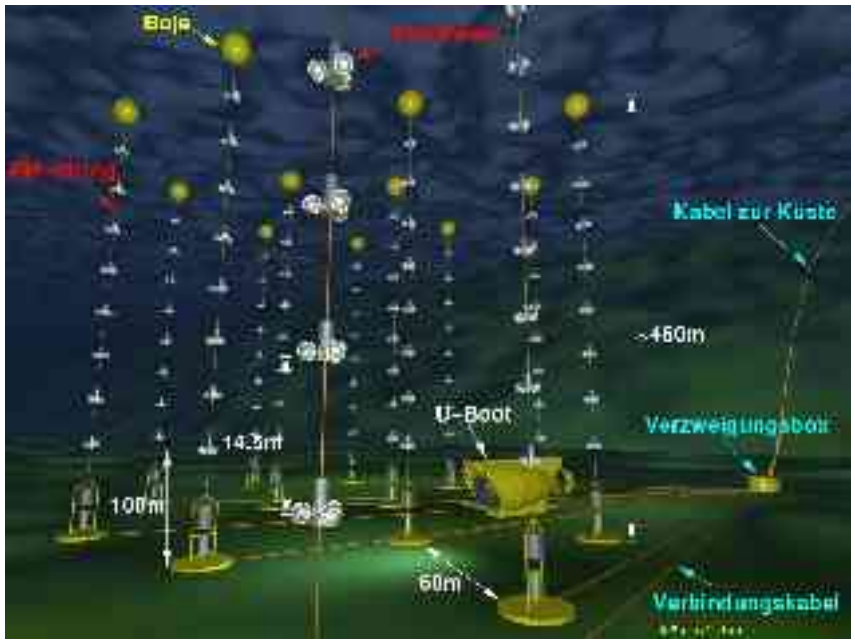
- Linux heute sehr verbreitet
- Einheitliche Plattform:
  - Notebook
  - Arbeitsplatz
  - Großrechner (Mehrprozessor-Rechner, auch nicht -x86)
  - Rechencluster
  - Datennahme (auch nicht-x86)
  - Unterschiedliche Distros: Debian, SuSE, RedHat, Scientific Linux....
- Frei verfügbar & modifizierbar
- Plattform für eigene Entwicklungen
- Netzwerkfähigkeit (im Gegensatz zu anderen Betriebsystemen für x86)

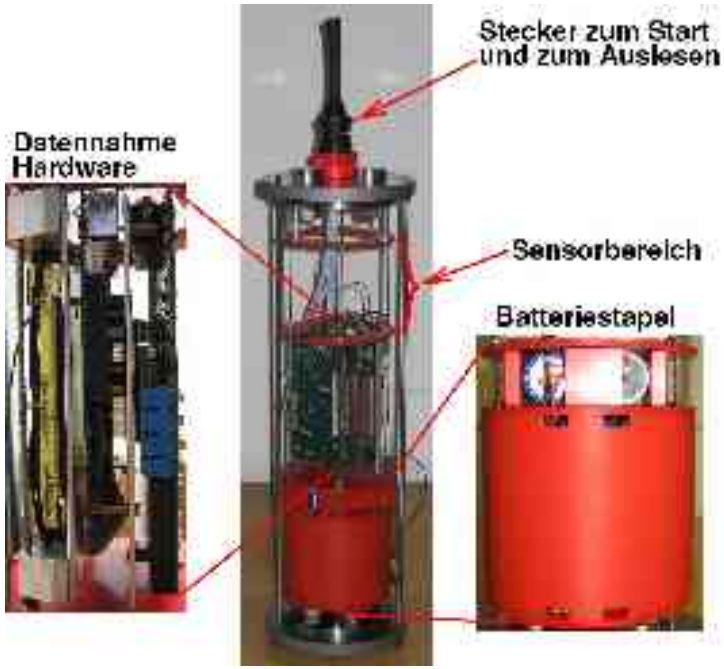
Customizing: AMS

- AMS: Experiment an ISS angedockt (2008)
- Herausforderungen für Datennahmesystem:
  - mechanische und elektronische Stabilität
  - Leistungsaufnahme &Kühlung
- Board entwickelt auf Basis von PPC
- Betriebsystem:
  - Modifiziertes Linux

Customizing 2: Antares

- Antares: Vor französischer Küste in 2500 m Tiefe
- Gruppe Uni Erlangen:
  - Autonomes Testsystem zur Messung des akustischen Untergrunds in der Tiefsee
- Herausforderungen:
  - Kleines aber billiges System
  - Absolut rauschfrei während der Messung
  - große Datenmengen
- Router-Board mit Geode
- Linuxsystem auf Flash-Speicher, fährt Festplatte zum Speichern kurz hoch





Clusterbau

- Aus verteiltem Rechnen im Desktopcluster entstanden
- Cluster unterschiedlicher Grösse:
  - EKP: ~30 CPU, 15 TB
  - L3-Farm CDF: ~400 CPU
  - FZK: ~1300 CPU, ~250 TB
- x86(-64) Plattform mit Linux ideal
- Linux/OpenSource-Tools zur
  - Installation: Kickstart, Quattor, ...
  - Monitoring: Ganglia...
  - und zahlreiche Bordmittel von Linux-Distros
- Skalierbarkeit

3) OpenSource in der HEP

- OpenSource in HEP kann in drei Klassen eingeteilt werden
  - Allgemeine Software (nicht HEP spezifisch)
  - HEP-spezfische Software
  - Experiment-spezifische Software

Allgemeine Anwendungen

In der HEP gibt es die “klassischen” Open-Source Anwendungen
- Linux auf Servern/Workstations
- KDE/GNOME based Desktops
- GCC & Co Development
- LaTeX, ...

- Kein fundamentaler Unterschied zu anderen Bereichen
- Einsatz von Linux als Scientific Desktop populär

HEP-Spezifische Anwendungen

- Dies sind Anwendungen, die für das gesamte Feld interessant sind:
  - generische Event-Simulationen (Monte-Carlos)
  - HEP-Spezifische Libraries
  - HEP Programm-Pakete
- Die meisten Pakete sind im SourceCode erhältlich

Open Source: Generelles

Verschiedene Gründe sprechen für Open Source:

- Reproduzierbarkeit: Open Source macht Ergebnisse leichter nachprüfbar
- Öffentlich finanzierte Forschung: Ergebnisse müssen frei zugänglich sein
- Anpassbarkeit: Transfer von Software-Know how ist einfacher
- Gewinnstreben: Ist in diesem Feld unterdurchschnitt-lich, kein Interessen an Patenten Copyrights, etc.

HEP Lizenzen

- Ein Großteil Software ist “as-is” (Autoren werden als Anerkennung zitiert)
- Einige Pakete unter GPL oder GPL-ähnlichen Lizenzen
- Experiment-Lizenzen: Prinzipell zum Einsatz innerhalb des Experiments gedacht.

PYTHIA : Ein Monte-Carlo Generator

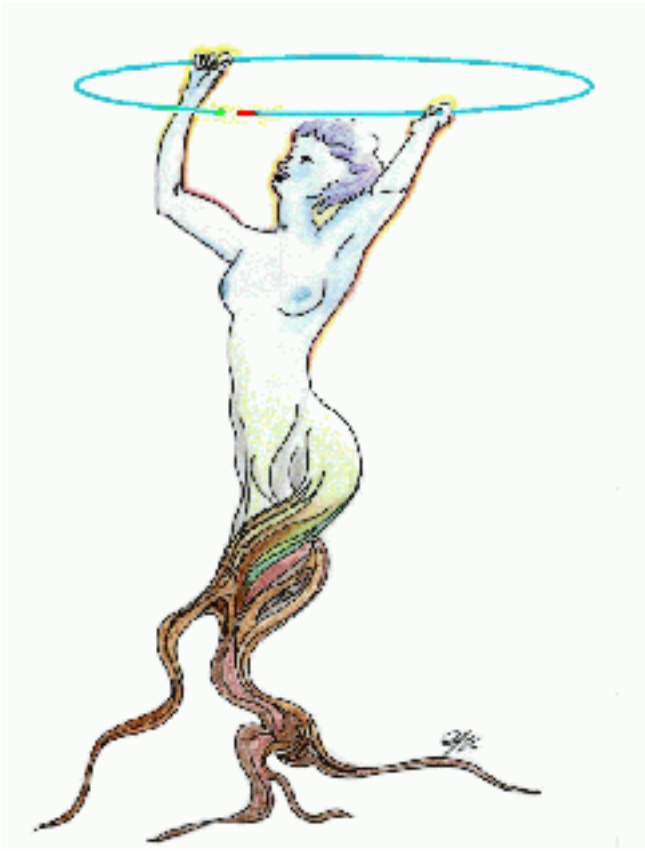
- PYTHIA simuliert Teilchen-Kollisionen
- ca. 60000 Zeilen Fortran 77
- Quelltext frei erhältlich, Autoren werden zitiert
- Entwickelt über 20 Jahre
- Zentraler Bestandteil vieler weitere Pakete
- Unzählige Anpassungen an spezielle Probleme dank freien Quelltextes

CERNLIB

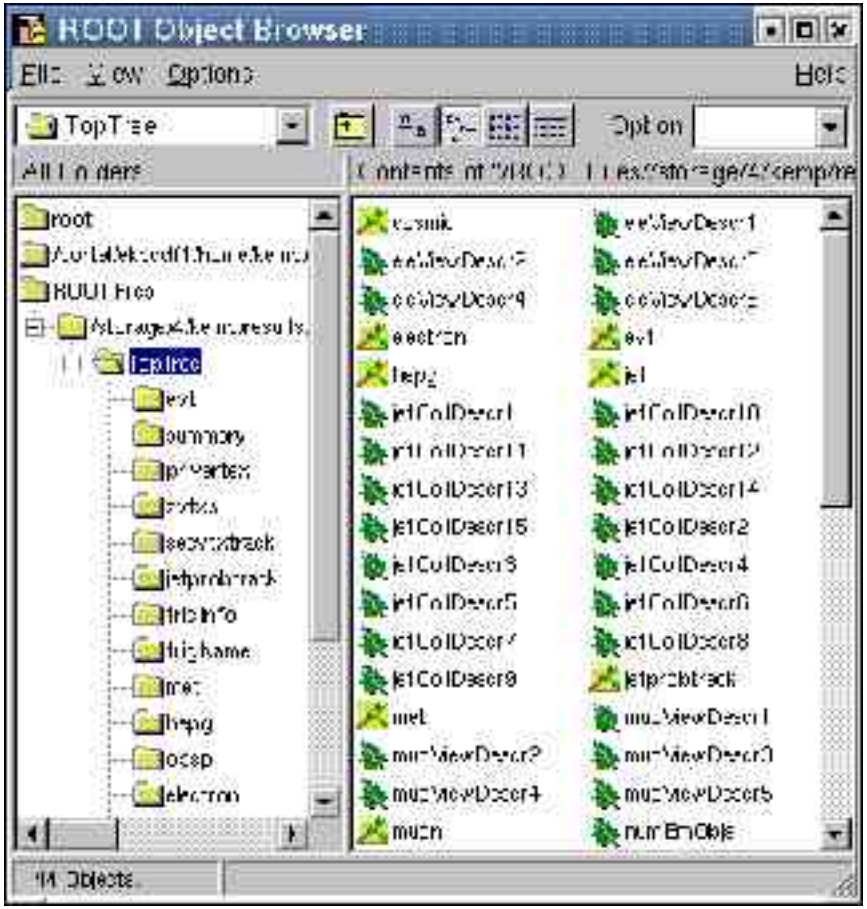
- Seit 1970 entwickelte Bibliothek für HEP Anwendungen (Fortan 77 und C)
- 218 MB Quelltext
- 1996 portieren Freiwillige die CERNLIB auf ein unbedeutendes Unix (Linux)
- 1998 Linux wird offiziell unterstützt
- ~ 2002 CERNLIB wird unter die GPL gestellt.
- 2003 Code-Freeze, keine Weiterentwicklung mehr

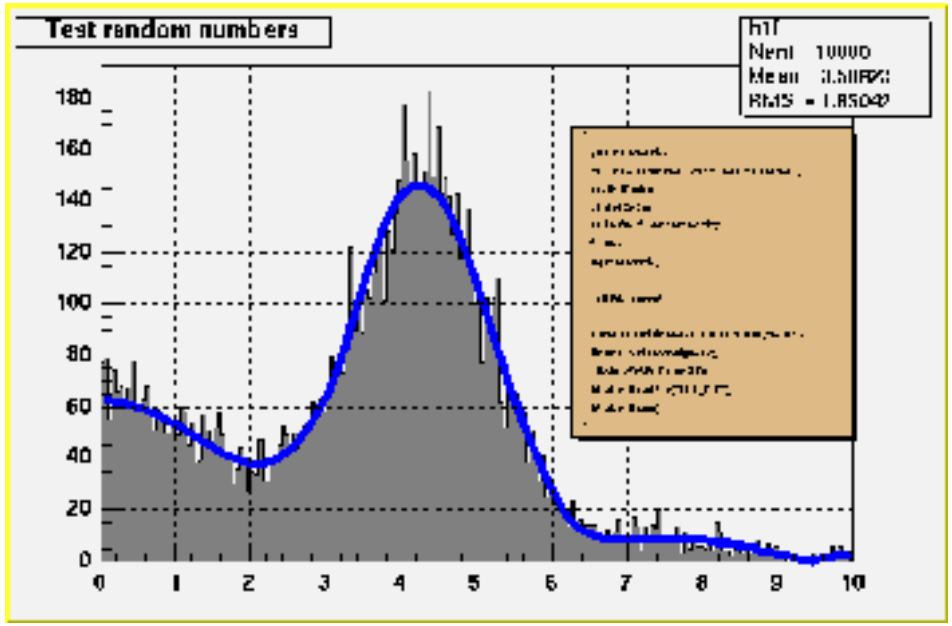
ROOT – OO Datenanalyse Framework

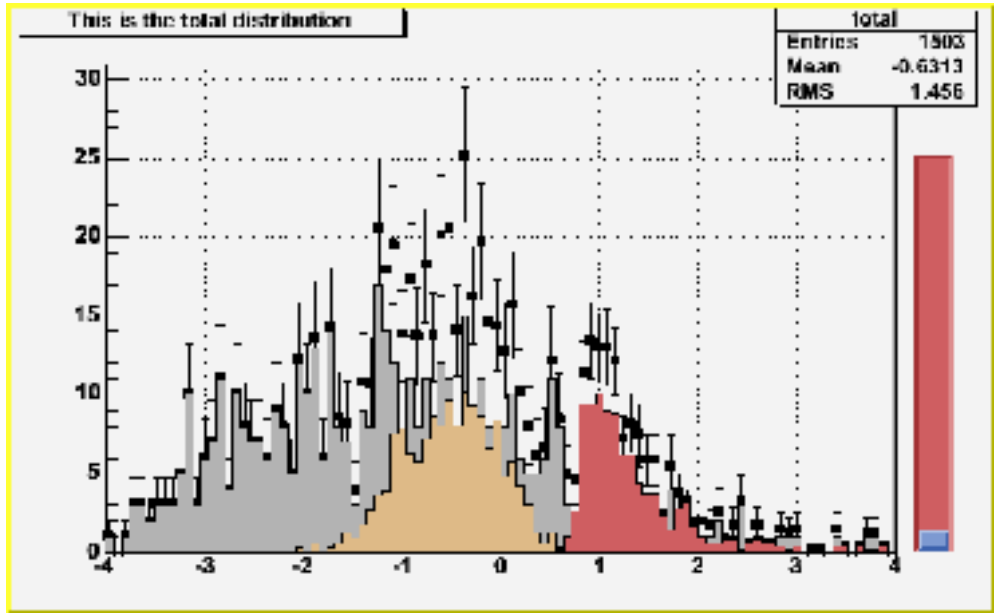
- CERNLIB war ein Erfolg, aber jetzt bitte in C++
- Ziele für ROOT als Nachfol-geanwendung
  - Portabel
  - Erweiterbar
  - Wartbar
- Zentraler Bestandteil der Tevatron/LHC Experimente
- Aktive Weiterentwicklung



Root in Aktion







Experimentsoftware

- Jedes Experiment entwickelt seine eigene Software zur Detektorauslese und Ereignisrekonstruktion
- Warum ?
  - Jedes Experiment ist ein Einzelstück
  - Speziell Elektronikauslese ist zu spezifisch

Open Source als Notwendigkeit

- Die Experiment-Software muss offen sein
  - Jedes Mitglied einer Kollaboration muss Einblick haben/ändern können
  - Lebensdauer der Software kann mehr als 10 Jahre betragen: vom ersten Design bis zum Ende der Analyse
  - Bei Probleme müssen schnell Änderungen/Fixes möglich sein
  - Anpassungen an neuere Hardware/Betriebssystem so möglich

4) GRID computing

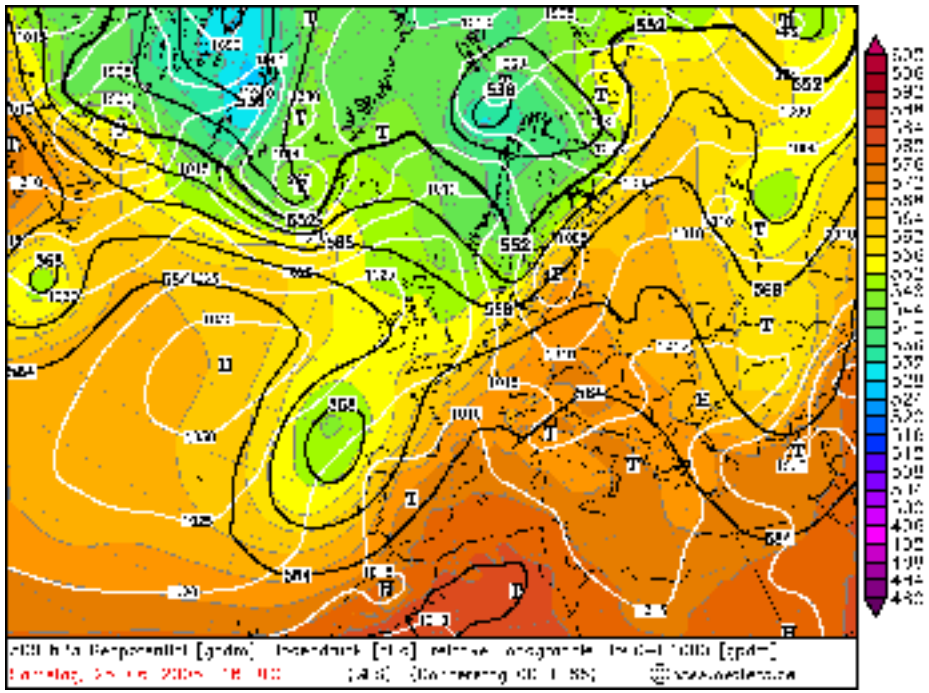
Was ist GRID? Definitionen von Ian Foster:

- Coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations
- The ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose.

Ziele:

- Forschung unabhängig vom Ort zu betreiben
- mit Mitarbeitern zusammen zu arbeiten
- gemeinsam Daten zu benutzen

Analyse: Fundamentaler Unterschied:



z.B. Wettervorhersage:

Daten ändern sich täglich

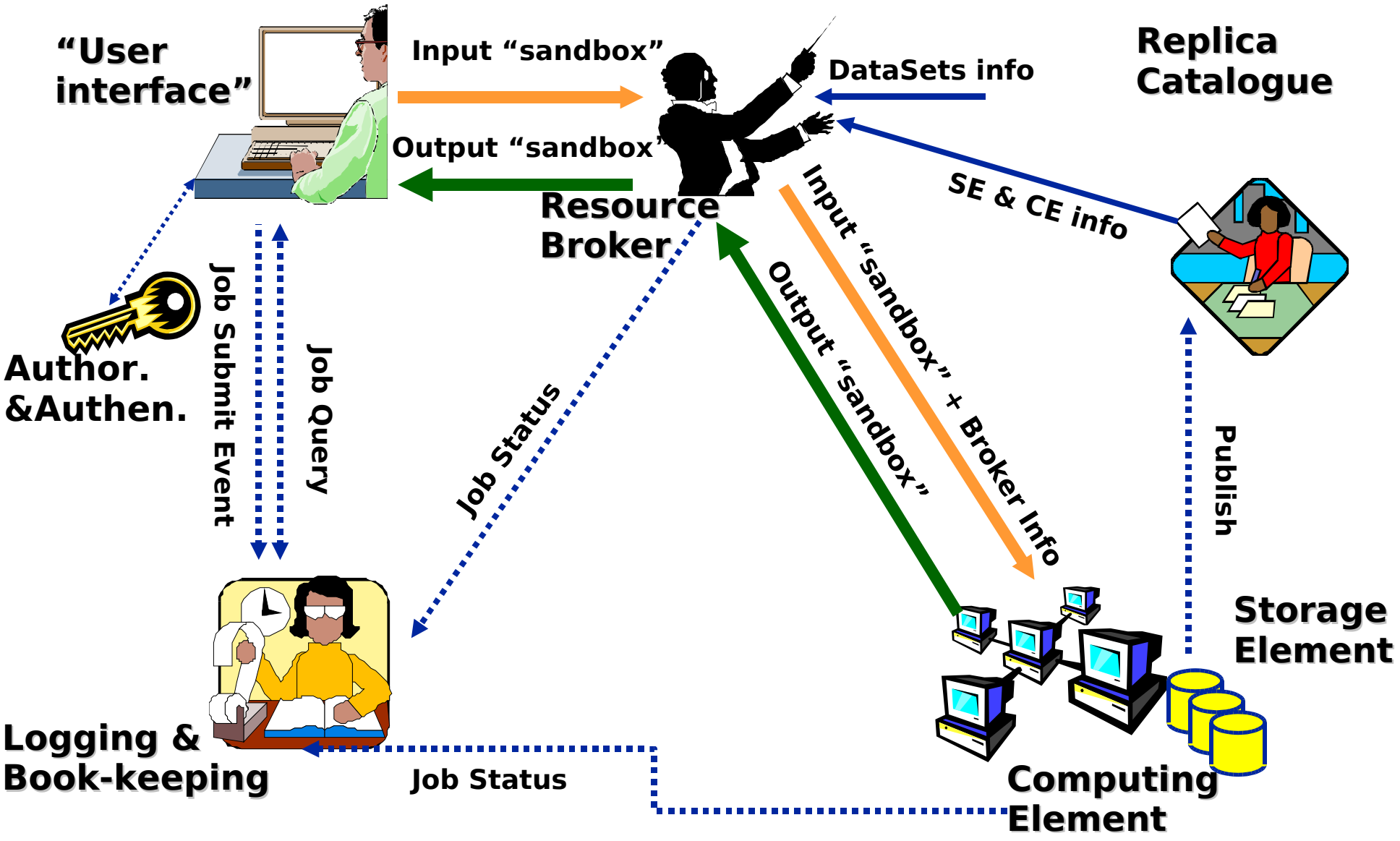
Analyse bleibt die gleiche

HEP:

Datensätze bleiben gleich

Unterschiedliche Analysen, wechseln ständig

Grid Komponenten



Grid am CMS-Experiment:

~ PB/sec

~ 100MB/sec

Online system

Offline farm

IPC* → PIII 1GHz

recorded data

Tier 0

CERN Computer center
~10K PCs*

Tier 1

France Regional Center ~2K PCs

GridKa

Fermilab Regional Center

~ 2.4 Gbits/sec

- Speicherung der Daten
- Große Analysen
- Simulation
- Regional Support

Tier 2

~500 PCs

Tier2 Center | Tier2 Center | Tier2 Center | . . .

~ 0.6 – 2. Gbits/sec

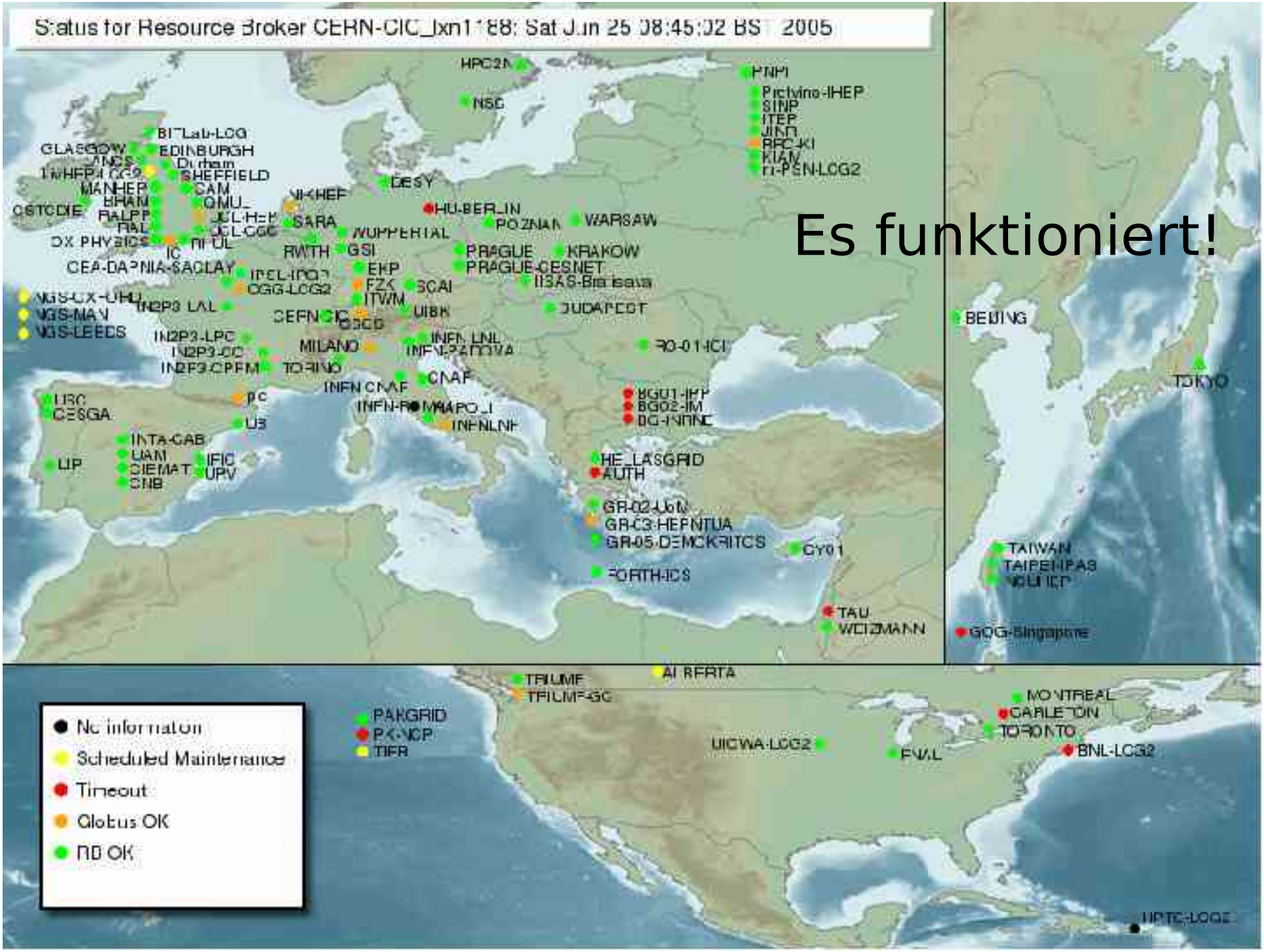
- Zwischenspeicher Daten
- Simulation
- "Normale" Analysen

Tier 3

InstituteA | InstituteB

workstation

~ 100-1000 Mbits/sec


Status for Resource Broker CERN-CIC_lxn1188: Sat Jun 25 08:45:02 BST 2005
Es funktioniert!
No information
Scheduled Maintenance
Timeout
Globus OK
RB OK
CERN-CIC
BEIJING
TOKYO
TAIWAN
TRIUMF
ALBERTA
MONTREAL
TORONTO
BNL-LCG2
FNAL
UICWA-LCG2

“Live” Grid-Demo

- Simulation eines physikalischen Prozesses
- Aufteilung auf mehrere Jobs
- Jobs werden deutschlandweit verteilt (Resource Broker am DESY):
  - Karlsruhe (EKP)
  - Aachen
  - Hamburg (DESY)
- Benutzer in Virtueller Organisation
  - Christopher Jung

Beispiel eines Grid-Jobs

Job lief auf folgenden Clustern:

bigmac-lcg-ce.physics.utoronto.ca	<- Kanada
cmslcgce.fnal.gov	<- USA
ce01.esc.qmul.ac.uk	
ce1.pp.rhul.ac.uk	
epgce1.ph.bham.ac.uk	<-- Großbritannien
heplnx201.pp.rl.ac.uk	
lcgce01.gridpp.rl.ac.uk	
ekp-lcg-ce.physik.uni-karlsruhe.de	<- EKP Karlsruhe
fornax-ce.itwm.fhg.de	<- Fraunhofer Institut Kaiserslautern
grid012.ct.infn.it	
grid0.fe.infn.it	
gridba2.ba.infn.it	
grid-ce.pv.infn.it	<-- Italien
prod-ce-01.pd.infn.it	
t2-ce-01.lnl.infn.it	
ce01-lcg.projects.cscs.ch	<- Schweiz
lxgate15.cern.ch	<- CERN
zeus02.cyf-kr.edu.pl	<- Polen

5) Nutzen für Linux und Open Source

- Nutzen vielfaltig
- Einerseits “soziologisch”
  - HEP-Community ähnlich aufgebaut wie Linux und Open-Source
  - Ähnliche Probleme, nur viel früher
- Geographische Aufteilung
  - Kommunikation?
  - Einsatz von Email, Newsgroup
  - WWW: 1989 am CERN entwickelt von Tim Berners-Lee
  - Die Zukunft: Grid

Science is a community effort. It depends on free access to information and exchange of ideas. In this spirit, **CERN is not an isolated laboratory**, but rather a focus for an extensive community that now includes about 80 countries and about 6500 scientists.

(CERN website)

Entwicklung/Wartung von Programmen

- Einige Beispiel
  - NFS-client
  - GPIB Treiber
  - ATM Treiber

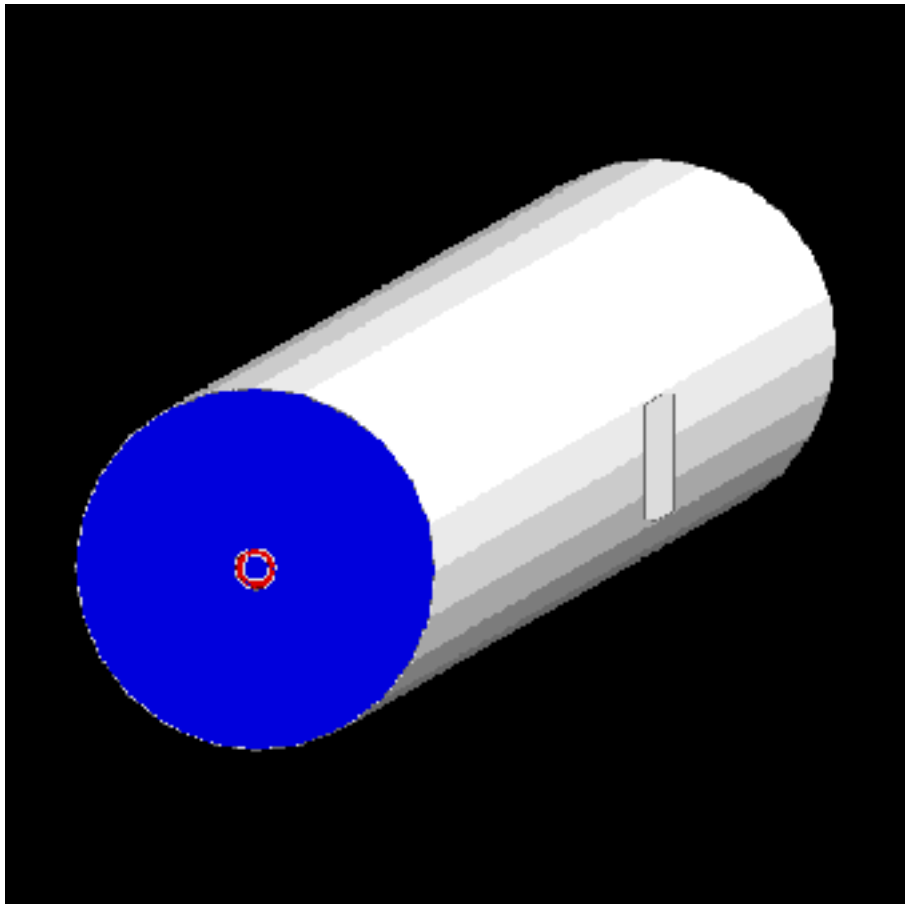
HEP-Community

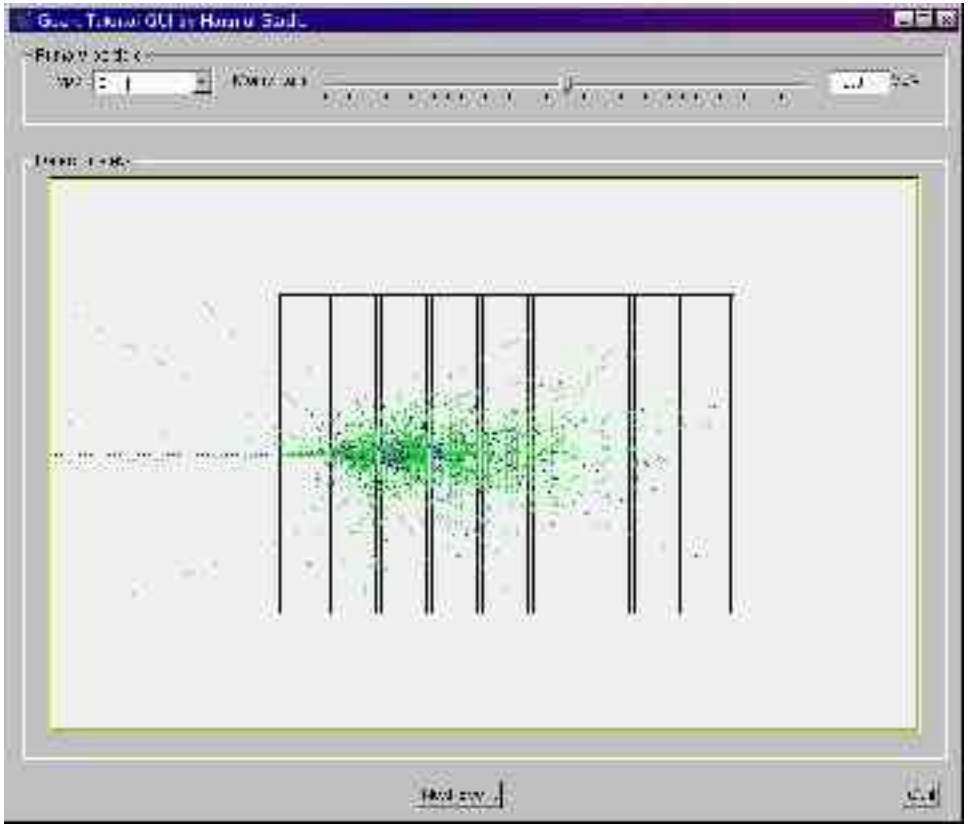
- Große Community
  - Starker Einsatz von Linux und Open-Source
  - Einsatz in kritischen Bereichen
  - Feedback an Entwickler
- EKP: Kontakt mit Distributoren in Früh-phase von Linux für x86-64 CPUs

Einsatz von HEP-Programmen

- ROOT
  - Interface zu Mathematica
  - Einsatz in der Informatik
  - Von manchen sourceforge-Projekten genutzt
  - Medizinische Datenanalyse
- Geant
  - Medizinische Applikationen
  - Dosis-Bestimmung der ISS
  - Strahlenbelastung von Weltraum-Missionen
- Methoden für HEP entwickelt, aber ausserhalb eingesetzt
  - Fitter
  - Statistische Analyse-Methoden

GEANT & Medizin

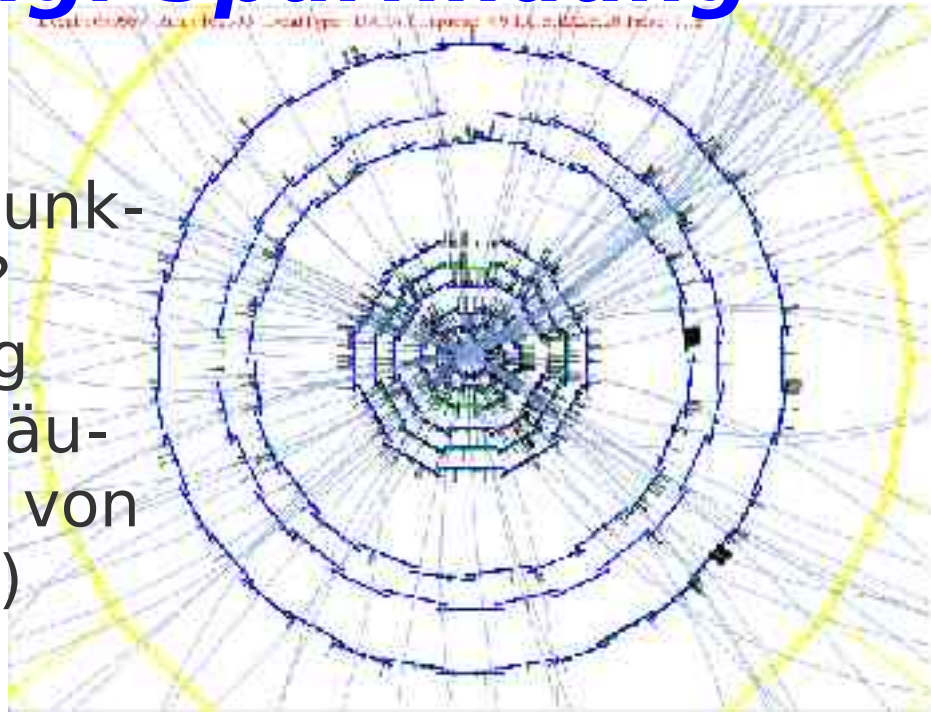




6) Entwicklungen am EKP

- Forschungsschwerpunkt Teilchen-Physik
- Aber: Computing wichtiges Hilfsmittel
  - Software-Entwicklung
  - Portierung auf AMD64
  - Linux-Cluster für GRID

Software-Entwicklung: Spurfindung



- Probleme:
  - Kombinatorik: Welche Messpunkte formen eine Teilchenspur?
  - Statistik: Optimale Schätzung der Spurparameter (muss häufig neuberechnet werden, da von der Teilchenmasse abhängig)
- Weiterhin: Effizienz:
  - Finden, was da ist
  - CPU-Zeit (<1s) und Speicher (<500MB)
- Zuverlässig:
  - Große Datenmengen (~10TB)
  - Online (wochenlang ohne Neustart)

Grundlage der physikalischen Forschung mit dem CDF-Experiment

Software-Entwicklung: Online-Bereich

- Online-Überwachung der Datenqualität
  - 365 Tage im Jahr, 24/7
  - Automatisierte Tests
  - Einheitliches Framework
- Client-Server Architektur
  - Server: Stellt Daten zur Verfügung (~1 Hz) (online!)
  - Consumer: Analysieren Daten (10 Instanzen), erstellt Histogramme, meldet Fehler
  - Display Server: Verwaltet Histogramme
  - Display Client: Zeigt Histogramme an (prinzipiell weltweit)
- 3 Universitäten, ~17.000 Zeilen C/C++ code
- Generisch programmiert: Kann von anderen Experimenten leicht angepasst werden

Software-Entwicklung:

- Abstrahierung komplexer und spezialisierter Detektordaten auf unabhängige und einfache Strukturen (Impuls und Masse)
- Eigentliche Analyse ist unabhängig vom Experiment
- Modularer Aufbau, schlankes Design
- Container und Iteratoren benutzen die *Standard Template Library*
- Benutzt wahlweise CLHEP oder ROOT
- Wird eingesetzt am Tevatron und am CERN (LHC)
- Entwicklungsteam Karlsruhe /Aachen / CERN

Portierung auf AMD64

- sehr kurz nach Einführung des Opterons
- Verschiedene Distributionen
  - 64bit: SLES, Mandrake
  - 32bit: Knoppix, SuSE, RedHat...
- Testen von wichtiger Software (ROOT...)
- Portierung von Software (Geant, Cern-lib, ZFITTER)
- Evaluierung in Dauereinsatz
- 32bit/64bit

→ Stand LinuxTag 2003

Linux-Cluster für Grid

- GRID: “Wir benutzen unsere Ressourcen gemeinsam wenn wir sie brauchen”
  - Viele Ressourcen benötigt
  - Gutes Preis/Leistungs-Verhältnis
  - “Wir steuern das Ding selbst”
  - Unterschiedliche Gruppen
  - Corollar 1: “Gleiches (ähnliches) Betriebsys-tem” (Linux)
  - Corollar 2: “Gleiche (ähnliche) Hardware-Plattform” (x86-(64), Consumer-Hardware)
  - Corollar 3: “Unterschiede auf experiment-spezifische Rechner ausgelagert”

Umsetzung:

- Rechenknoten:
  - x86-(64), momentan noch im 32-bit Modus
  - modifiziertes Linux (RedHat oder Scientific Linux), bootet über Netz
  - einfach austauschbar
  - Zur Zeit 27 Rechner und 36 CPU
- Speicherplatz:
  - 3 NAS-Boxen, exportieren per NFS
  - Intern IDE/SATA-Platten, RAID-5
  - Debian (wegen Wartbarkeit)
- Benutzerverwaltung:
  - NIS und NFS
- Netzwerk:
  - 100Mbit/Gbit Ethernet
  - Myrinet/InfiniBand: Momentan kein Bedarf

Umsetzung, cont.

- Experiment-spezifische Rechner:
  - Login-Portale
  - Enthalten eventuell spezielle Linux-Distro
  - Experiment-Software, Kompiler, ...
  - exportieren per NFS an die Knoten
- Batch-Queue:
  - Anfangs OpenPBS, jetzt Maui/Torque
  - Fair-Share
  - Externe Benutzer können via Grid-Rechner an lokale Queue submittieren
- Monitoring:
  - Ganglia
  - Temperatur und Strom: Selbst gestrickt

Exkurs: Was ist  Scientific Linux?

- Vergangenheit:
  - Große Zentren (CERN, FNAL,...) setzen auf Red Hat
  - modifizieren Distro
- Probleme:
  - Lizenzpolitik von Red Hat:
  - Lizenz/Support nur für Red Hat Enterprise: teuer
- Lösung:
  - Eigene Distro: Zuerst HEP Linux
  - Interesse von anderen Zweigen: Scientific Linux
  - rekompilierter Enterprise Server
  - binärkompatibel zu Enterprise Server
  - Erweiterungen, je nach Experiment
  - Keine Lizenzkosten, Support und Security Updates von CERN/FNAL oder der Community

Erfahrungen mit Scientific Linux

- Am EKP: Scientific Linux CERN Release 3.0.4
- 2.4.21 Kernel (2.6 erst mit release 4)
- Unterstützte Architekturen: i386 und x86_64
- Installation auf Kontrollrechner
- Installation von Rechenknoten
- Umstieg von RedHat 7.3 auf SL: problemlos
- Distribution sehr stabil
- Unter RH 7.3 kompilierte Binaries laufen
- Umstieg auf allen Servern geplant
- Umstieg auf Desktop-Rechnern nicht geplant

Zusammenfassung

- HEP sehr interessantes Feld
- Großer Einsatz von Rechnern
- Linux Betriebsystem Nummer 1 geworden
- Open-Source Mentalität in der HEP
- Einsatz von Linux und Open-Source zu beiderseitigem Nutzen
- EKP in Karlsruhe sehr aktiv

Linux everywhere

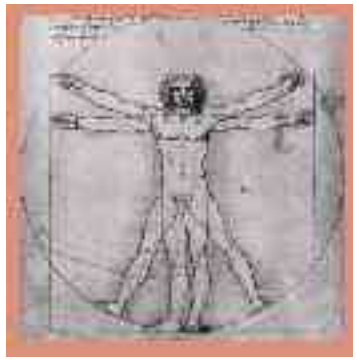
→ auch in der HEP

Backup Slides

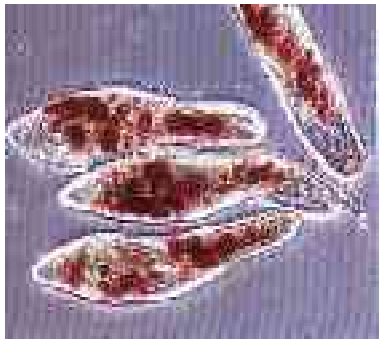
Skalen der Physik

10^{26} m

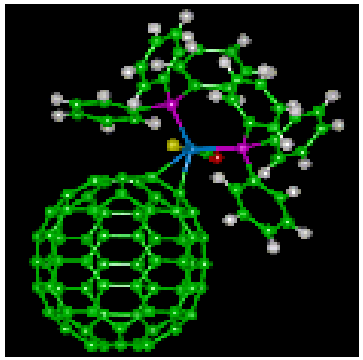
10^{7} m



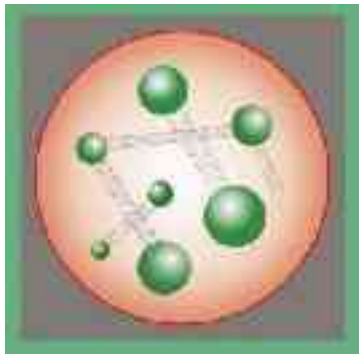
10^{1} m



10^{-4} m

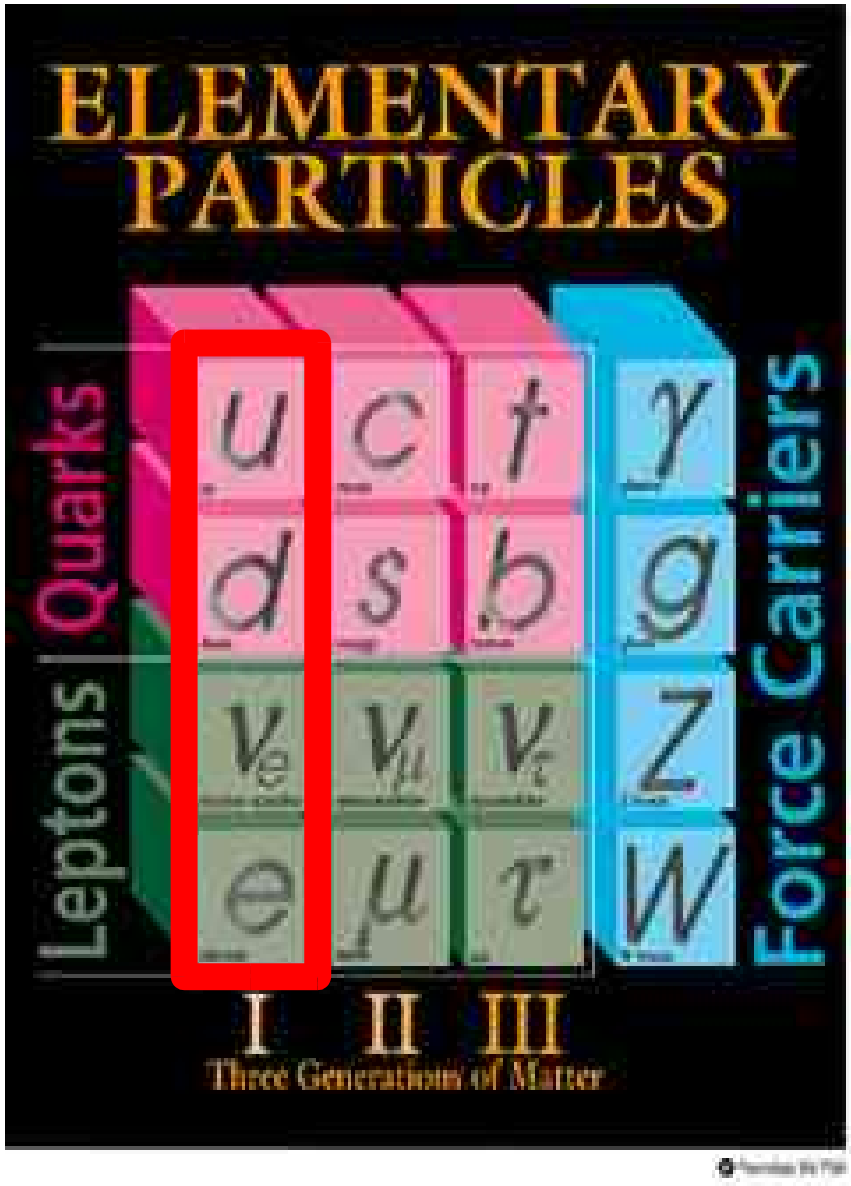


10^{-8} m



10^{-15} m

Standardmodell der Teilchenphysik (SM)

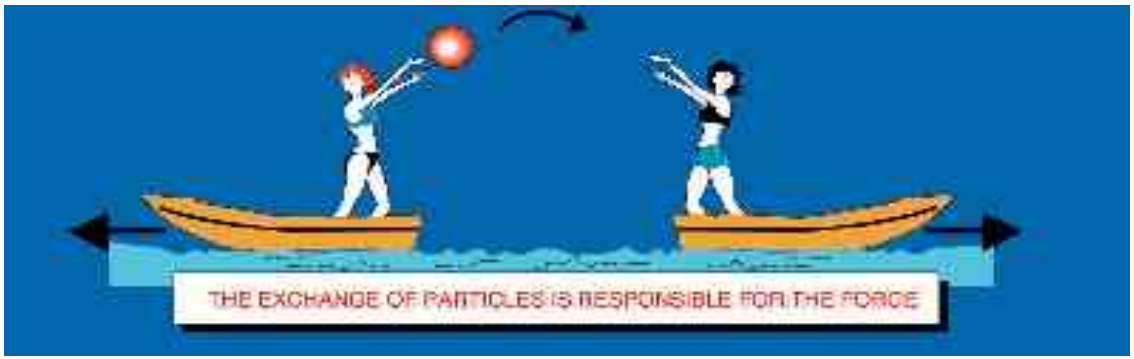


Baukasten der Natur!

alle bekannten Teilchen lassen sich aus diesen zusammensetzen

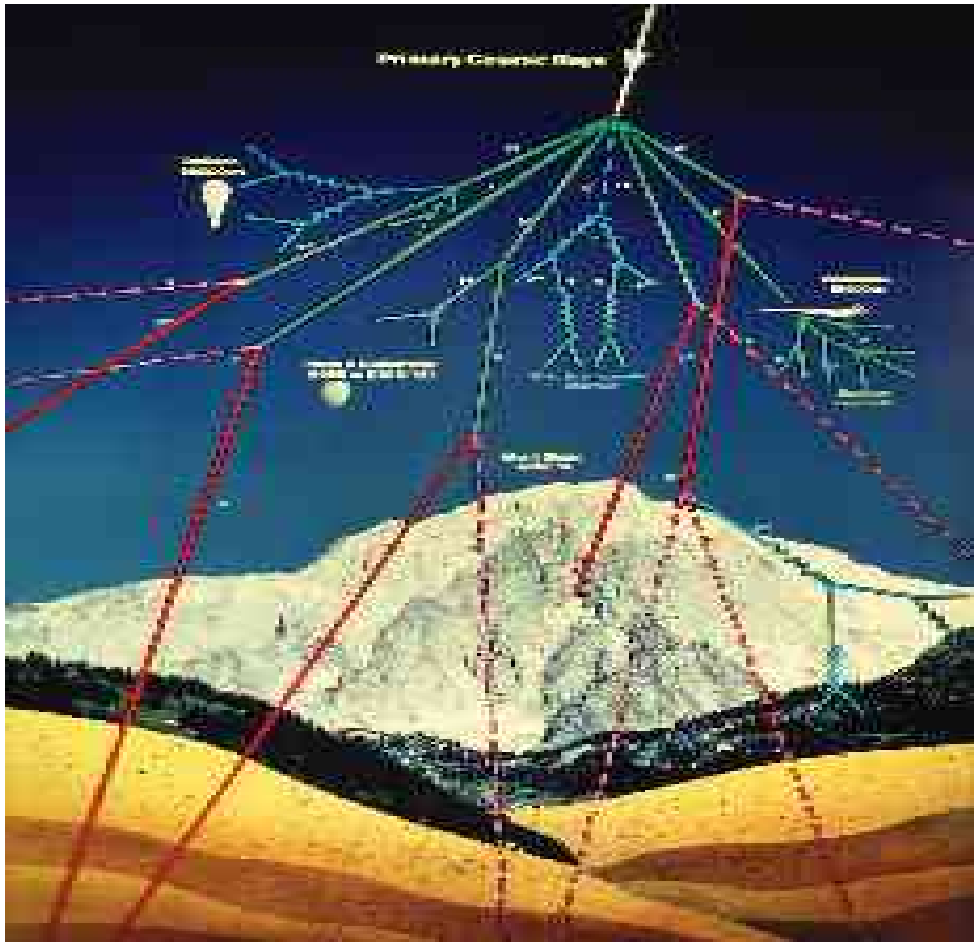
alle stabile Materie besteht nur aus Kombinationen der 1. Generation

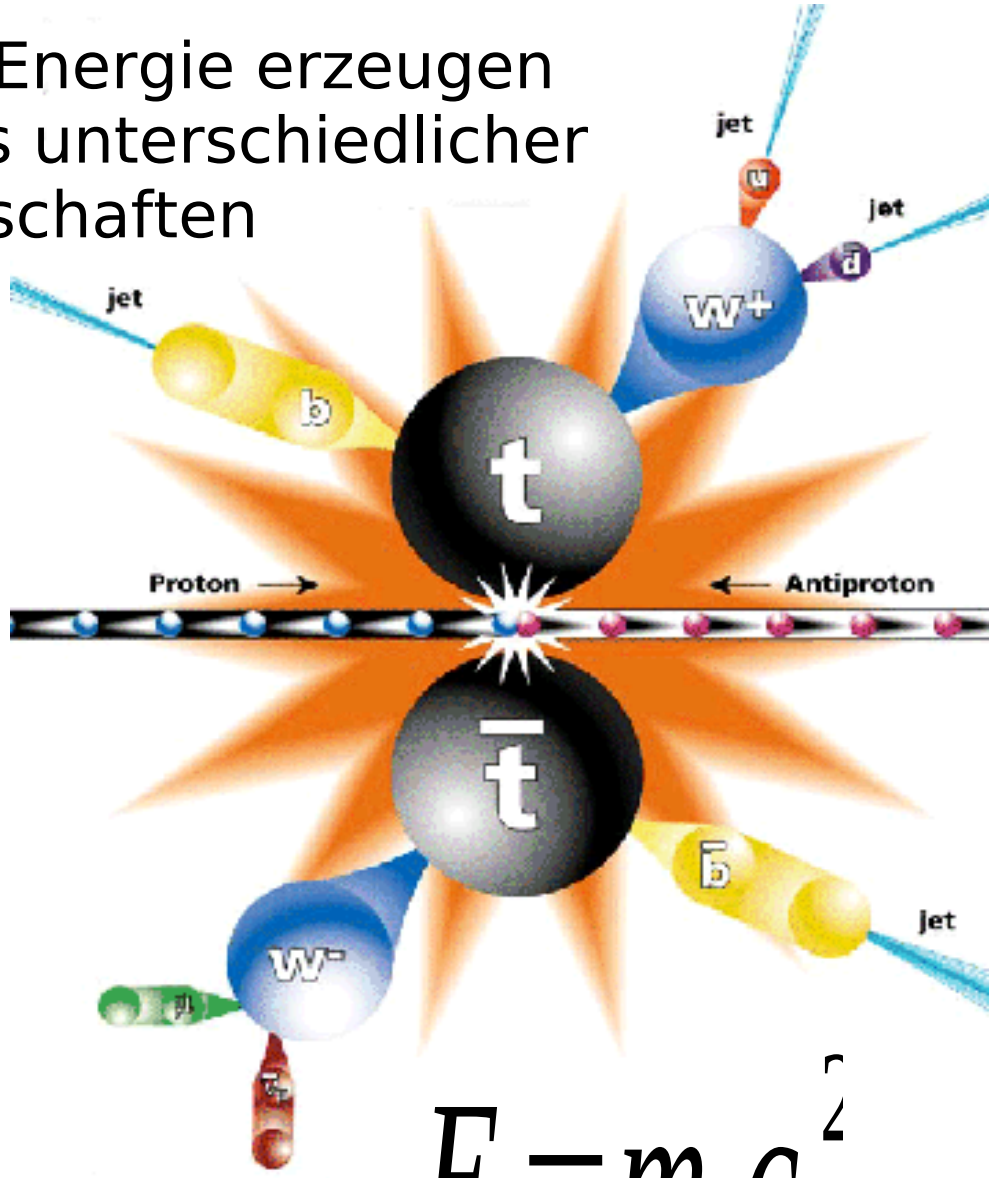
Beispiel:
Proton (uud) Neutron (udd) Elektron



Wie wird neue Materie aus allen Teilchen erzeugt?

Teilchenkollisionen mit hoher Energie erzeugen viele neue Teilchen mit jeweils unterschiedlicher Zusammensetzung und Eigenschaften





Online-Software bei CDF

- Detektorauslese über VME-Elektronik
  - VME Crate CPU (PPC mit Vxworks) als Interface zur Kontrolsoftware
  - RealTime Linux zum Designzeitpunkt noch nicht verfügbar
- Steuersoftware in Java
  - Java um Hardwareabhängigkeiten zu vermeiden
  - Linux als Plattform des Online-Clusters