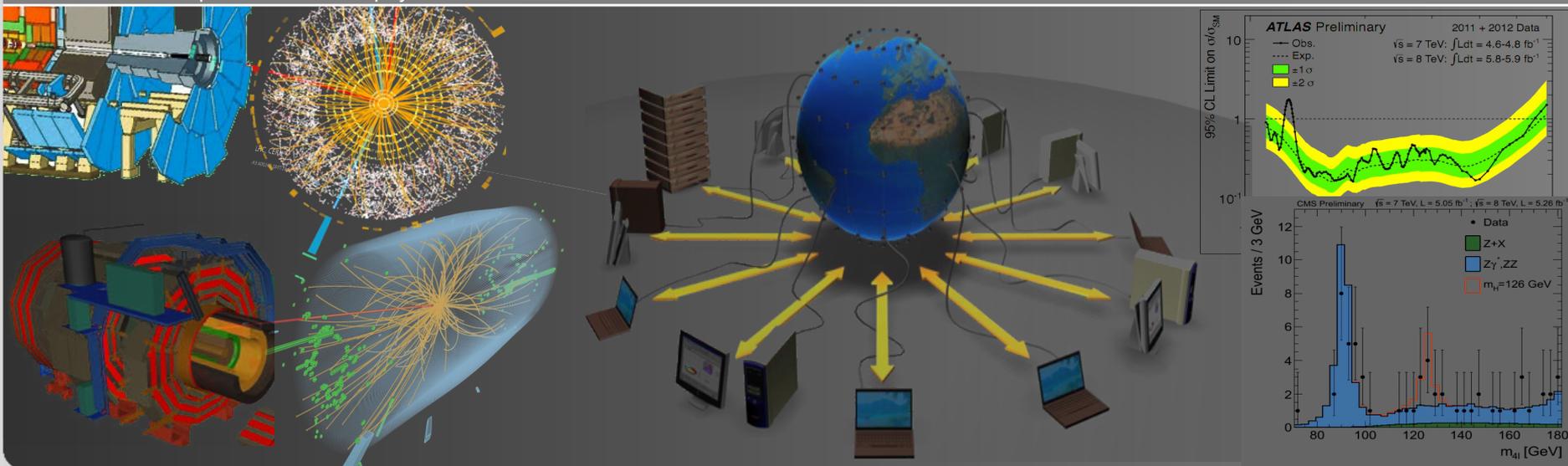


Next-Generation Software und Computing für die LHC - Datenanalyse

Günter Quast

Fakultät für Physik
Institut für Experimentelle Kernphysik

DPG - Tagung Mainz, 26. März 2014



Inhalt

- World-Wide LHC Computing Grid
- Neue Herausforderungen
in Software-Entwicklung und Computing
- Aktivitäten im Long-Shutdown1
⇒ „Next-Generation Computing“ für die Teilchenphysik

Bestandsaufnahme

WLCG: Worldwide LHC Computing Grid



Steckbrief (Stand 2013):

250'000 Prozessorkerne

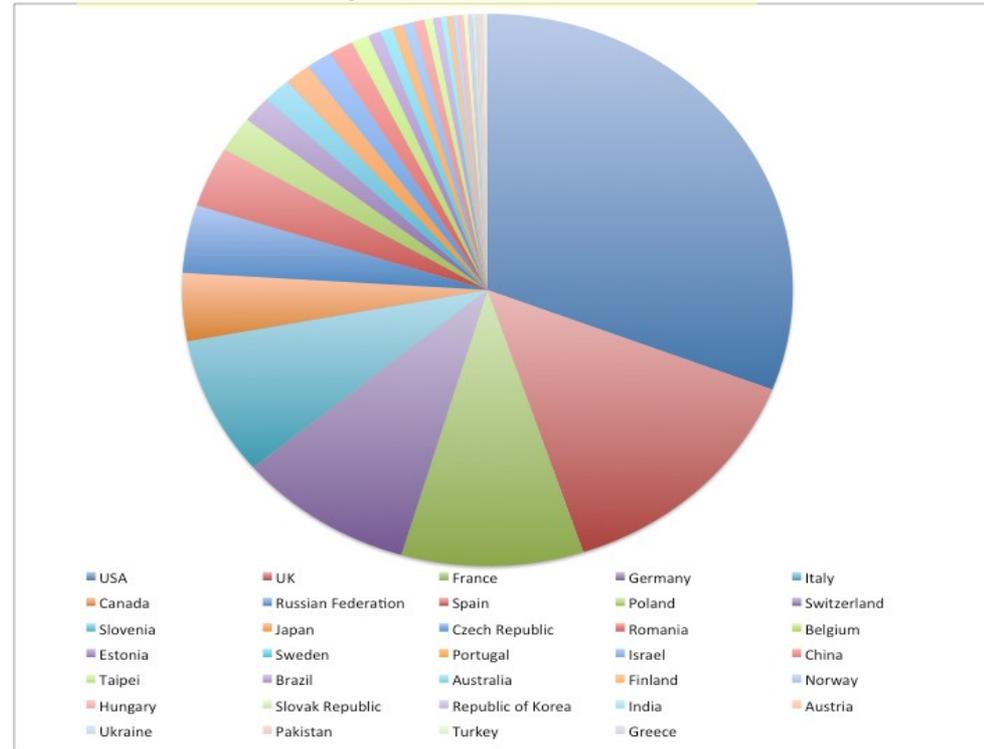
200'000 TB Plattenspeicher

200'000 TB Bandspeicher

1,5 Millionen Jobs/Tag

10 GB/s weltweite Datentransfers

CPU-Beitrag einzelner Länder



Deutscher Anteil

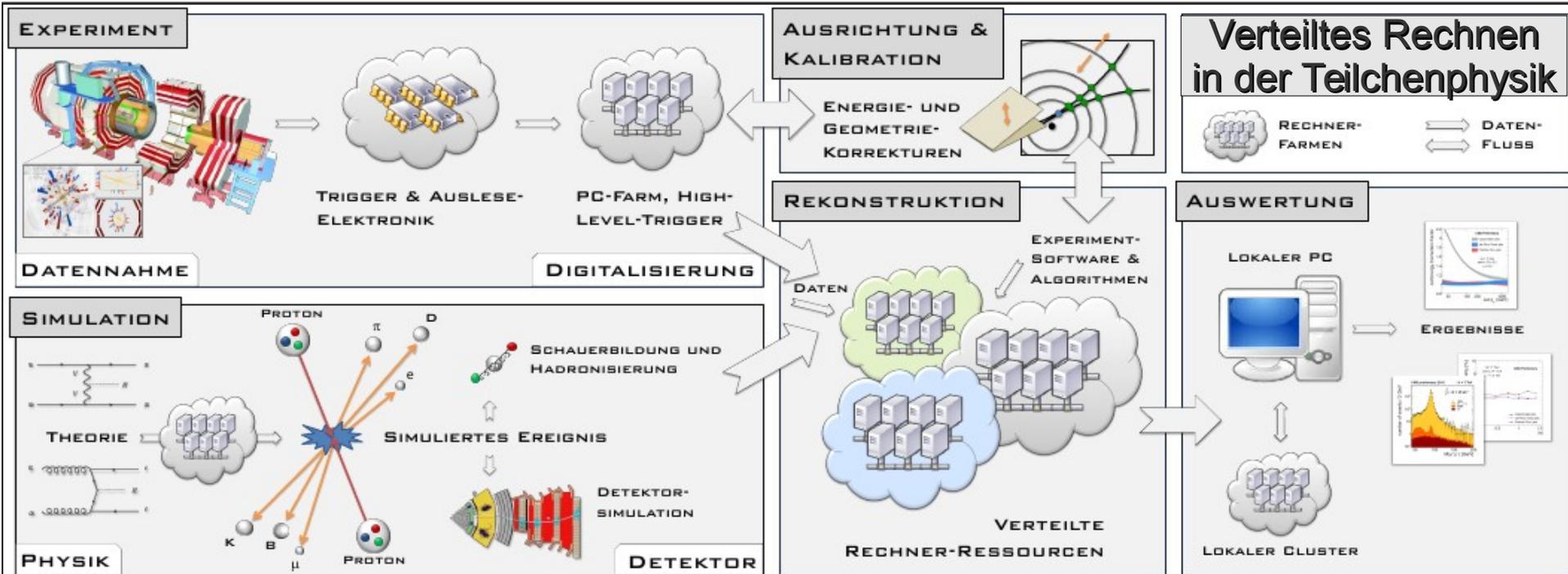
~15% Tier-1

~10% Tier-2

T1: GridKa (KIT), T2: DESY, MPG,
Univ. Aachen, Freiburg, Göttingen,
München, Wuppertal

sehr zuverlässig und sichtbar

Heterogenes Spektrum von Anwendungen



Breite Palette an Anwendungen mit ganz unterschiedlichem Bedarf an CPU-Leistung und I/O:

- zentral organisierte Simulation und Rekonstruktion großer Datensätze
benötigen viel Rechenleistung
- Datenauswahl und Verteilung
benötigen hohe I/O- und Netzwerk-Bandbreite
- Analyse durch einzelne Physiker mit wahlfreiem Zugriff auf Daten und Ressourcen mit oft unausgereiftem Code → die wirkliche Herausforderung !



Global Effort → Global Success

July 4, 2012

Results today only possible due to extraordinary performance of accelerators – experiments – Grid computing

Observation of a new particle consistent with a Higgs Boson (but which one...?)

Historic Milestone but only the beginning

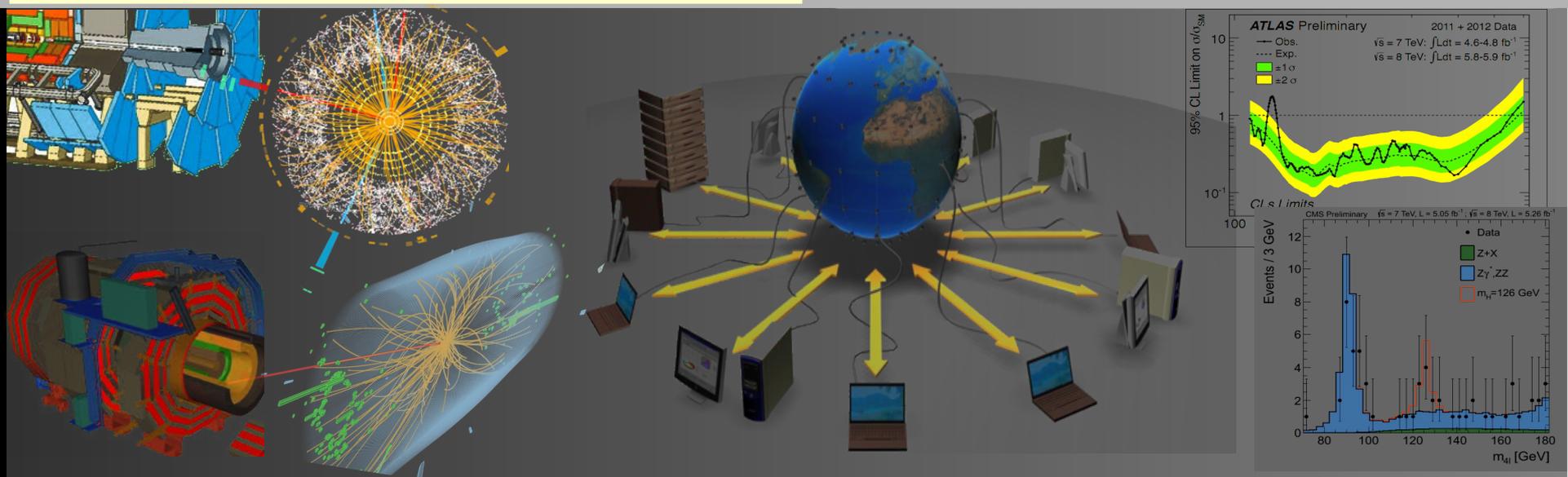
Global Implications for the future



R-D Heuer

Großer Erfolg des Physik-Programms am LHC auch Dank der hervorragenden Leistungsfähigkeit des Computings

Did the LHC Computing work?



July 4, 2012

Global Effort → Global Success

Results today only possible due to extraordinary performance of accelerators – experiments – Grid Computing

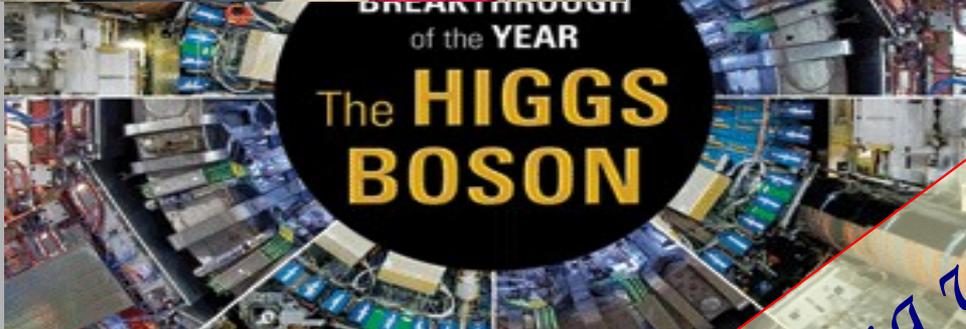
Observation of a new particle consistent with a Higgs Boson (but which one...?)

Historic Milestone but only the beginning

Global Implications for the future



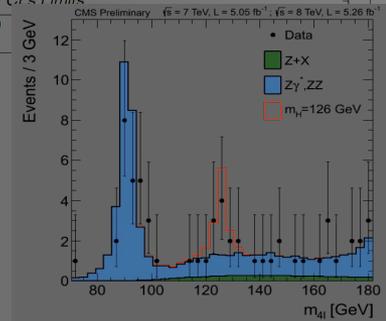
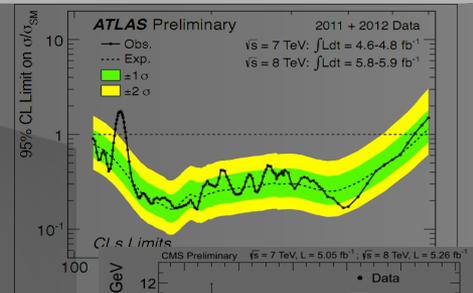
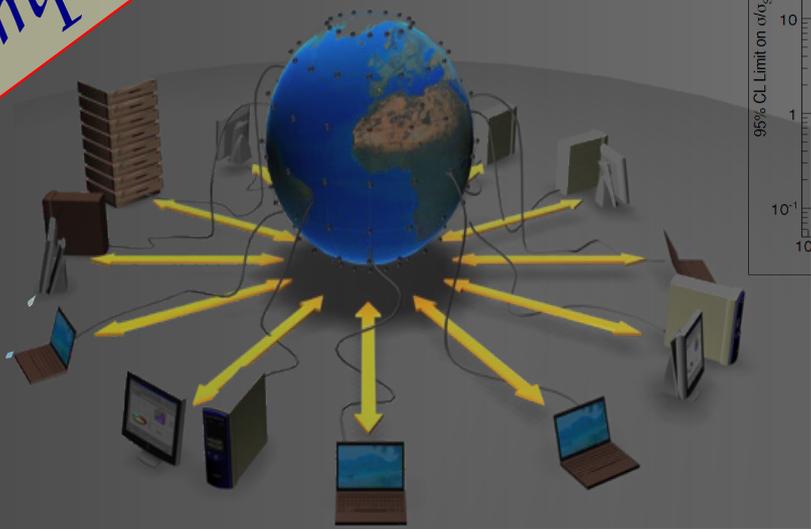
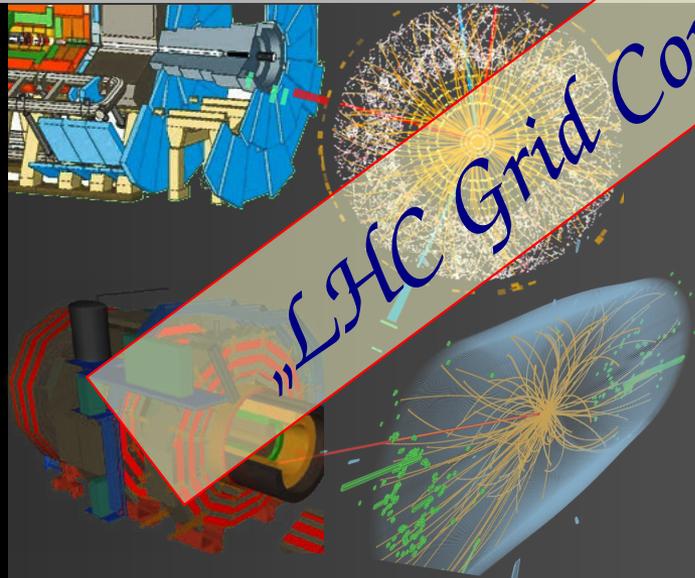
R-D Heuer



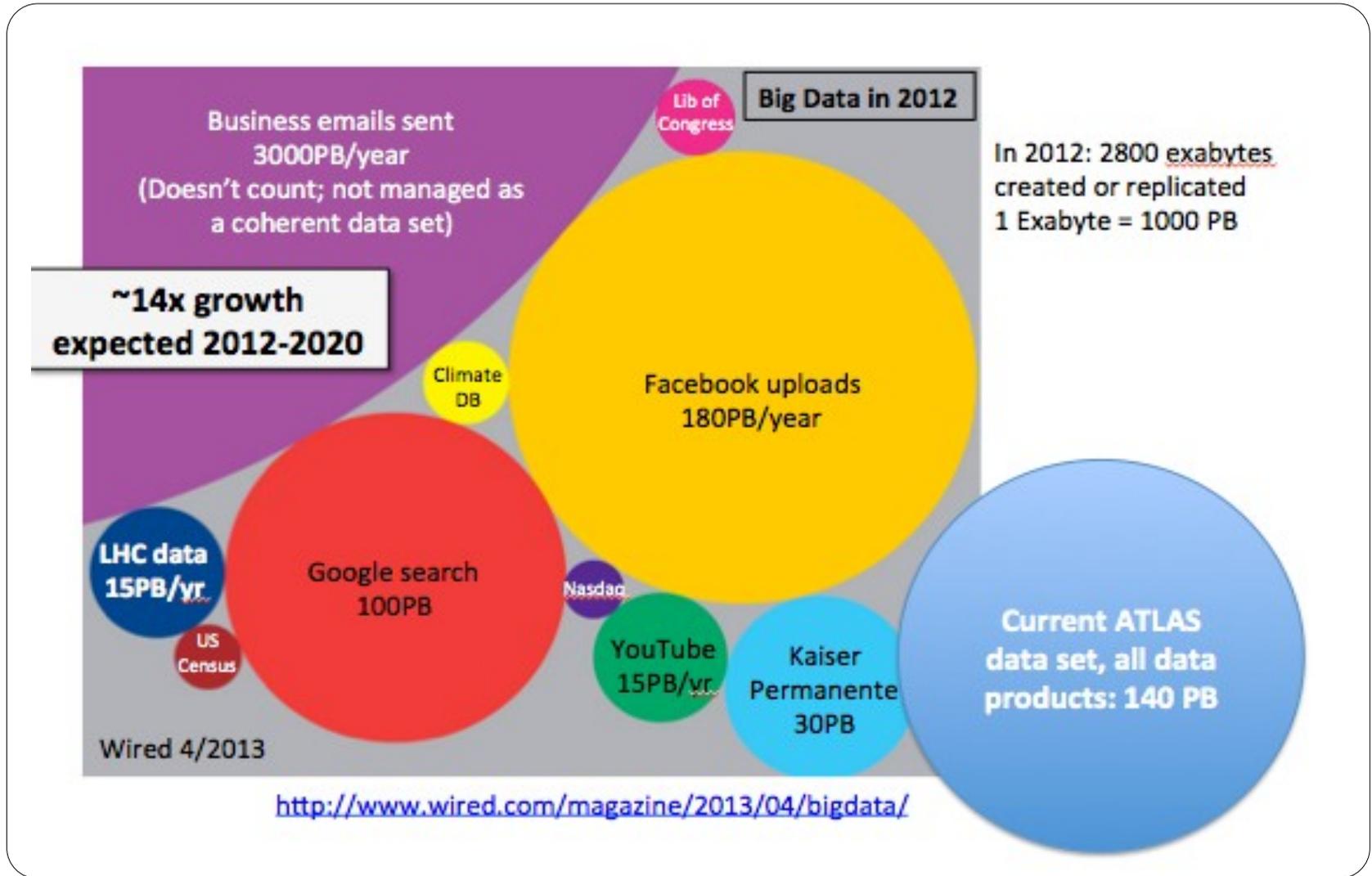
Did the LHC-Computing work?

Großer Erfolg des Physik-Programms am LHC auch Dank der hervorragenden Leistungsfähigkeit des Computings

LHC Grid Computing was a success story



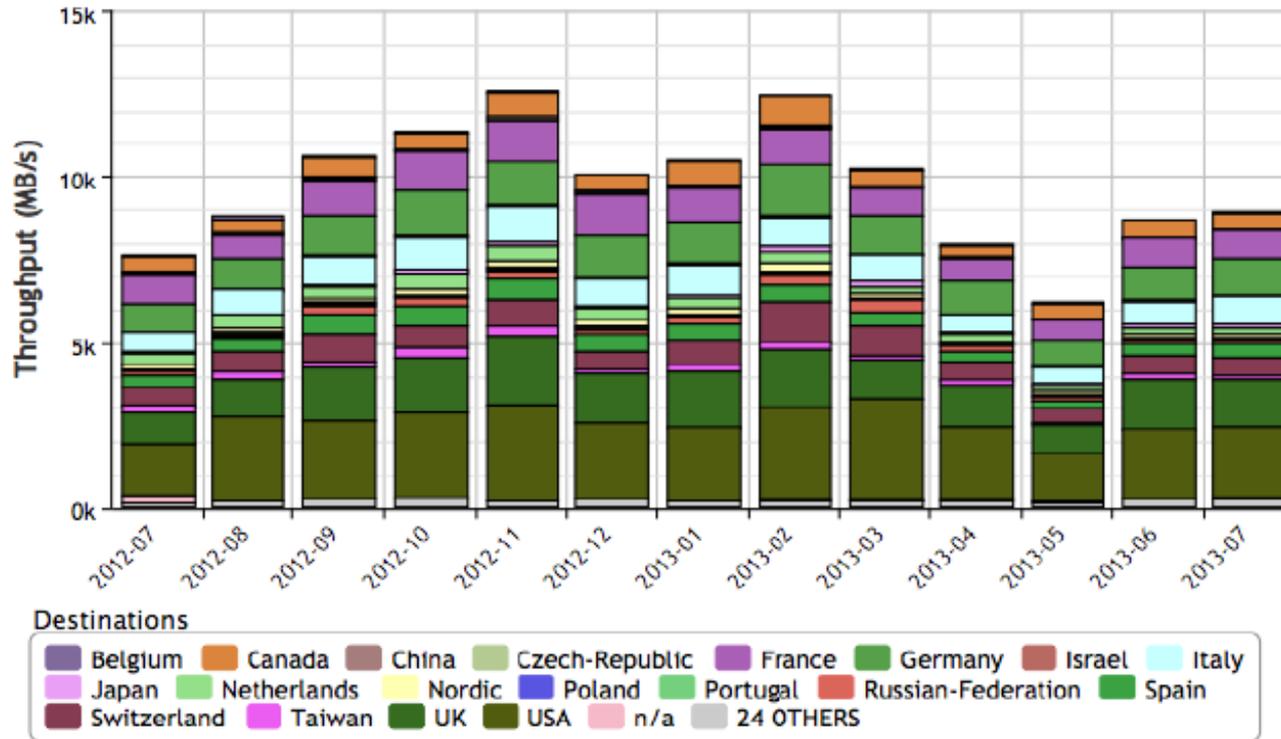
Ist HEP „Big Data“ ?



JA !

Weltweite Datentransfers

Datentransfer zwischen Tier-0 / -1 / -2-Zentren Juli '12 – Juli '13

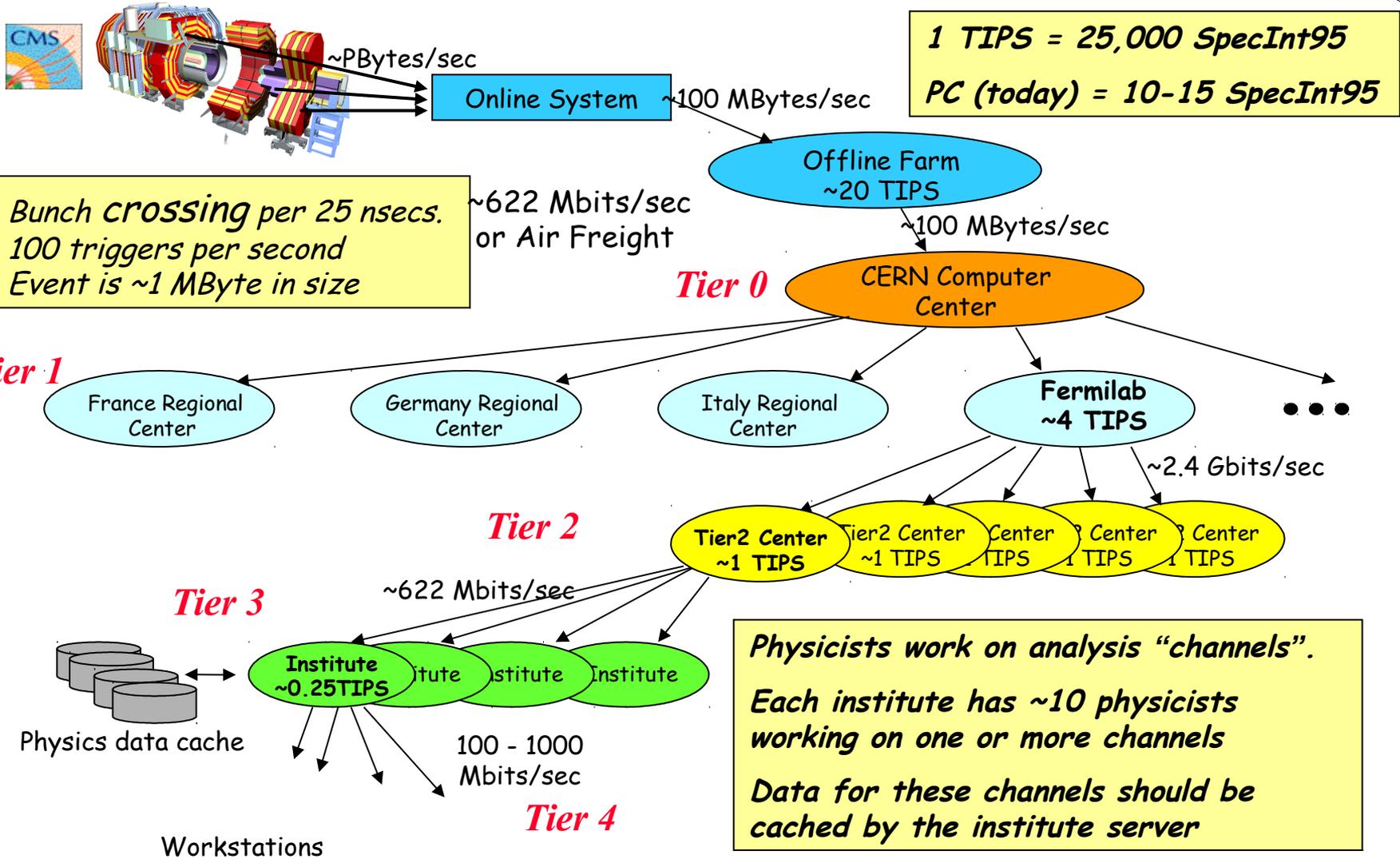


Im Mittel ~10 Gbyte/sec \cong 25 PByte / Monat

- kein Einbruch nach Ende der Datennahme:
Reprozessierung, Monte-Carlo-Produktion, Datenverteilung, Analyse

„Im Computing gibt es keinen Shut-down!“

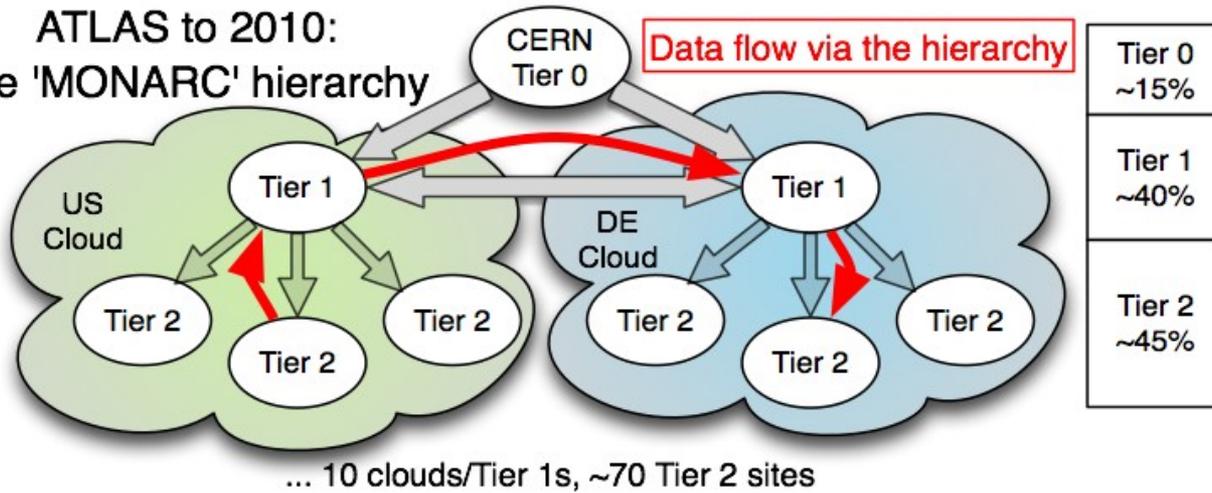
Ursprüngliches Computing-Modell (Monarc '99)



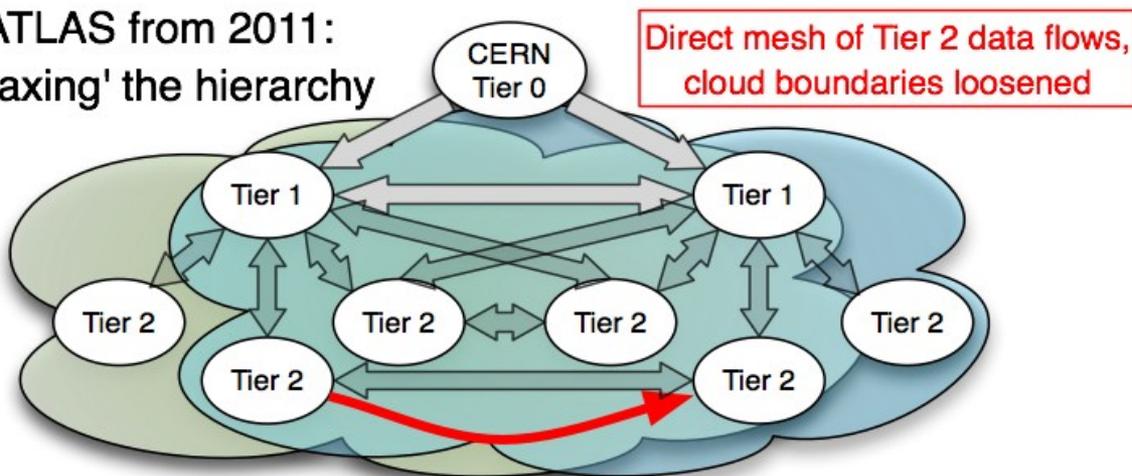
1999: Hierarchisches Modell mit starker regionaler Abhängigkeit T1 → T2 → T3

Modelle entwickeln sich: Bsp. ATLAS

ATLAS to 2010:
The 'MONARC' hierarchy



ATLAS from 2011:
'relaxing' the hierarchy

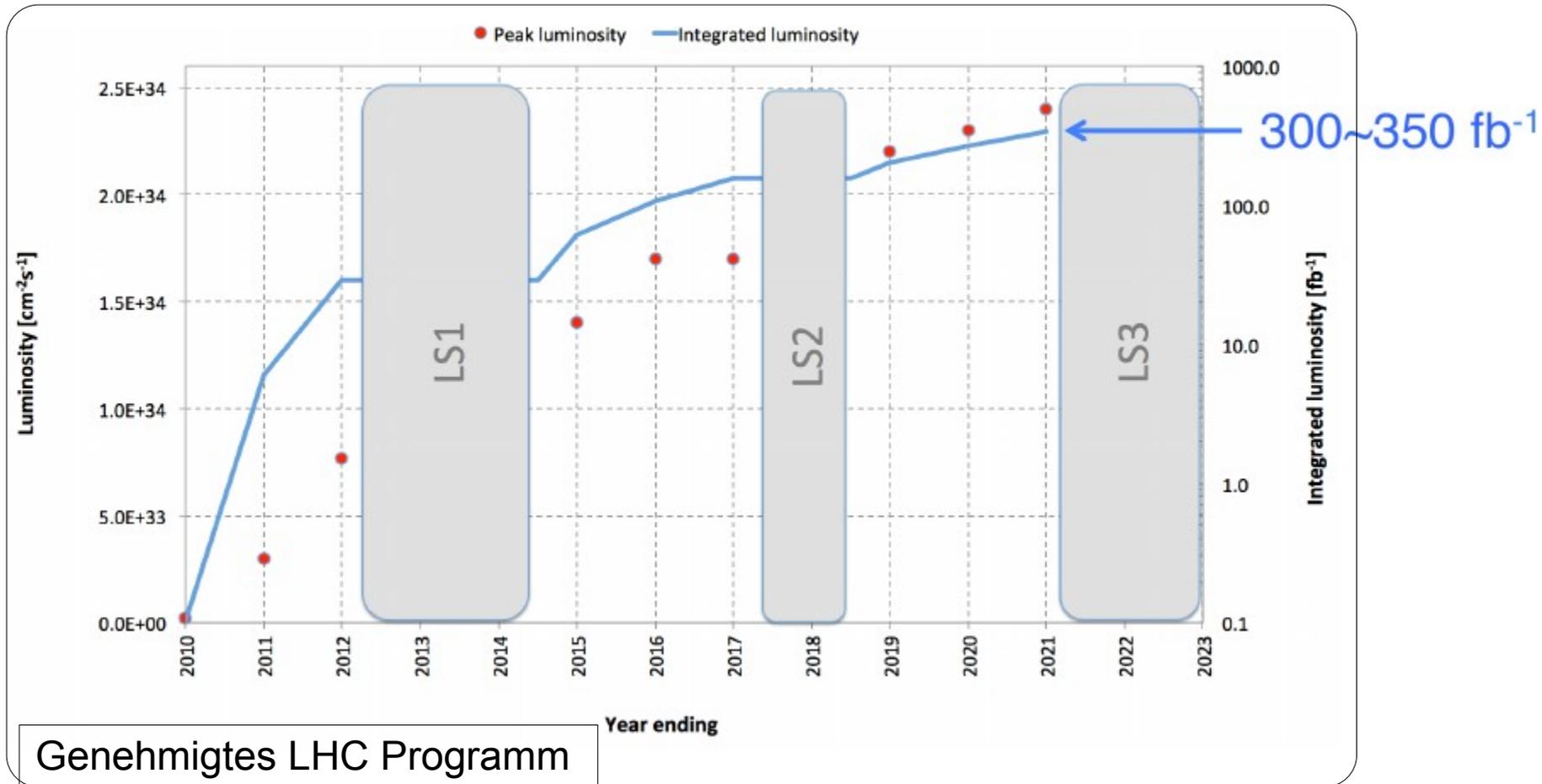


Unterschiede zw.
T-0 / -1/ -2 lösen
sich auf

Anm.: CMS hat
ein solches „Mesh“-
Modell schon
länger

Heute: **Datentransport i. A. billiger als Datenspeicherung !**

Zukünftiges LHC – Programm ...



... führt zu erhöhten Anforderungen auch an Software & Computing

Steigende Anforderungen im LHC-Run 2

- 7/8 TeV → 13/14 TeV
 - höhere Luminosität
 - höhere Trigger-Raten geplant
- O(einige 100 Hz) → 1 kHz

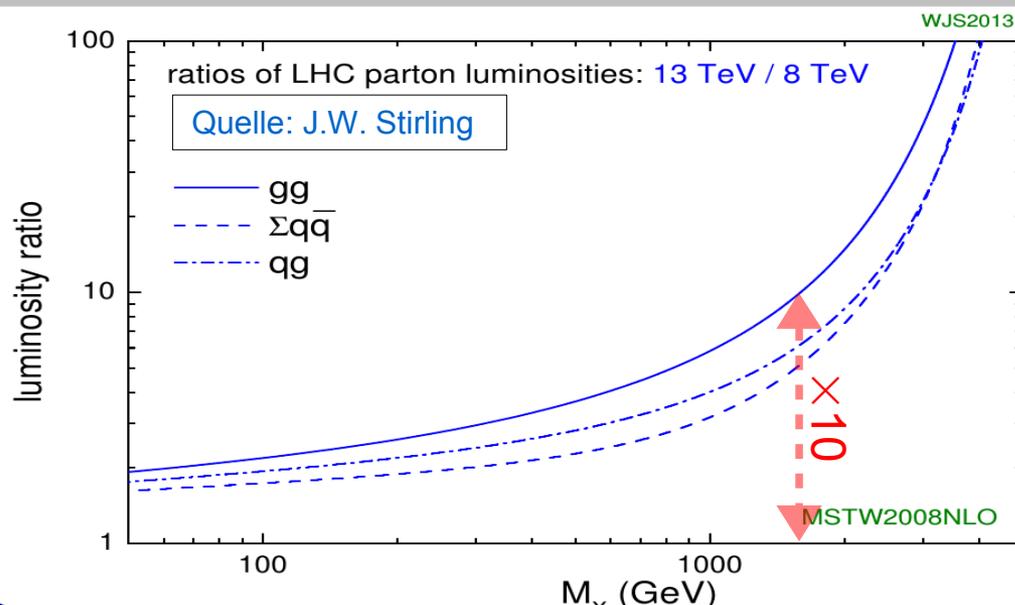
- mehr Ereignisse
- komplexere Ereignistopologie
- höherer (auch out-of-time) Pile-Up

4 mal **höhere integrierte Luminosität** Ende 2017
und **>2 mal längere Laufzeit**
der Rekonstruktionssoftware

d.h. Computing-Bedarf eine Größenordnung höher !!!

aber: im günstigsten Fall **flaches Budget** für Ersatz alter & für neue Hardware sowie Personal
Moore's Law hilft nur noch bedingt!

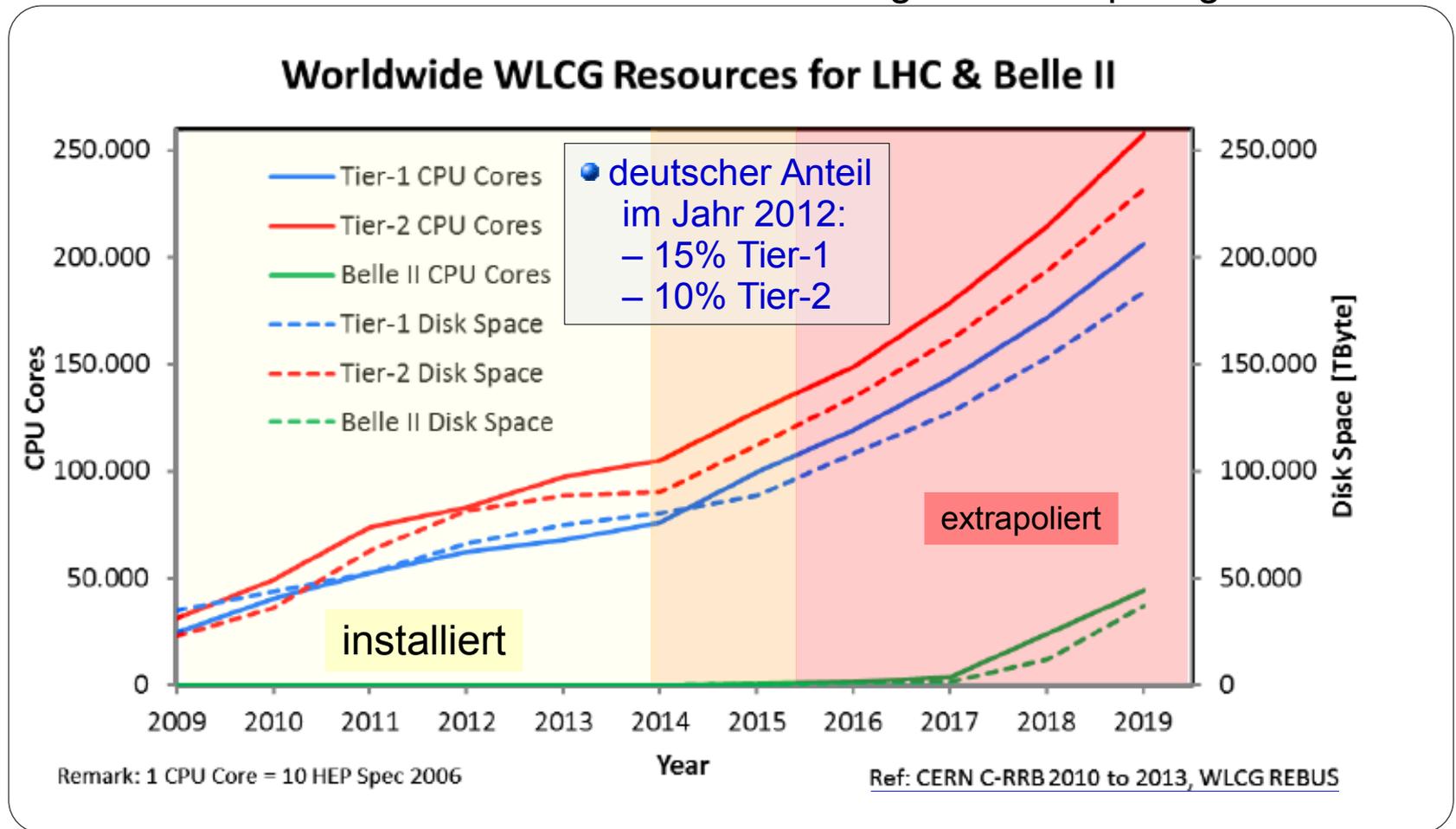
Verhältnis der Parton-Luminositäten bei 13 und 8 TeV



⇒ **Müssen uns auf „Ressourcen-limitierten“ Betrieb einstellen und Optimierungspotenzial ausschöpfen sowie zusätzliche, opportunistische Ressourcen nutzen**

Entwicklung der Ressourcen 2015-19

Annahme: im wesentlichen flaches Budget für Computing



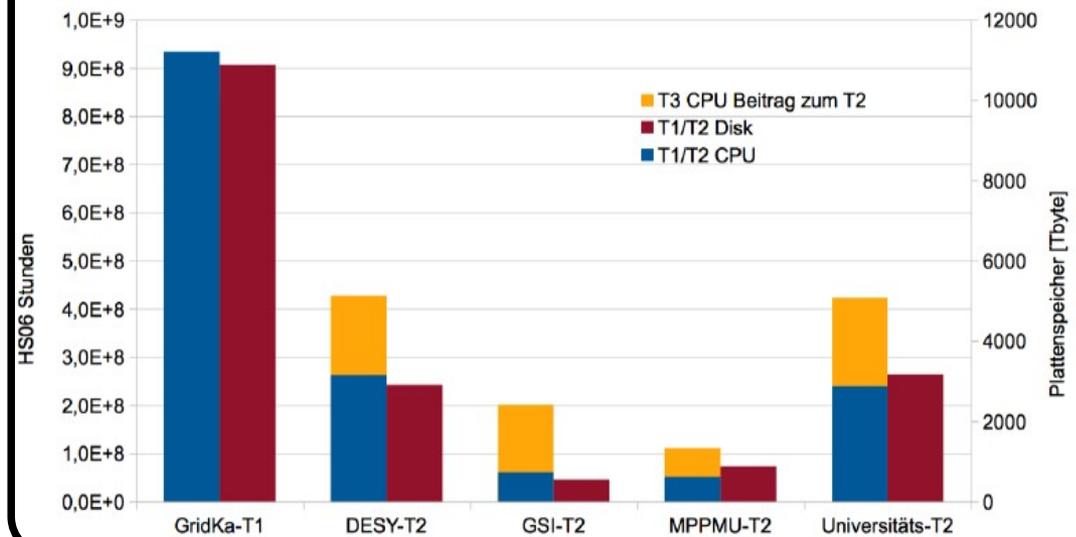
aus: Bericht des erweiterten Grid Project Boards der Helmholtz-Allianz an das KET, Okt. 2013

Anstieg der Ressourcen um Faktor 2,5 bis 2019

Situation in Deutschland

- Standorte:
 - Tier-1: KIT
 - Tier-2: DESY, GSI
MPI München, 5 Univ.
AC, FR, GÖ, M, W
- Deutscher Anteil am WLCG:
 - 15 % Tier-1
 - 10 % Tier-2
- Anteile in D:
 - 60% Tier-1
 - 40% Tier-2

Beitrag (CPU und Disk) der deutschen Standorte



GridPB und KET, Okt. 2013

Antragskonzept zur Finanzierung der Zentren an HGF-Instituten und MPG existiert.

ATLAS und CMS:
Tier-2 Anteil der Universitäten
ca. 45%

weitere Finanzierung
z. Zt. völlig offen

Detailinformationen

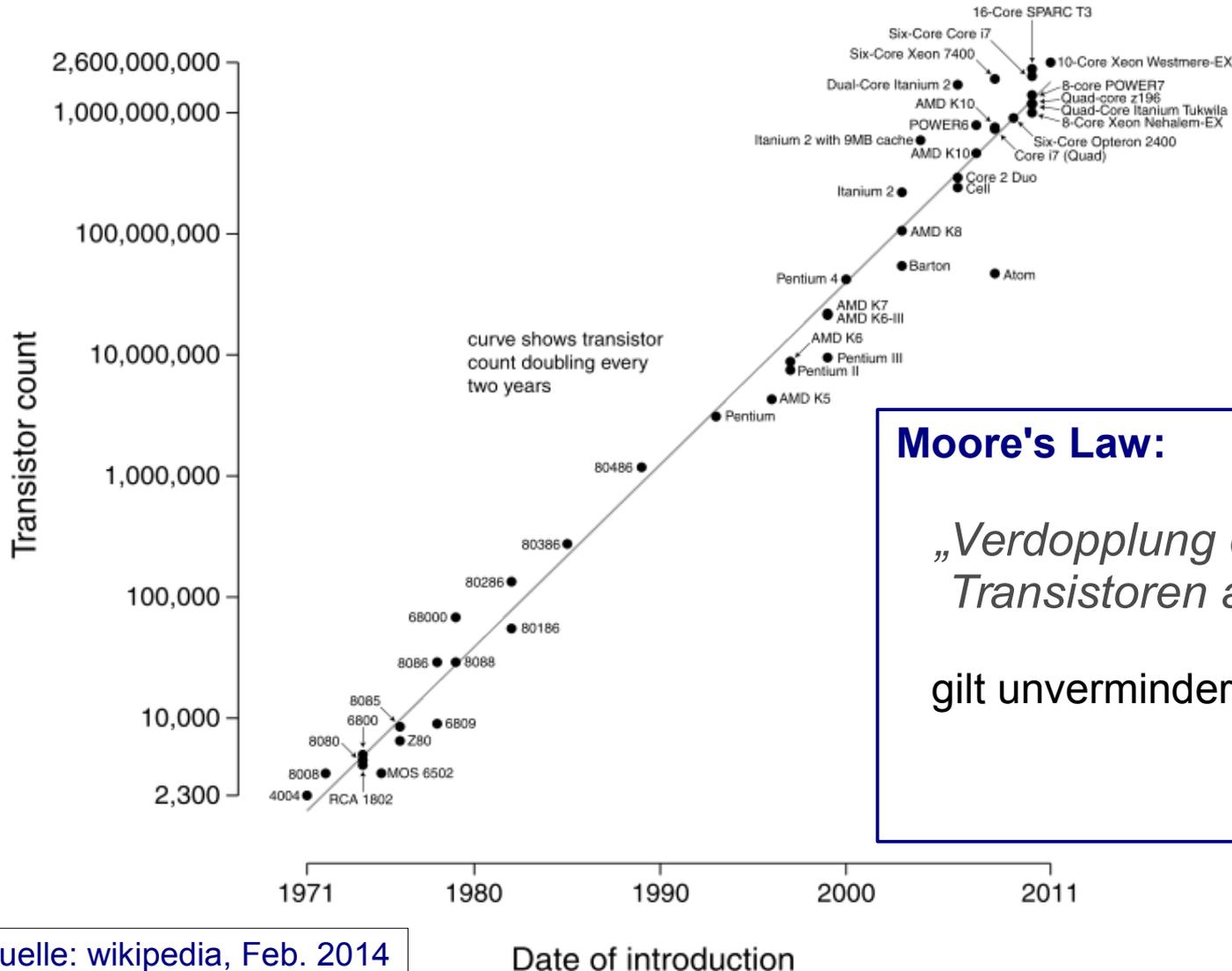
Bericht des erweiterten GRID Project Boards der Terascale Allianz im Auftrag des KET:
Computing in der Hochenergiephysik in Deutschland Bestandsaufnahme und Aussicht
Kurzfassung: *Computing in der Hochenergiephysik (Oktober 2013)*

HEP Code ist z.T. 15 Jahre,
Software der Experimente fast 10 Jahre alt

Was hat sich geändert ?

Moore's Law gilt weiter ... gut !

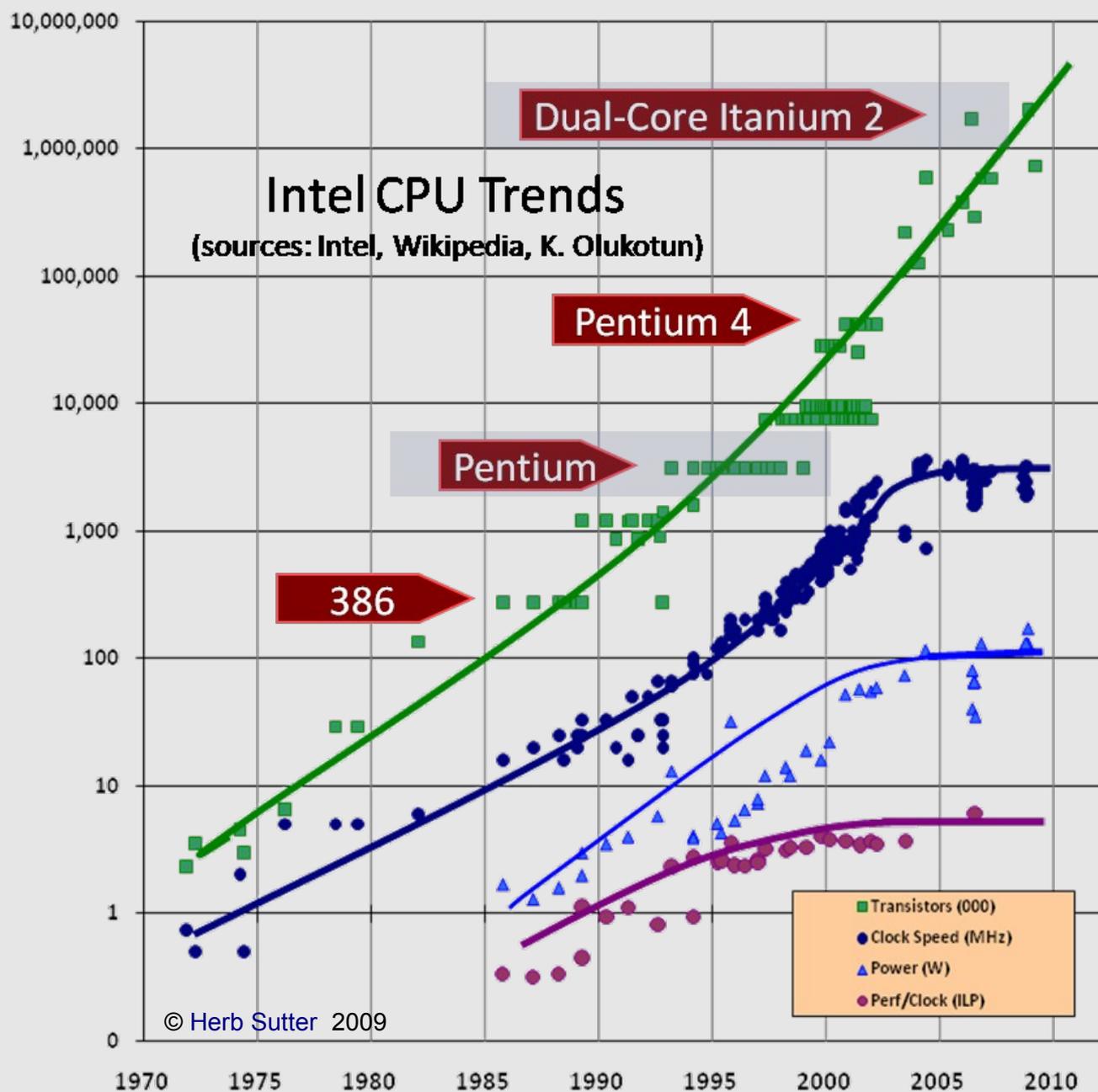
Microprocessor Transistor Counts 1971-2011 & Moore's Law



Quelle: wikipedia, Feb. 2014

aber ...

... eine neue Situation seit ~2005



Kleinere Strukturen
– führten früher zu
Steigerung der
Geschwindigkeit

heute: Grenze
der energetisch
vertretbaren Takt-
raten erreicht

- Steigerung der
Komplexität der
CPUs:
- Parallelisierung
 - Grafikeinheiten
- Multi-Core-
Architekturen

*große
Herausforderung
für die Software-
Entwicklung*

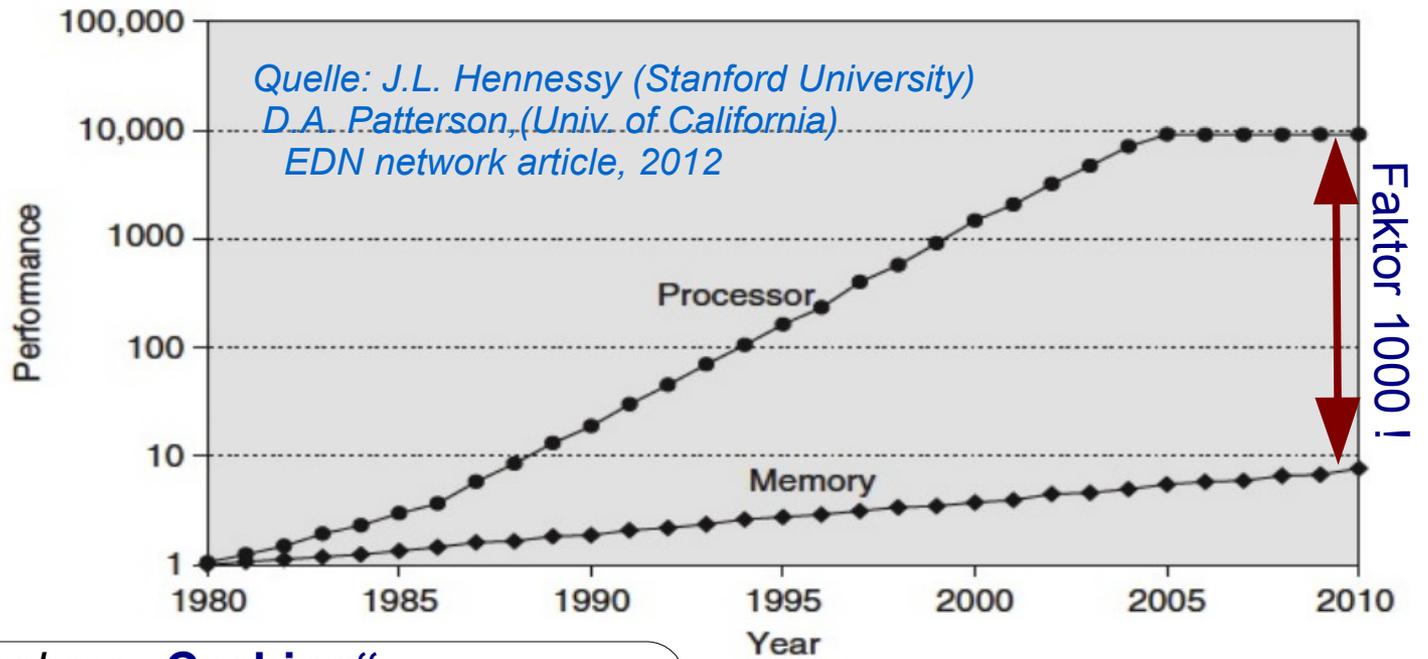
Entwicklungen der letzten Zeit

immer mehr Transistoren/Chip, aber seit 2005 keine signifikante Steigerung der Taktraten mehr – *„free Lunch is over“*

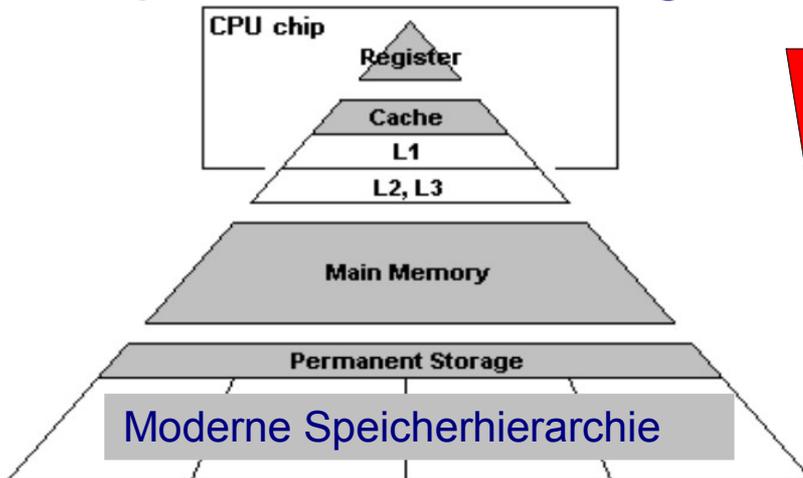
- komplexere Prozessor-Architekturen mit
- mehr/größeren Registern
 - größere, mehrstufige Cache-Speicher
 - Vektorisierung (**SIMD**=„**S**ingle **I**nstruction **M**ultiple **D**ata“) z. B. MMX, SSE, AVX
 - Multi-Core-Architekturen mit 2/4/6/8 ... Kernen pro CPU auch „Hyperthreading“ bei Intel-Architektur
 - Pipelining (d.h. parallele Ausführung) von Instruktionen
 - Sprung-Vorhersage
 - integrierte Grafik-Einheiten (GPU)

**Zunehmende Parallelisierung und Heterogenität der Architekturen:
Herausforderung für die Entwicklung effizienter Programmpakete**

Bandbreiten-Lücke zwischen Speicher und CPU



Gegenmaßnahme „Caching“



Cache enthält Kopie des gerade adressierten Speicherbereichs

Multi-Core-Architekturen:

Zugriff von mehreren Kernen auf den gleichen Speicher verschlimmern Situation !

Speicher pro Core ist in den letzten Jahren kaum noch gewachsen

Problem: Cache-Misses

einfaches Problem

```
...  
m_px+=px  
m_py+=py  
...
```

typische **Implementierung mit Polymorphismus** in C++,
d.h. Varianten der Methode `p4()` existieren in abgeleiteten Klassen

```
for (DaughterIt it=m_daughters.begin();  
     it!=m_daughters.end(); ++it){  
    m_p4.Add(it->p4() ); }  
}
```

jeweils benötigte Methoden werden dreimal über
Zeiger-Referenzen (`vtable` des Objekts) bestimmt;
Ziele liegen (wahrscheinlich) außerhalb des Cache !
→ CPU führt Wartezyklen aus (= tut nichts ...)

Sprünge im Code mit sehr wenigen Instruktionen dazwischen verschlimmern das Problem !

CPI-Index (**C**ycles **P**er **I**nstruction) von typischem HEP-Code zwischen 1,5 und 2,5
⇒ größter Teil der Leistungsfähigkeit moderner CPUs nicht ausgeschöpft,
>50% der möglichen Rechenleistung liegt brach!

Widerspruch zwischen objektorientierter Programmierung und Effizienz ?!
leider ja ...

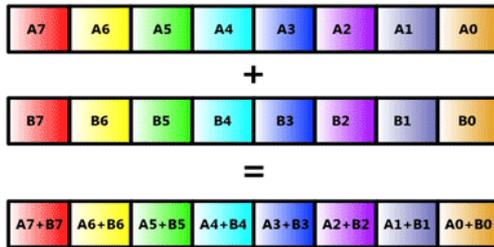
Wenn es darauf ankommt, besser „Daten-orientiert“ programmieren!
erfordert gewisses Verständnis von Prozessorarchitekturen ...

Vektorisierung

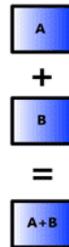
SIMD = Single Data Multiple Instructions

Quelle: Intel

SIMD Mode



Scalar Mode



Vektor-Befehle ermöglichen das parallele Prozessieren von 8 (AVX-256) bzw. 16 (AVX-512) reellen Zahlen mit einem Befehl.

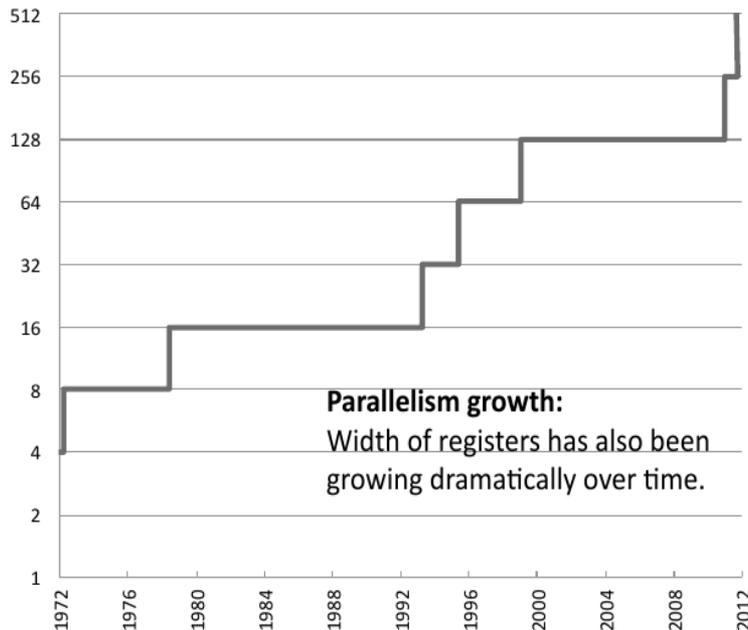
AVX-1024 schon auf den Road-Maps der Hersteller ...

Damit Vektorisierung funktioniert, müssen Daten im Speicher passend organisiert sein



an kritischen Stellen

Datenstrukturen anpassen !



gnu-c++ Compiler unterstützt seit langem „Auto-Vektorisierung“
(Option `-ftree-vectorize`)

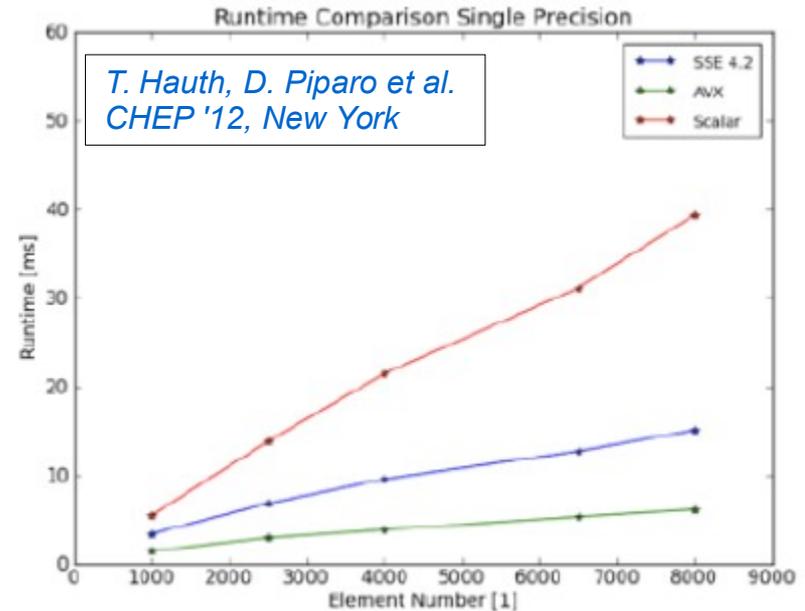
Nutzen von Vektorisierung

Code-Fragment zur Berechnung eines Polynoms

```
double* x = new double[ArraySize];
double* y = new double[ArraySize];

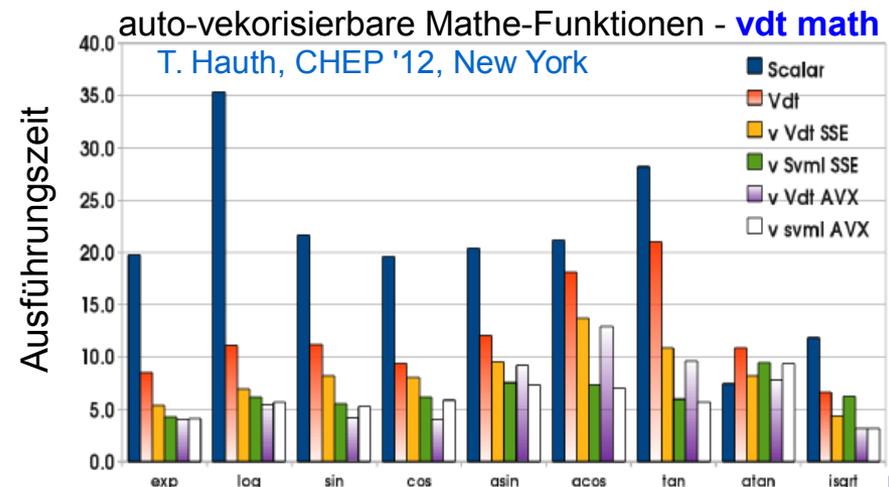
...

for (size_t j = 0; j < iterations ; j++)
{
    for ( size_t i = 0; i < ArraySize; ++ i)
    {
        // evaluate polynom
        y[i] = a_3 * ( x[i] * x[i] * x[i] )
              + a_2 * ( x[i] * x[i] )
              + a_1 * x[i] + a_0;
    }
}
```



Beispiel CMS 2011 → 2012:

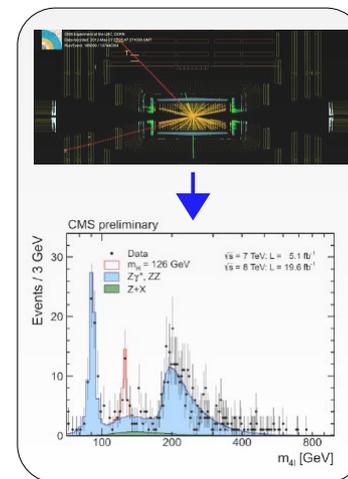
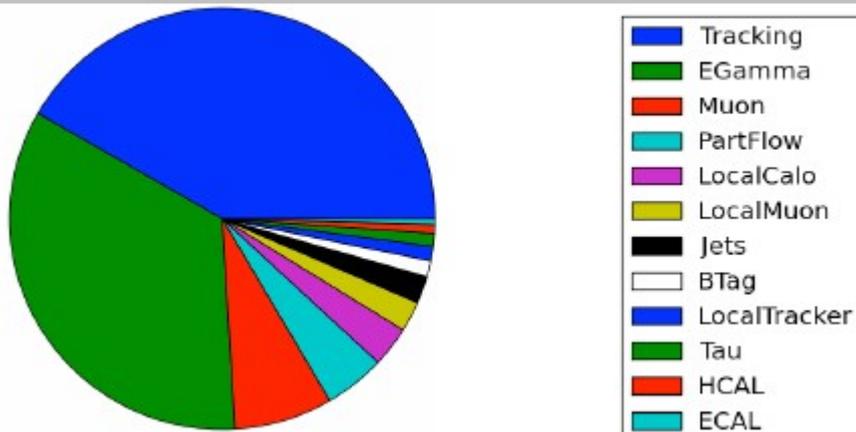
- neuer Compiler (*gcc 4.6*) mit neuem C++11-Standard, Auto-Vektorisierung und entspr. Code-Anpassungen
 - Verzicht auf einige „virtuelle Funktionen“
 - Umstieg auf schnellere MathLibs
 - Verbesserungen der Speicherverwaltung
 - sowie auch effizientere Algorithmen
- führten zu **~3 x schnellerer Rekonstruktion**



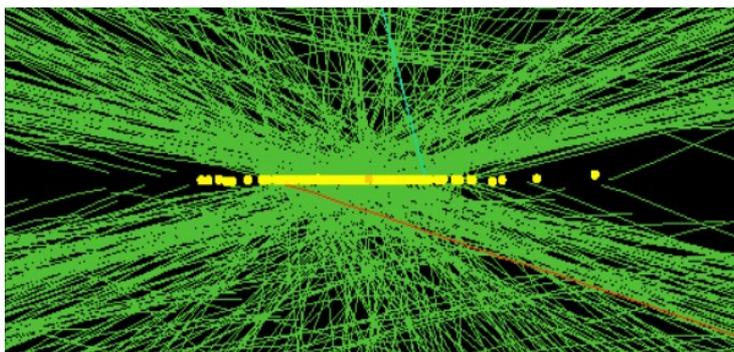
Beispiel: Ereignis-Rekonstruktion in CMS

Anteil CPU-Zeit für verschiedene Rekonstruktionschritte in Top-Paar-Ereignissen (CMS 2011)

CMS CR-2011-002

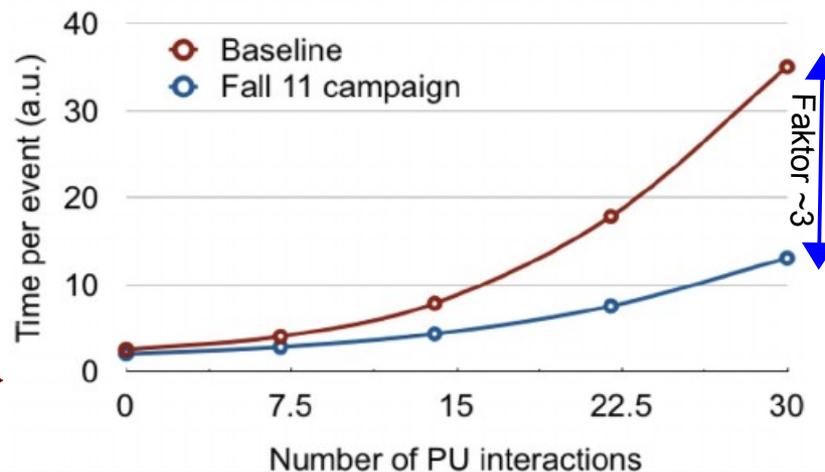


Viele überlagerte pp-Kollisionen („Pile-up“)



treiben Rekonstruktionszeit hoch →

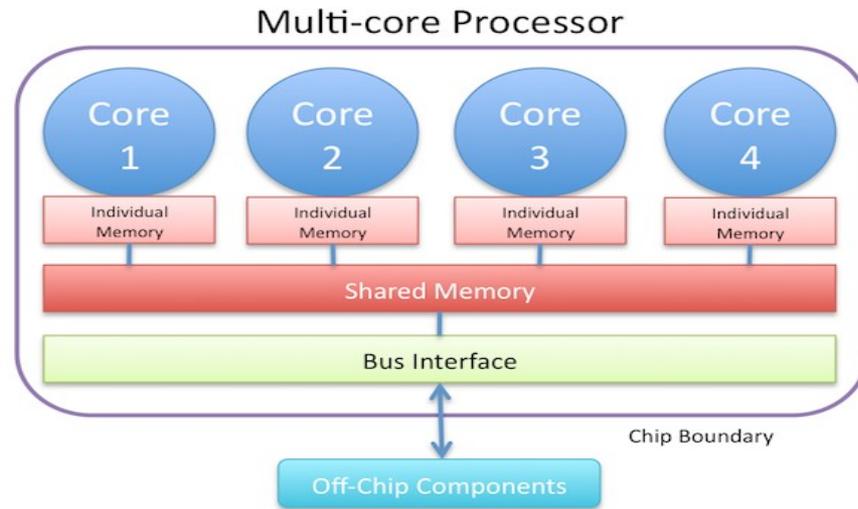
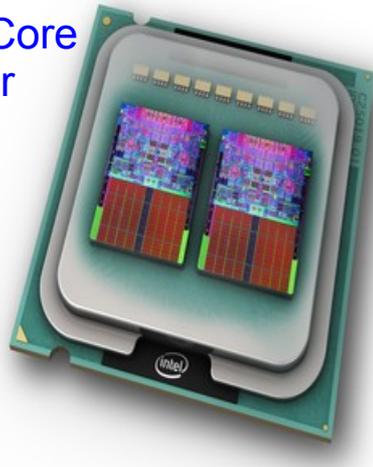
CPU-Zeit für QCD-Ereignisse (CMS 2011 & 2012)



~3x schnellere Rekonstruktion durch Optimierungen & neue Techniken

Herausforderungen auf Software-Seite

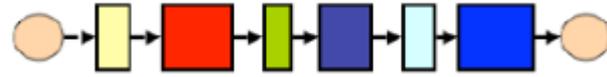
Intel Quad-Core
Prozessor



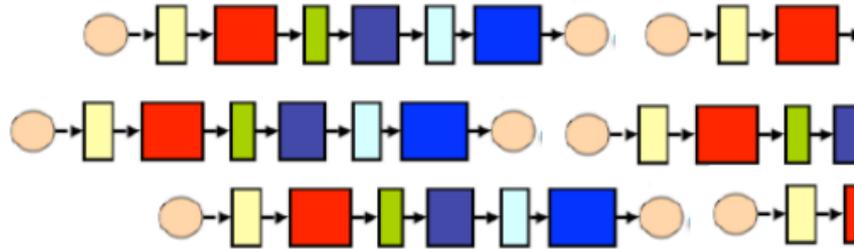
- Vektoreinheiten im Prozessor sind vorhanden und werden weiter anwachsen
→ **Vektorsierungsmöglichkeiten nutzen**
- Datentransfer zum Speicher wird immer mehr zum Flaschenhals
→ **effiziente Nutzung von (weiter wachsenden) Caches**
- Zahl der Cores pro Chip steigt weiter an (z. Zt. 12-cores in Server-CPU's)
→ **Parallelisierung der Arbeitsabläufe, Multi-Threading**
- Chips mit integriertem Grafik-Prozessor im Kommen
→ **Einsatzmöglichkeiten für GPUs evaluieren**
- energieeffiziente Prozessoren (aus dem Mobil-Bereich) in Servern
→ **Code für heterogene Architekturen**

Stufen der Parallelisierung

sequentielle Ausführung
einer Folge von Algorithmen



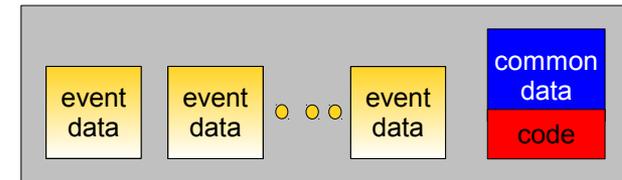
sequentielle Ausführung,
unabhängig auf mehreren
Prozessor-Kernen



triviale
Parallelisierung
(was wir immer
schon machen)

Falls dies auf dem gleichen Knoten geschieht:
Code und Speicher für gemeinsame Daten
(Geometrie, Kalibration etc.) nur einmal !

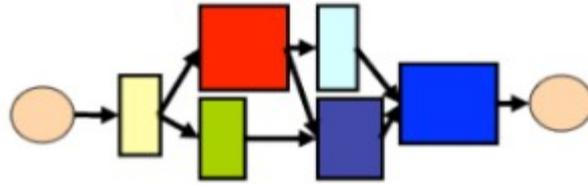
⇒ **Speicherersparnis**



- LHC-Experimente führen z. Zt. „**Multi-Core / Whole-Node Scheduling**“ auf den T1 ein

Stufen der Parallelisierung (2)

Parallele Ausführung
einiger Algorithmen

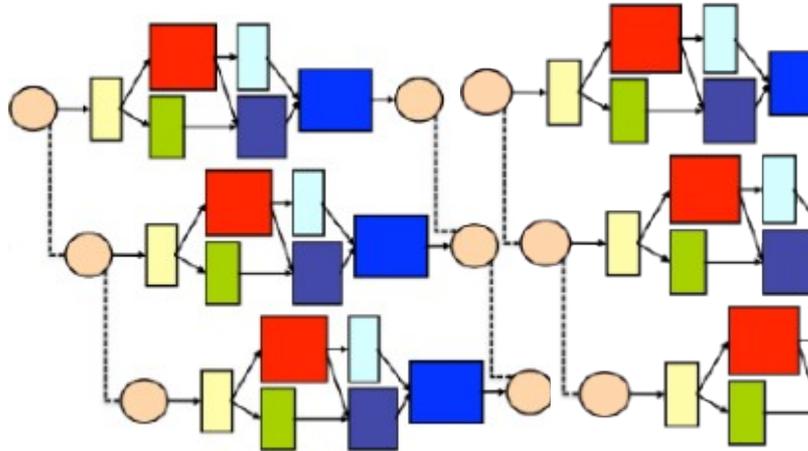


„Multi-Threading“

Noch viel Arbeit am HEP-Code notwendig,
„Thread-Safety“ nicht einfach zu gewährleisten

Fernziel:

Parallelität
auf
allen
Ebenen



Optimale Ausnutzung
der Leistungsfähigkeit
der Prozessoren

- entsprechende „Parallel Frameworks“ sind in Arbeit
concurrency framework project („CF4Hep“) GaudiHive bei LHCb/ATLAS
(s. <http://concurrency.web.cern.ch/GaudiHive>) und CMSSW 7 bei CMS

Evt. Massiv Parallel: Multi-Event-Algorithmen

z.B. Tracking, auf Grafik-Prozessoren auslagern

Rechnen auf Grafik-Prozessoren (GPUs)

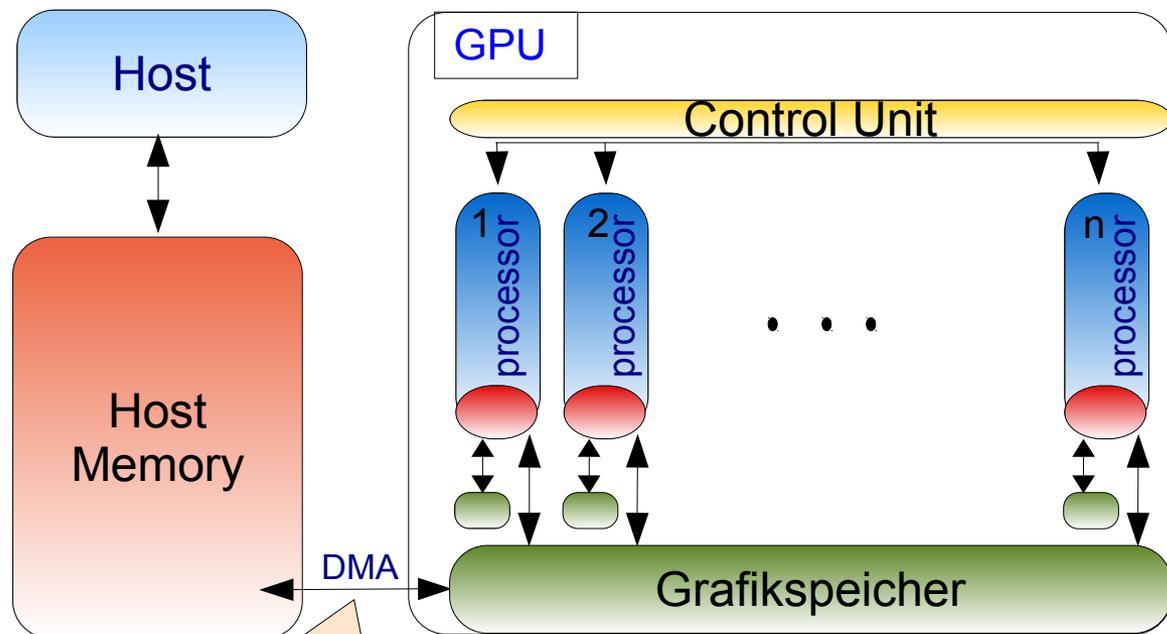
typische Leistungsdaten

CPU:

Core i7-3930K
(6 cores, 3.20GHz)
500 EUR, 150 GFlop
1.2 GFlop /W

GPU:

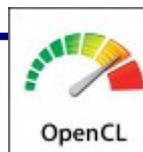
GeForce GTX 660
250 EUR, 1880 GFlop
13.4 GFlop/W



für „kleine“ Probleme wird dieser Datentransfer zum Flaschenhals

Programmierung z.B. in OpenCL (Obermenge des C99-Standards)

- unterstützt auch Multi-Core CPUs (ab SSE3)
 - Herstellerspezifische Implementierung für NVIDIA-GPUs: CUDA
- „Hardware- und Datenorientierte“ Programmieretechniken erforderlich**

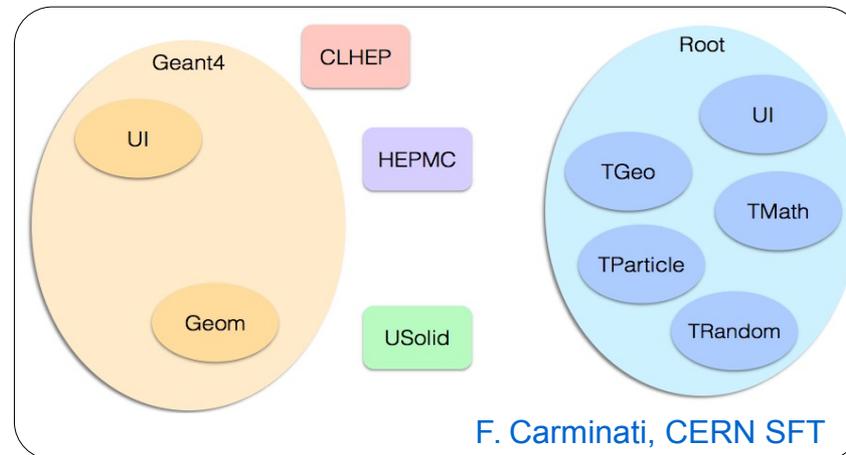


- GPUs sind hochinteressant im Trigger- und Online-Bereich
zahlreiche Projekte, z. B. im Bereich Online-Tracking
- ob sich GPUs im „Offline-Bereich“ durchsetzen, ist derzeit nicht klar

HEP-Software: Common Tools

- Millionen Zeilen C++ Code - z.T. 10-15 Jahre alt & Tausende von Bibliotheken
- wenige Instruktionen zwischen Sprüngen durch Funktionsaufrufe und virtuelle Methoden
 - hebt Instruction-Pipelining im Prozessor aus
 - führt zu exzessiv vielen Cache-Misses

Missbrauch der Möglichkeiten von C++ ???



- Von zentraler Bedeutung:
ROOT & GEANT

– zwei getrennte Systeme ohne gemeinsame Koordinierung mit vielen Duplizierungen

Potential für Verbesserungen mit Nutzen für gesamte HEP

Die Aufgabe: Potenzial moderner Architekturen ausreizen

- Performance-Schwachstellen in vorhandenem Code finden und beheben
Reengineering von Frameworks, Datenstrukturen und Algorithmen
- Möglichkeiten für „concurrency“ (=gleichzeitige, unabhängige Ausführung) in Problem finden
Ian Bird: *„HEP must develop expertise in concurrent programming“*

- „parallel Denken“ schon in der Design-Phase
 - Algorithmen und Datenstrukturen parallelisierbar auslegen
 - Hardware-nahe, Daten-orientierte Programmierung zeitkritischer Algorithmen
 - neue Möglichkeiten (insb. C++11) nutzen

Müssen effiziente Datenprozessierung & OO richtig kombinieren lernen

(auch im Hinblick auf HL-LHC, Belle II, ILC)

an jüngere Kollegen und Doktoranden:

Schulungsangebote

(z. B. GridKa Computing School oder C++ School der Terascale Allianz) nutzen

„Bessere Physik mit besserer Software“

!!! Eine interessante Herausforderung für jede/n mit Interesse am Programmieren !!!

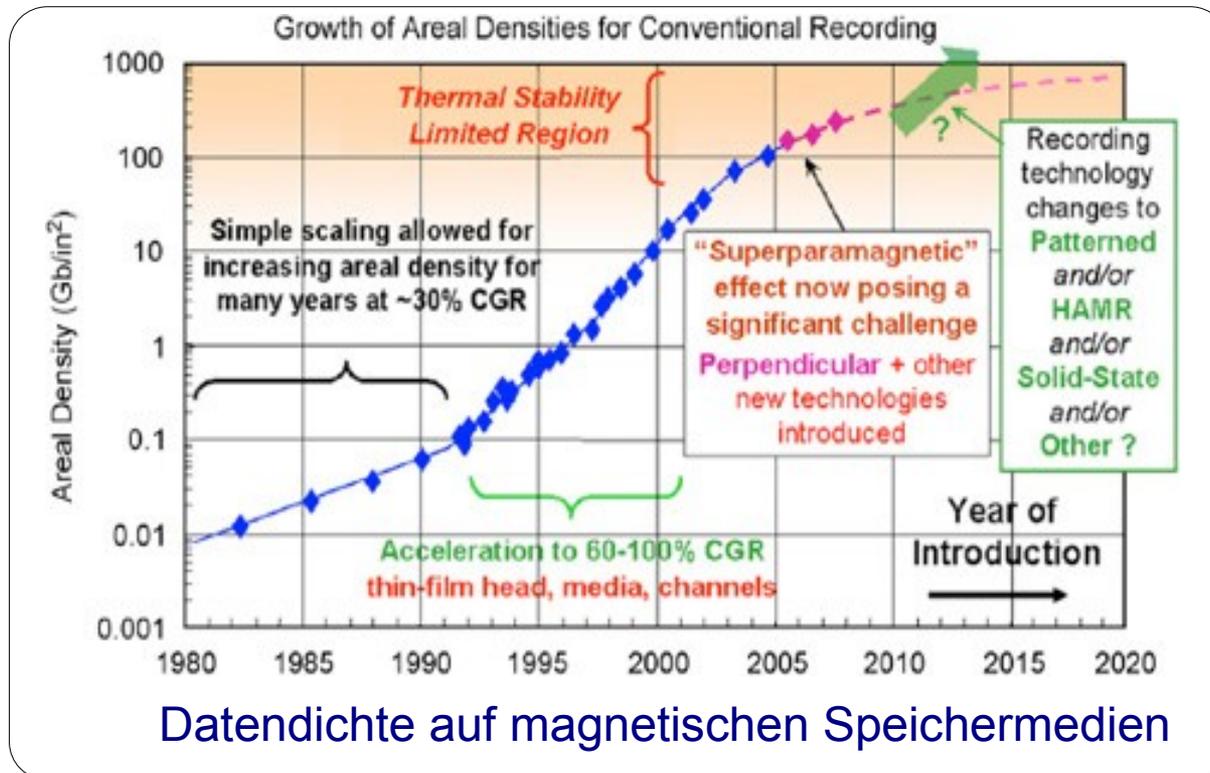
Programmieren ist das „Löten“ des 21. Jahrhunderts

**HEP (auch in D) droht Anschluss an Stand der Softwareentwicklung zu verlieren
ähnlich ernste Situation wie Ende der 90er (als der Umstieg von FORTRAN bewältigt wurde)**

HEP Computing:

Laufende Aktivitäten und Zukunft

Massenspeicher



Quelle: Journal of Magnetism and Materials, „Future Hard Disk Drive Systems“, Roger Wood

Datendichte auf magnetischen Speichermedien

Zunahme verlangsamt sich seit einigen Jahren signifikant

Preis / GB fällt weiter exponentiell, aber Kapazität / Platte steigt kaum noch

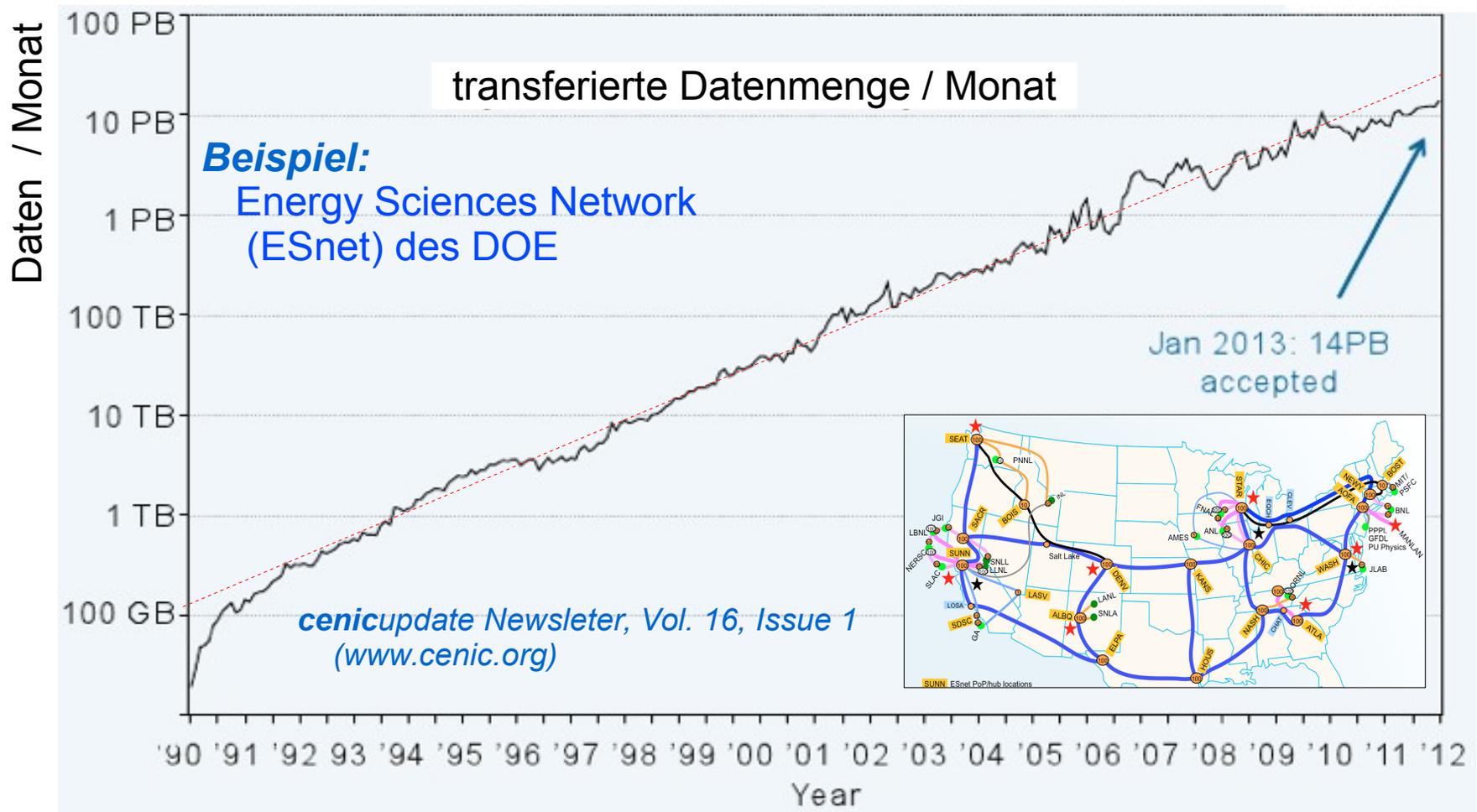
→ Speichersysteme werden größer, damit steigen Aufwand für Betrieb & Wartung sowie Energiebedarf

Plattenspeicher ist heute schon die teuerste Komponente in T1 / T2-Zentren

Verhältnis der Kosten von CPU und Disk wird sich weiter in Richtung Disk verschieben

s. auch T35.6 M. Fischer, „Datenmanagement für hochperformante Analysen“
T12.3 G. Duckeck, „Hadoop for Parallel Root Analysis“

Netzwerkverbindungen



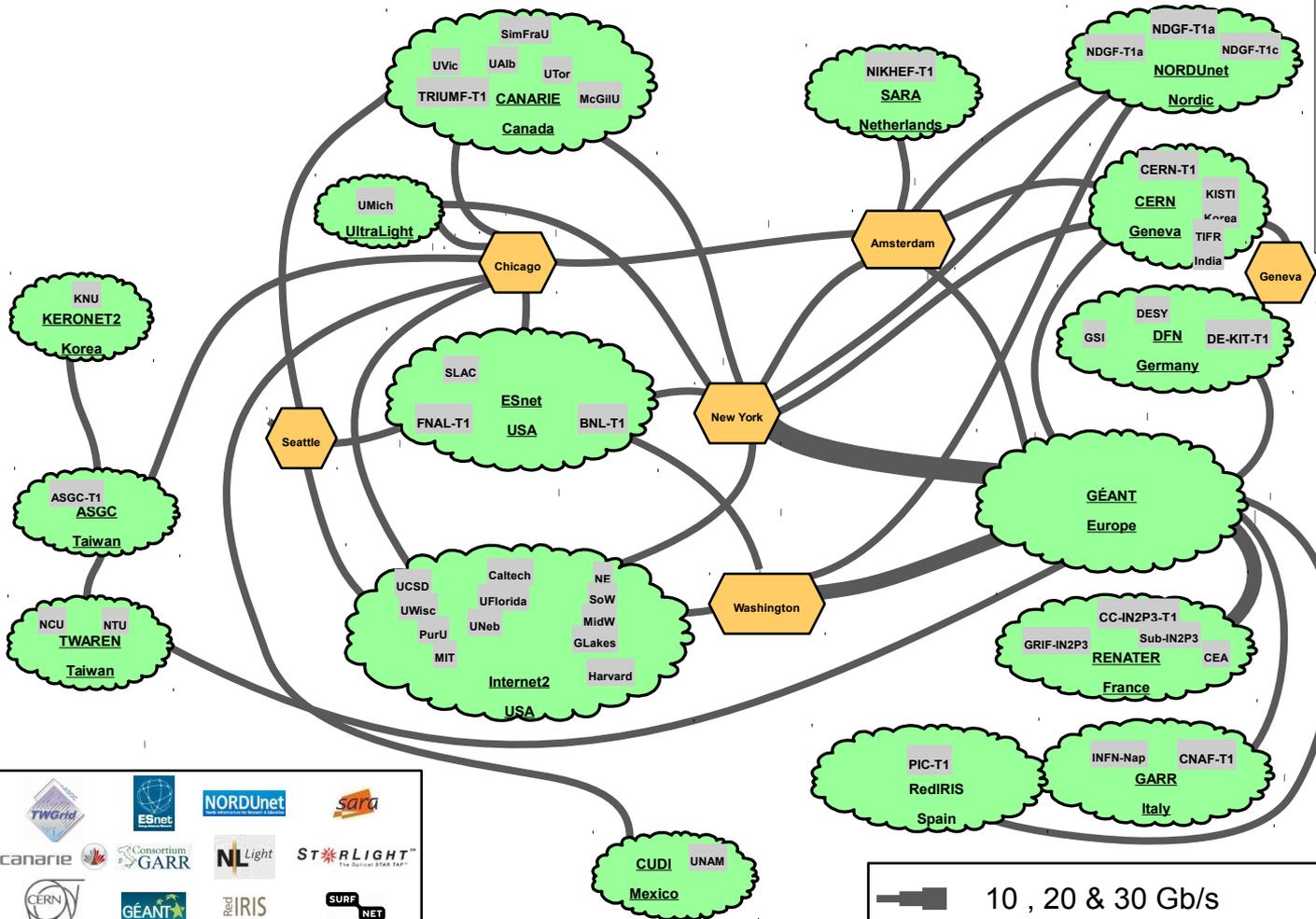
- exponentielle Zunahme mit Faktor $\sim 1,8 / a$ in den letzten 20 Jahren
- **Netzbetreiber erwarten keine Trendänderung bis mindestens 2017**

Netzwerk (2)

Schon lange: **LHC Optical Private Network** für T0↔T1, T1↔T1

neu: LHCOne für Transfers T1 ↔ T2 & T2 ↔ T2
relevant für End-Datenanalyse

- **Privates Netzwerk** auf dem Internet
 - nur vertrauenswürdige Partner, keine Firewalls notwendig
- Trennt LHC Datenverkehr vom allgemeinen Wissenschaftsnetz
- Management durch regionale / nationale Netzbetreiber



Deutscher Anteil an den Kosten (ca. 130k€/a) bisher von der Helmholtz-Allianz „Physik an der Teraskala“ getragen.



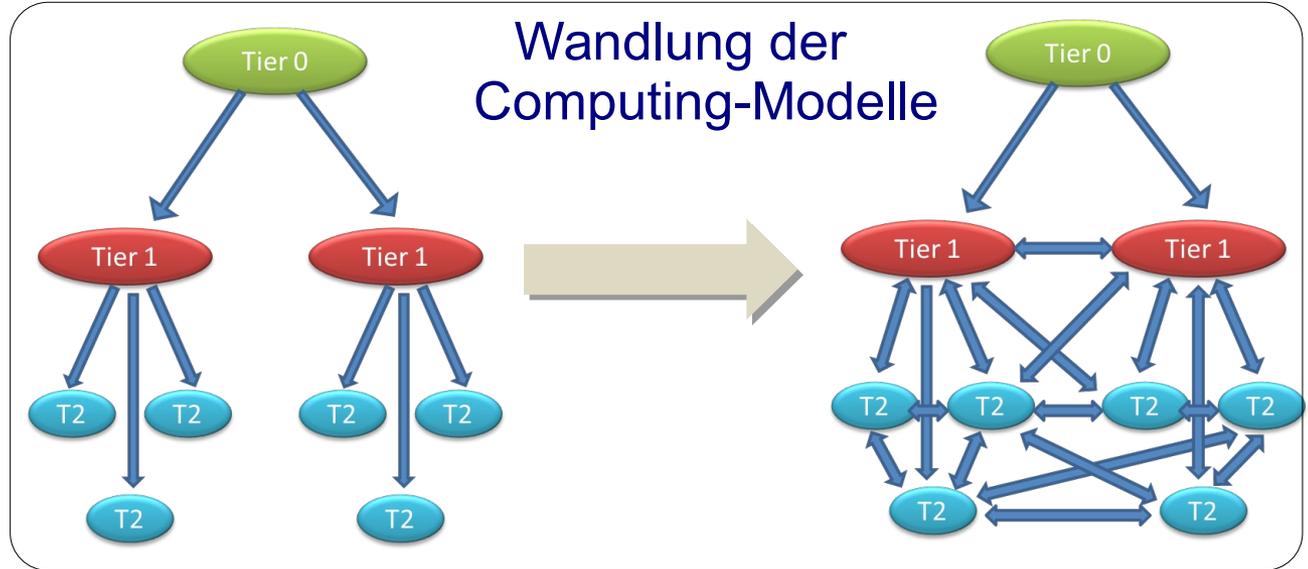
<http://lhcone.net>

Weiterentwicklung der Computing-Modelle

wachsende
Netzwerk-
bandbreite



&



ermöglichen

Datenprozessierung über Netzwerk

„Transferieren ist billiger als Speichern“

„Datenföderation“: T1, T2 (& T3) als einheitliches, verteiltes Speichersystem

basiert auf (HEP-spezifischem)



XRooT

ATLAS: „FAX“ (Federating ATLAS Storage Systems using xrootd)

CMS: „AAA“ (Any data Anytime Anywhere)

übrigens: schon lange bei ALICE erprobt !

100 GB/s – Technik im Einsatz

2x100 GB/s Links

CERN ↔ Wigner Data Centre (Budapest)
als Erweiterung des CERN T0 in Produktion



Erwarten weitere 100 GB/s-Netze, auch in Deutschland:

- Verbindung zwischen HGF-Zentren
- einige Ländernetze (BelWü: KIT ↔ Heidelberg schon im Testbetrieb ...)

Ressourcen außerhalb des WLCG - Heterogene Hardware

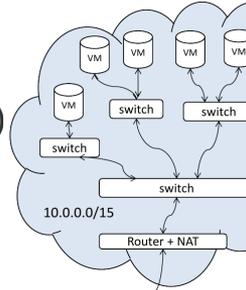
- Cloud-Ressourcen

- privat (z.B. Instituts-Cluster)
- kommerziell (Amazon, Google, Vodafone ...)

gegen nutzungsabhängige Abrechnung

hat R&D-Phase längst verlassen

→ „Grid of Clouds“ ist Realität



z.B. Open Stack



Beispiel: Nutzung der ATLAS & CMS HLT-Farmen während des LS 1

- HPC – Cluster

- viele HEP-Applikationen laufen
- MPI-Interfaces und SAN für HEP nicht benötigt

→ ein teuer Weg !



z.B. Super-MUC am LRZ in München, SDCC b.CMS oder zukünftig bwHPC Cluster (ATLAS/CMS/LHCb)

- keine Grid-Dienste/Authentifizierung
- schnelle, aber kleine und teure Disk
- oft geringe WAN-Bandbreite
- unterschiedliche „Site Policies“

Spezielle Lösung für jeden Einzelfall notwendig → Personal !

- andere Prozessor-Architekturen (ARM)

Ziel: **Optimierung von Zyklen / Watt**

für spezielle Anwendungen

Tests in allen Experimenten

- viele Applikationen laufen auf ARM

erste Server mit ARM-Architektur angekündigt

Ist das die Zukunft ?

Test-Cluster mit 16 ARM-Prozessoren und SATA-Platten am KIT (SCC)

seit 2012 wurden mehr Mobil- als X-86-Prozessoren verkauft

Vorbereitungen auf LHC-Run 2

Alle Experimente erwarten **Ressourcen-limitierten Betrieb** der Computing - Infrastruktur ab 2015 durch **höhere Ereignis-Raten** (1kHz Trigger) und **Pile-up**

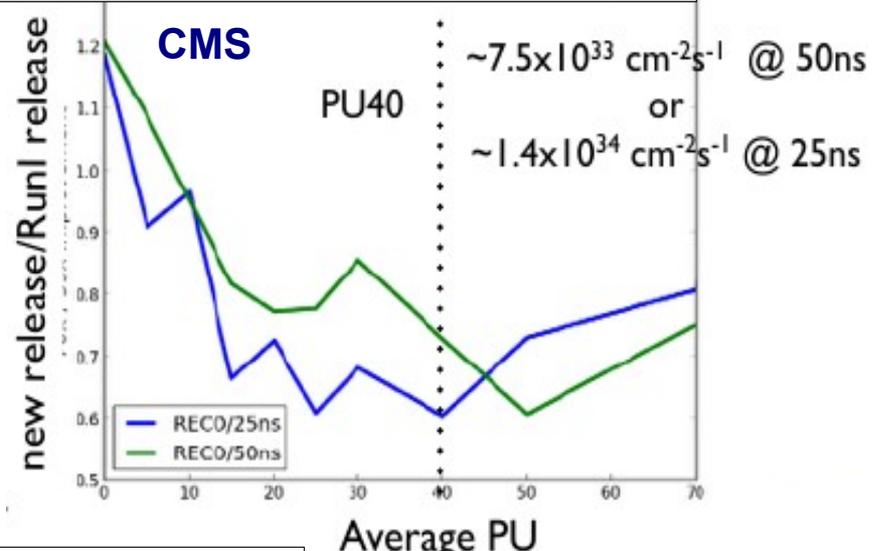
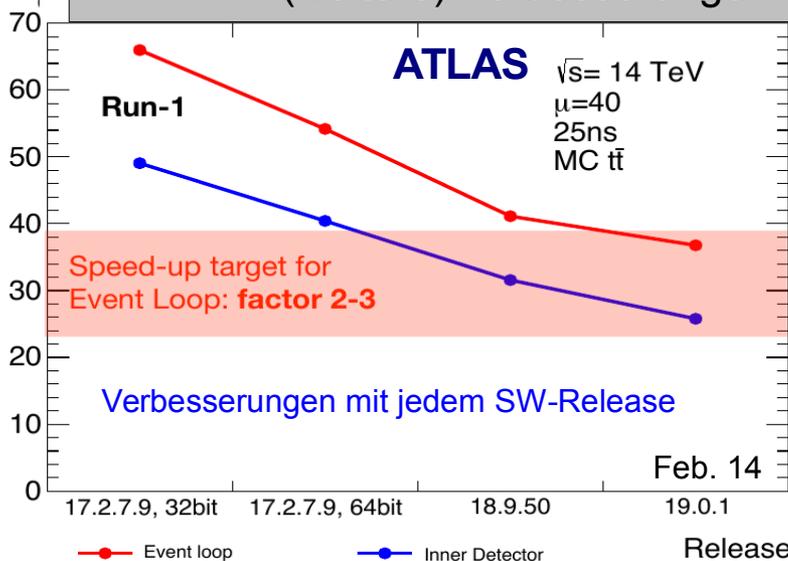
Ziel: optimale Nutzung aller Ressourcen

„No Room for Mistakes“

- Weitere, intensive Anstrengungen zur
 - Beschleunigung der Rekonstruktionssoftware
 - Reduktion des Speicherbedarfs



(weitere) Verbesserungen der Rekonstruktions-Software



Quelle: LHCC, März 2014

Vorbereitungen auf LHC-Run 2 (2)

- z. Zt. Einführung von **Multi-Core / Whole Node – Queues** an T1-Zentren
längerfristig auch Parallelisierung von Algorithmen / Modulen vorgesehen
- **Datenföderationen** brechen Hierarchie T1 / 2 / 3 auf
 - – Flexibilität zur Nutzung freier CPUs
 - Shared-Workflows zwischen Sites
 - T3-Zentren ohne Platten, auch als opportunistische Ressource

s.auch T35.7 F. Legger, „Tests of the Federated ATLAS XrootD system with Hammer Cloud“

bei CMS z. Zt. Disk / Tape-Trennung bei Tier-1 als Voraussetzung

Ziel CMS: 20% der Datenzugriffe über WAN bei 200k Jobs pro Tag

- Verbesserungen von **Workflows** und **Datenformaten**
 - effektive Nutzung vorhandener Ressourcen

Probelaufe im Sommer für Test und Validierung vorgesehen

→ **Next-Generation-Computing für die LHC - Datenanalyse**
auf dem Weg, mit neuen Herausforderungen an die Computing - Teams !

Nutzung „Opportunistischer Ressourcen“

Zahlreiche Beispiele:

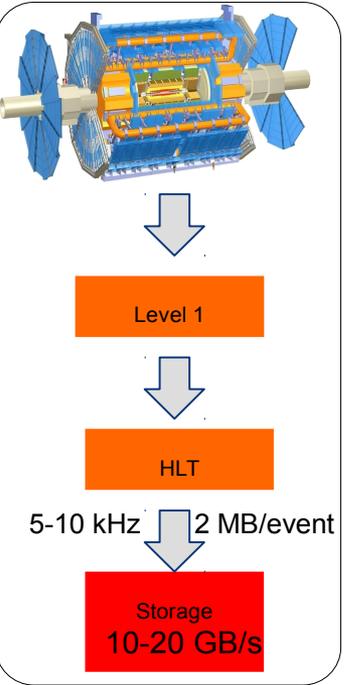
- Nutzung der HLT-Farmen von ATLAS und CMS über OpenStack sowie Instituts-Clouds über OpenStack bzw. OpenNebula
- Nutzung von Ressourcen an HPC-Zentren (CMS, ATLAS) getestet
s. auch T12.1 R. Walker, „HPC usage for ATLAS“
- Evaluation kommerzieller Clouds (Amazon, Google, Vodafone, ...)

WLCG + Grid of Clouds

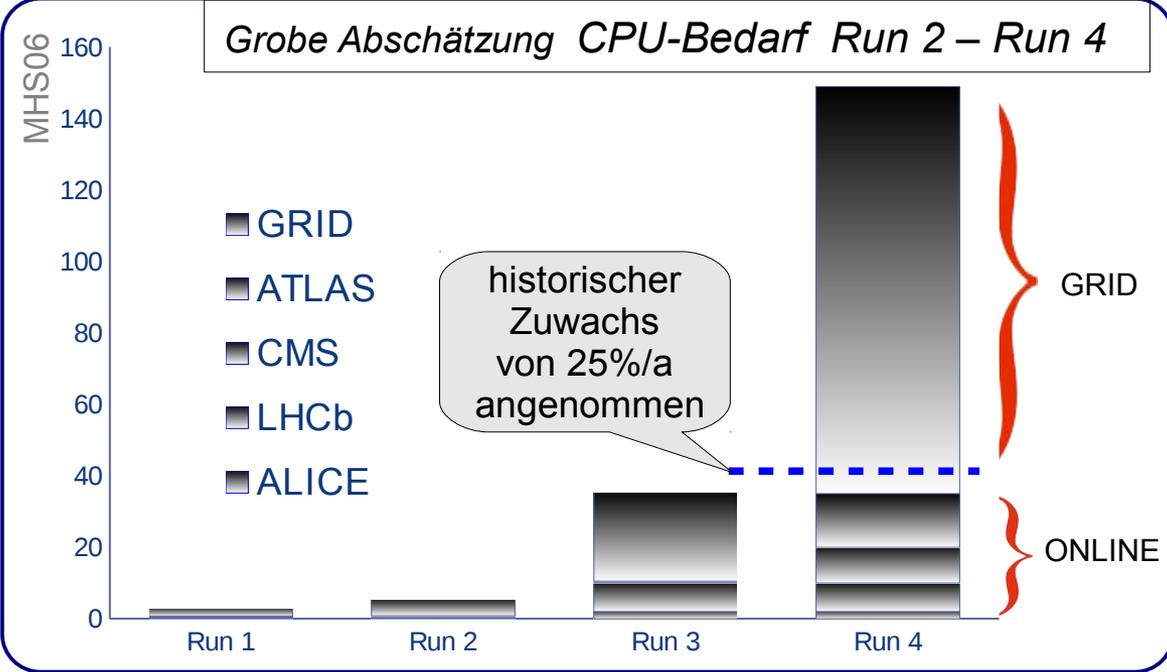
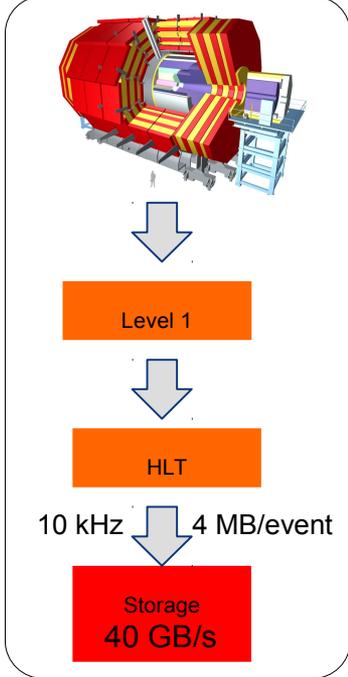
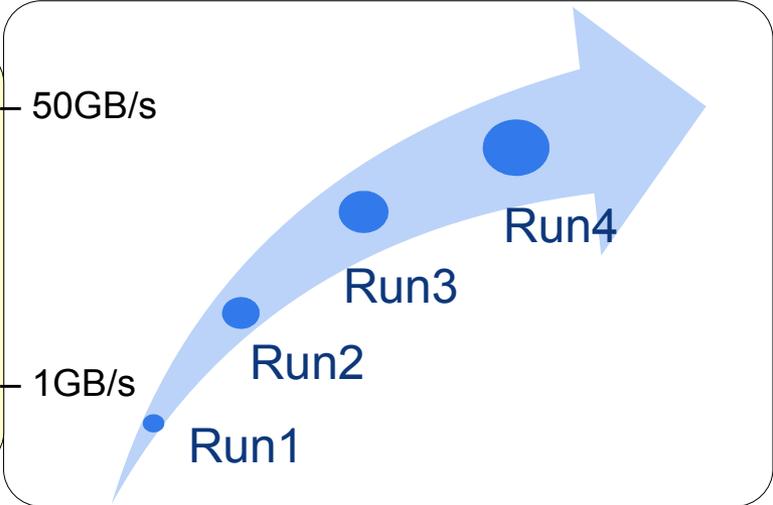
Untersuchung alternativer Architekturen

- LHCb Software läuft auf ARM-Architektur
~10x langsamer lt. HEPSpec, in der Praxis Faktor 5-7 im Vergl. zu X86
- LHCb - Track-Rekonstruktion auf GPUs getestet
overhead durch Datentransfers evtl. kompensierbar durch gleichzeitiges Prozessieren mehrerer Ereignisse
- auch Teile der Spurrekonstruktion in CMS (CHEP '13, D. Funke et al.) und ATLAS (Wuppertal mit FH Münster) auf GPUs evaluiert

(fernere) Zukunft der HEP-Software



ATLAS & CMS rechnen mit ~50x höherer Datenrate in Run 4



Quelle: Ian Bird (WLCG)

• **Bedarf** durch komplexere Detektoren und höhere Ereignisraten in Run 4 **4x höher** als durch technologische Weiterentwicklung erwarteter **Ressourcenzuwachs**

Schlussfolgerungen

- **LHC-Grid Computing war und ist sehr erfolgreich**

steht aber vor enormen Herausforderungen durch

- stark steigende Anforderungen (Komplexität und Rate der Ereignisse)
- neue technologische Entwicklungen

- HEP-Software ist (größtenteils) fast 15 Jahre alt und muss an Hardware-Entwicklungen und neue Standards (C++11) angepasst werden

„Ohne Invest in Software werden unsere Grid-Zentren zu teuren Heizkraftwerken“

(B. Hegner, CERN)

- Paradigmenwechsel ähnlich wichtig wie Ende der 90er:

Vektorisierung & Parallelisierung von Algorithmen und Frameworks,
neue Architekturen und Nutzung von (Grafik-) Coprozessoren

Viel Raum für Kreativität, neue Ideen und originelle Beiträge !

- Computing-Anforderungen von LHC Run 3 und 4 liegen weit jenseits der Extrapolation aufgrund der Wachstumsraten der letzten Jahre

Detektor-Upgrades erfordern ebenso Anstrengungen bei Software & Computing

- Expertise in der Teilchenphysik-Community und außerhalb sollte gebündelt werden
Vorschlag einer „**HEP Software Collaboration**“

Einladung von Ian Bird und John Harvey (i. A. von Sergio Bertolucci, CERN Director of Research)

erstes Meeting am 3. - 4. April '14 am CERN

Dank allen Kollegen des WLCG, auf deren Material dieser Vortrag beruht,
und ganz besonders an G. Duckeck, J. Elmshäuser, B. Hegner, D. Piparo, C. Wissing

ENDE

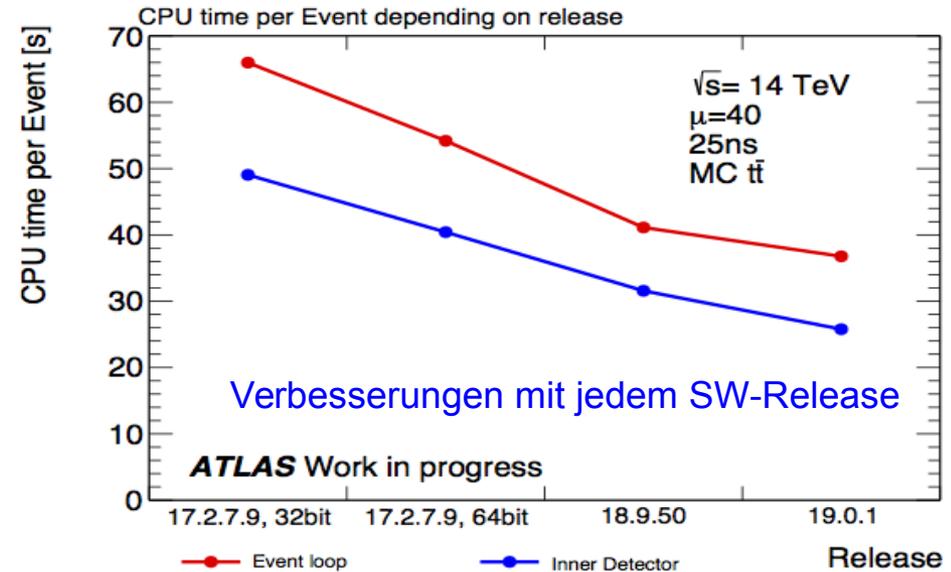
Materialsammlung

LS1: Planungen ATLAS

- ~3-fache Beschleunigung des Codes als Ziel im LS 1

- AthenaMP mit Multi-Processing und verringertem Speicherbedarf

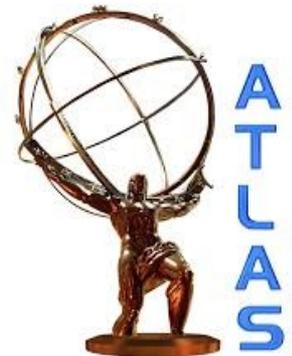
- Langfristiges Ziel:
Multi-threaded Framework
GaudiHive mit Parallelisierung auf Ebene von Ereignissen und Algorithmen



- Nutzung von Cloud-Ressourcen hat R&D-Phase verlassen
(ATLAS HLT Farm, BNL Cloud, Kommerzielle Clouds Amazon & Google, ...)

→ WLCG + „Grid of Clouds“

- Weiterentwicklung des Workflow- und Job-Management Systems
- Datenföderation mit XRootD
- Exploration zur Nutzung von HPC-Ressourcen ist jetzt koordinierte ATLAS-Aktivität



LS1: Planungen CMS



Ressourcen-limitierter Betrieb ab 2015 durch höhere Trigger-Rate (1kHz) und Pile-up

Ziel: optimale Nutzung aller Ressourcen

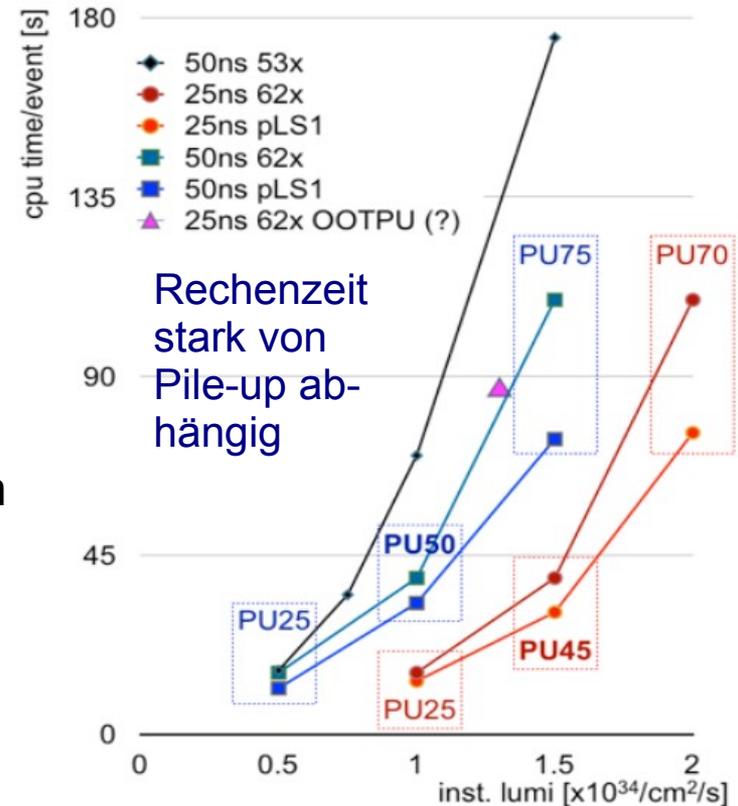
„No Room for Mistakes“

- Weitere Anstrengungen zur **Beschleunigung der Rekonstruktionssoftware**, Reduktion des Speicherbedarfs, Unterstützung für Multi-Core / Whole Node - Queues
- Disk / Tape-Trennung bei T1 und **Datenföderation** bricht Hierarchie T1 / 2 / 3 auf
 - – Flexibilität zur Nutzung freier CPUs
 - Shared-Workflows zwischen Sites
 - T3-Zentren ohne Platten, auch als opportunistische Ressource

Ziel: 20% der Datenzugriffe über WAN bei 200k Jobs pro Tag

- Nutzbarmachung von „**Opportunistic Resources**“ über Cloud-Interface fester Bestandteil der Planungen incl. Kommerzieller Ressourcen außerhalb der normalen Arbeitszeit („Vodafone“)

Ziel: 8'000 Cores/Jahr zusätzlich



LS1: Planungen LHCb



- Multi-Core/Whole-Node Scheduling in Vorbereitung
- Nutzung von Cloud-Ressourcen jenseits von Testbetrieb:
CERN OpenStack, OpenNebula am PIC, RAL, ...
- keine revolutionären Änderungen (Gaudi u. DIRAC) für Run 2
- Computing-Modell enthält nun auch Tier2 – als „Tier1 ohne Tape“
- Entwicklung des Multi-Threading Frameworks Gaudi
(u.a. auch von ATLAS verwendet)

Beiträge auf dieser Tagung

T12.1 R. Walker, „HPC usage for ATLAS“

T12.3 G. Duckeck, „Hadoop for Parallel Root Analysis“

T35.6 M. Fischer, „Datenmanagement für hochperformante Analysen“

T35.7 F. Legger, „Tests of the Federated ATLAS XrootD system with Hammer Cloud“

C++11 ISO-Standard seit Aug. 2011

„provide proper solutions for real-world problems“

- löst C++03 aus dem Jahr 2003 ab
- Kompatibilität mit C++98 weitgehend erhalten
- zahlreiche Sprachzusätze und Erweiterungen
- Unterstützung für Multi-Threading

C++14 steht vor der Tür