AWSAC: Amazon Web Services for ATLAS Computing

Jan-Philip Gehrcke

Universität Würzburg und MPI für Physik München

17. Oktober 2008

Übersicht

Was versteht man unter ATLAS Computing (AC)?

Rekonstruktion

Softwareanalyse von Messdaten ($O(\frac{PB}{Jahr})$) zur Rekonstruktion von Ereignissen

- produziert auf das Wesentliche reduzierte Daten

Simulation

Simulation der Detektorantwort auf vorgegebene Ereignisse

- kalibrieren einzelner Detektorkomponenten mit Simulationsergebnissen
- optimieren von Rekonstruktionsalgorithmen

Was versteht man unter ATLAS Computing (AC)?

Rekonstruktion

Softwareanalyse von Messdaten ($O(\frac{PB}{Jahr})$) zur Rekonstruktion von Ereignissen

- produziert auf das Wesentliche reduzierte Daten

Simulation

Simulation der Detektorantwort auf vorgegebene Ereignisse

- kalibrieren einzelner Detektorkomponenten mit Simulationsergebnissen
- optimieren von Rekonstruktionsalgorithmen

Was versteht man unter ATLAS Computing (AC)?

Rekonstruktion

Softwareanalyse von Messdaten ($O(\frac{PB}{Jahr})$) zur Rekonstruktion von Ereignissen

- produziert auf das Wesentliche reduzierte Daten

Simulation

Simulation der Detektorantwort auf vorgegebene Ereignisse

- kalibrieren einzelner Detektorkomponenten mit Simulationsergebnissen
- optimieren von Rekonstruktionsalgorithmen

Was versteht man unter ATLAS Computing (AC)?

Rekonstruktion

Softwareanalyse von Messdaten ($O(\frac{PB}{Jahr})$) zur Rekonstruktion von Ereignissen

- produziert auf das Wesentliche reduzierte Daten

Simulation

Simulation der Detektorantwort auf vorgegebene Ereignisse

- kalibrieren einzelner Detektorkomponenten mit Simulationsergebnissen
- optimieren von Rekonstruktionsalgorithmen

Was versteht man unter ATLAS Computing (AC)?

Rekonstruktion

Softwareanalyse von Messdaten ($O(\frac{PB}{Jahr})$) zur Rekonstruktion von Ereignissen

- produziert auf das Wesentliche reduzierte Daten

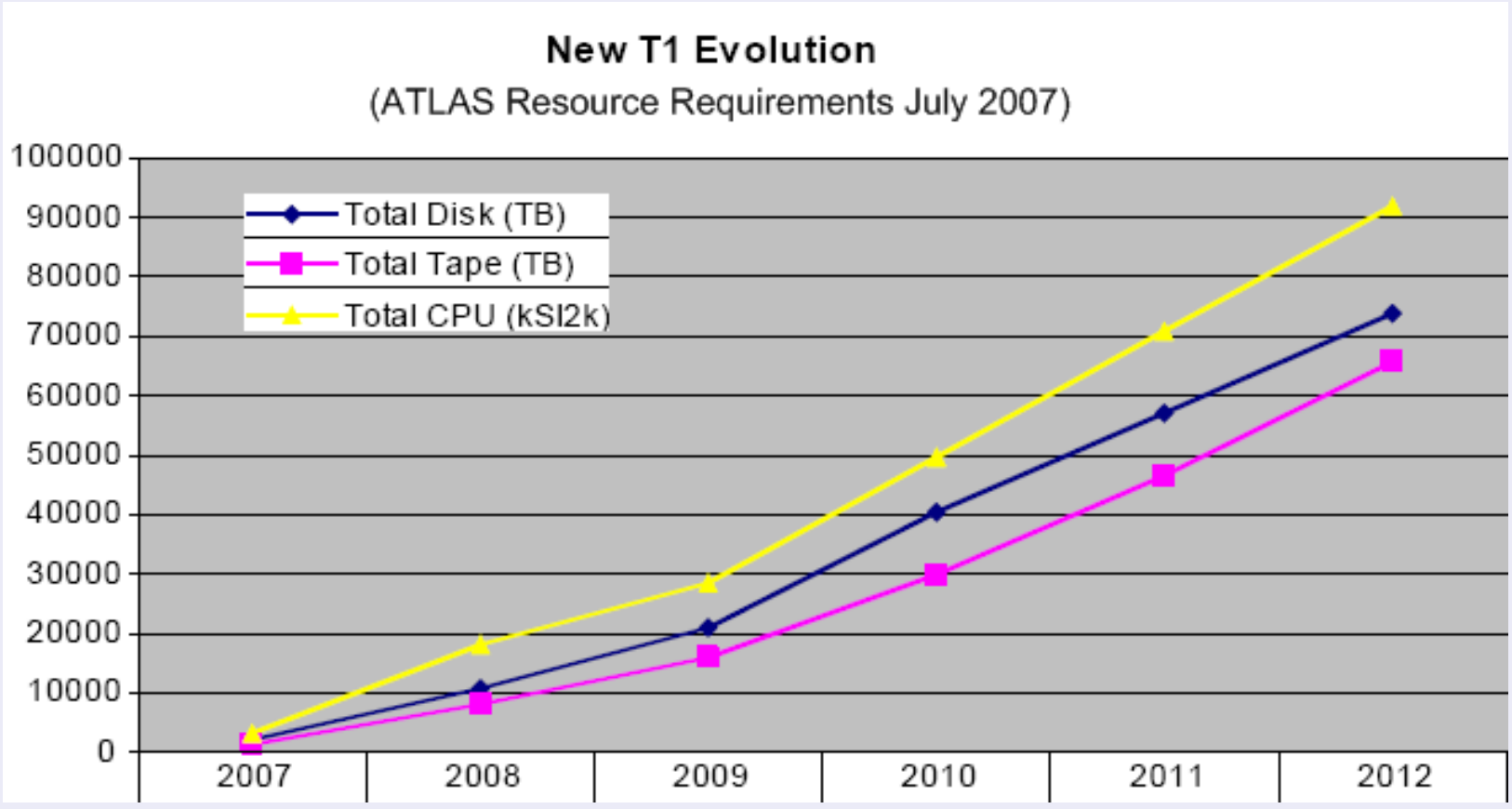
Simulation

Simulation der Detektorantwort auf vorgegebene Ereignisse

- kalibrieren einzelner Detektorkomponenten mit Simulationsergebnissen
- optimieren von Rekonstruktionsalgorithmen

Welche Dimensionen hat ATLAS Computing?

benötigte Tier 1 Resources ("nationale Zwischenstation")



(2 kSI2k $\hat{=}$ AMD Athlon 64 FX-62 (2x 2,8 GHz))

Wie wird ATLAS Computing umgesetzt?

Einsatz von ATLAS Software in LHC Computing Grid (LCG) RZ

- Softwarepaket kann sehr viel
- aber: buggy, instabil
- Verwaltung/Pflege von den Betriebssystemen und dem Jobsystem in den RZ des LCG ist personell sehr aufwendig und umständlich.

Stefan Kluth baut RZ für ATLAS Computing in München auf

- liest in iX über Amazon Web Services mit der Elastic Computing Cloud „EC2"
- → Cloud Computing (wie EC2) koennte viel einfacher und robuster sein als LCG

Warum dieser Entschluss?

Wie wird ATLAS Computing umgesetzt?

Einsatz von ATLAS Software in LHC Computing Grid (LCG) RZ

- Softwarepaket kann sehr viel
- aber: buggy, instabil
- Verwaltung/Pflege von den Betriebssystemen und dem Jobsystem in den RZ des LCG ist personell sehr aufwendig und umständlich.

Stefan Kluth baut RZ für ATLAS Computing in München auf

- liest in iX über Amazon Web Services mit der Elastic Computing Cloud „EC2"
- → Cloud Computing (wie EC2) koennte viel einfacher und robuster sein als LCG

Warum dieser Entschluss?

Wie wird ATLAS Computing umgesetzt?

Einsatz von ATLAS Software in LHC Computing Grid (LCG) RZ

- Softwarepaket kann sehr viel
- aber: buggy, instabil
- Verwaltung/Pflege von den Betriebssystemen und dem Jobsystem in den RZ des LCG ist personell sehr aufwendig und umständlich.

Stefan Kluth baut RZ für ATLAS Computing in München auf

- liest in iX über Amazon Web Services mit der Elastic Computing Cloud „EC2"
- → Cloud Computing (wie EC2) koennte viel einfacher und robuster sein als LCG

Warum dieser Entschluss?

Wie wird ATLAS Computing umgesetzt?

Einsatz von ATLAS Software in LHC Computing Grid (LCG) RZ

- Softwarepaket kann sehr viel
- aber: buggy, instabil
- Verwaltung/Pflege von den Betriebssystemen und dem Jobsystem in den RZ des LCG ist personell sehr aufwendig und umständlich.

Stefan Kluth baut RZ für ATLAS Computing in München auf

- liest in iX über Amazon Web Services mit der Elastic Computing Cloud „EC2"
- → Cloud Computing (wie EC2) koennte viel einfacher und robuster sein als LCG

Warum dieser Entschluss?

Wie wird ATLAS Computing umgesetzt?

Einsatz von ATLAS Software in LHC Computing Grid (LCG) RZ

- Softwarepaket kann sehr viel
- aber: buggy, instabil
- Verwaltung/Pflege von den Betriebssystemen und dem Jobsystem in den RZ des LCG ist personell sehr aufwendig und umständlich.

Stefan Kluth baut RZ für ATLAS Computing in München auf

- liest in iX über Amazon Web Services mit der Elastic Computing Cloud „EC2“
- → Cloud Computing (wie EC2) koennte viel einfacher und robuster sein als LCG

Warum dieser Entschluss?

Wie wird ATLAS Computing umgesetzt?

Einsatz von ATLAS Software in LHC Computing Grid (LCG) RZ

- Softwarepaket kann sehr viel
- aber: buggy, instabil
- Verwaltung/Pflege von den Betriebssystemen und dem Jobsystem in den RZ des LCG ist personell sehr aufwendig und umständlich.

Stefan Kluth baut RZ für ATLAS Computing in München auf

- liest in iX über Amazon Web Services mit der Elastic Computing Cloud „EC2“
- → Cloud Computing (wie EC2) koennte viel einfacher und robuster sein als LCG

Warum dieser Entschluss?

Wie wird ATLAS Computing umgesetzt?

Einsatz von ATLAS Software in LHC Computing Grid (LCG) RZ

- Softwarepaket kann sehr viel
- aber: buggy, instabil
- Verwaltung/Pflege von den Betriebssystemen und dem Jobsystem in den RZ des LCG ist personell sehr aufwendig und umständlich.

Stefan Kluth baut RZ für ATLAS Computing in München auf

- liest in iX über Amazon Web Services mit der Elastic Computing Cloud „EC2“
- $\longrightarrow$ Cloud Computing (wie EC2) koennte viel einfacher und robuster sein als LCG

Warum dieser Entschluss?

Wie wird ATLAS Computing umgesetzt?

Einsatz von ATLAS Software in LHC Computing Grid (LCG) RZ

- Softwarepaket kann sehr viel
- aber: buggy, instabil
- Verwaltung/Pflege von den Betriebssystemen und dem Jobsystem in den RZ des LCG ist personell sehr aufwendig und umständlich.

Stefan Kluth baut RZ für ATLAS Computing in München auf

- liest in iX über Amazon Web Services mit der Elastic Computing Cloud „EC2“
- $\longrightarrow$ Cloud Computing (wie EC2) koennte viel einfacher und robuster sein als LCG

Warum dieser Entschluss?

Übersicht

Was hat Amazon mit Web Services zu tun?

riesige Webapplikation mit riesiger technischer Infrastruktur!

- Webseiten rund um die Welt
- die Lösungen dazu komplett selbst entwickelt
- irgendwann Techniken anderen Webfirmen weiterverkauft
- 2006 Gründung „Amazon Web Services“
- AWS stellt Web-Infrastruktur für Webfirmen jeder Größe

Was hat Amazon mit Web Services zu tun?

riesige Webapplikation mit riesiger technischer Infrastruktur!

- Webseiten rund um die Welt
- die Lösungen dazu komplett selbst entwickelt
- irgendwann Techniken anderen Webfirmen weiterverkauft
- 2006 Gründung „Amazon Web Services“
- AWS stellt Web-Infrastruktur für Webfirmen jeder Größe

Was hat Amazon mit Web Services zu tun?

riesige Webapplikation mit riesiger technischer Infrastruktur!

- Webseiten rund um die Welt
- die Lösungen dazu komplett selbst entwickelt
- irgendwann Techniken anderen Webfirmen weiterverkauft
- 2006 Gründung „Amazon Web Services“
- AWS stellt Web-Infrastruktur für Webfirmen jeder Größe

Was hat Amazon mit Web Services zu tun?

riesige Webapplikation mit riesiger technischer Infrastruktur!

- Webseiten rund um die Welt
- die Lösungen dazu komplett selbst entwickelt
- irgendwann Techniken anderen Webfirmen weiterverkauft
- 2006 Gründung „Amazon Web Services“
- AWS stellt Web-Infrastruktur für Webfirmen jeder Größe

Was hat Amazon mit Web Services zu tun?

riesige Webapplikation mit riesiger technischer Infrastruktur!

- Webseiten rund um die Welt
- die Lösungen dazu komplett selbst entwickelt
- irgendwann Techniken anderen Webfirmen weiterverkauft
- 2006 Gründung „Amazon Web Services“
- AWS stellt Web-Infrastruktur für Webfirmen jeder Größe

Was hat Amazon mit Web Services zu tun?

riesige Webapplikation mit riesiger technischer Infrastruktur!

- Webseiten rund um die Welt
- die Lösungen dazu komplett selbst entwickelt
- irgendwann Techniken anderen Webfirmen weiterverkauft
- 2006 Gründung „Amazon Web Services“
- AWS stellt Web-Infrastruktur für Webfirmen jeder Größe

Was sind die AWS genau?

die wichtigsten drei Dienste..

- Speicher: „Simple Storage S3“
- Datenbank: „Simple DB“
- Rechenleistung: „Elastic Computing Cloud EC2“

die wichtigsten zwei Prinzipien..

- man kann sofort soviel von allem benutzen, wie man möchte
- man bezahlt genau das, was man verbraucht/benutzt

Was sind die AWS genau?

die wichtigsten drei Dienste..

- Speicher: „Simple Storage S3“
- Datenbank: „Simple DB“
- Rechenleistung: „Elastic Computing Cloud EC2“

die wichtigsten zwei Prinzipien..

- man kann sofort soviel von allem benutzen, wie man möchte
- man bezahlt genau das, was man verbraucht/benutzt

Was sind die AWS genau?

die wichtigsten drei Dienste..

- Speicher: „Simple Storage S3“
- Datenbank: „Simple DB“
- Rechenleistung: „Elastic Computing Cloud EC2“

die wichtigsten zwei Prinzipien..

- man kann sofort soviel von allem benutzen, wie man möchte
- man bezahlt genau das, was man verbraucht/benutzt

Was sind die AWS genau?

die wichtigsten drei Dienste..

- Speicher: „Simple Storage S3“
- Datenbank: „Simple DB“
- Rechenleistung: „Elastic Computing Cloud EC2“

die wichtigsten zwei Prinzipien..

- man kann sofort soviel von allem benutzen, wie man möchte
- man bezahlt genau das, was man verbraucht/benutzt

Was sind die AWS genau?

die wichtigsten drei Dienste..

- Speicher: „Simple Storage S3“
- Datenbank: „Simple DB“
- Rechenleistung: „Elastic Computing Cloud EC2“

die wichtigsten zwei Prinzipien..

- man kann sofort soviel von allem benutzen, wie man möchte
- man bezahlt genau das, was man verbraucht/benutzt

Was sind die AWS genau?

die wichtigsten drei Dienste..

- Speicher: „Simple Storage S3“
- Datenbank: „Simple DB“
- Rechenleistung: „Elastic Computing Cloud EC2“

die wichtigsten zwei Prinzipien..

- man kann sofort soviel von allem benutzen, wie man möchte
- man bezahlt genau das, was man verbraucht/benutzt

Was sind die AWS genau?

die wichtigsten drei Dienste..

- Speicher: „Simple Storage S3“
- Datenbank: „Simple DB“
- Rechenleistung: „Elastic Computing Cloud EC2“

die wichtigsten zwei Prinzipien..

- man kann sofort soviel von allem benutzen, wie man möchte
- man bezahlt genau das, was man verbraucht/benutzt

Übersicht

Was mietet man da eigentlich für Rechner?

Keine echten „Hardware-Rechner“, sondern sog. „virtuelle Maschinen“!

Virtuelle Maschine (VM):

Was mietet man da eigentlich für Rechner?

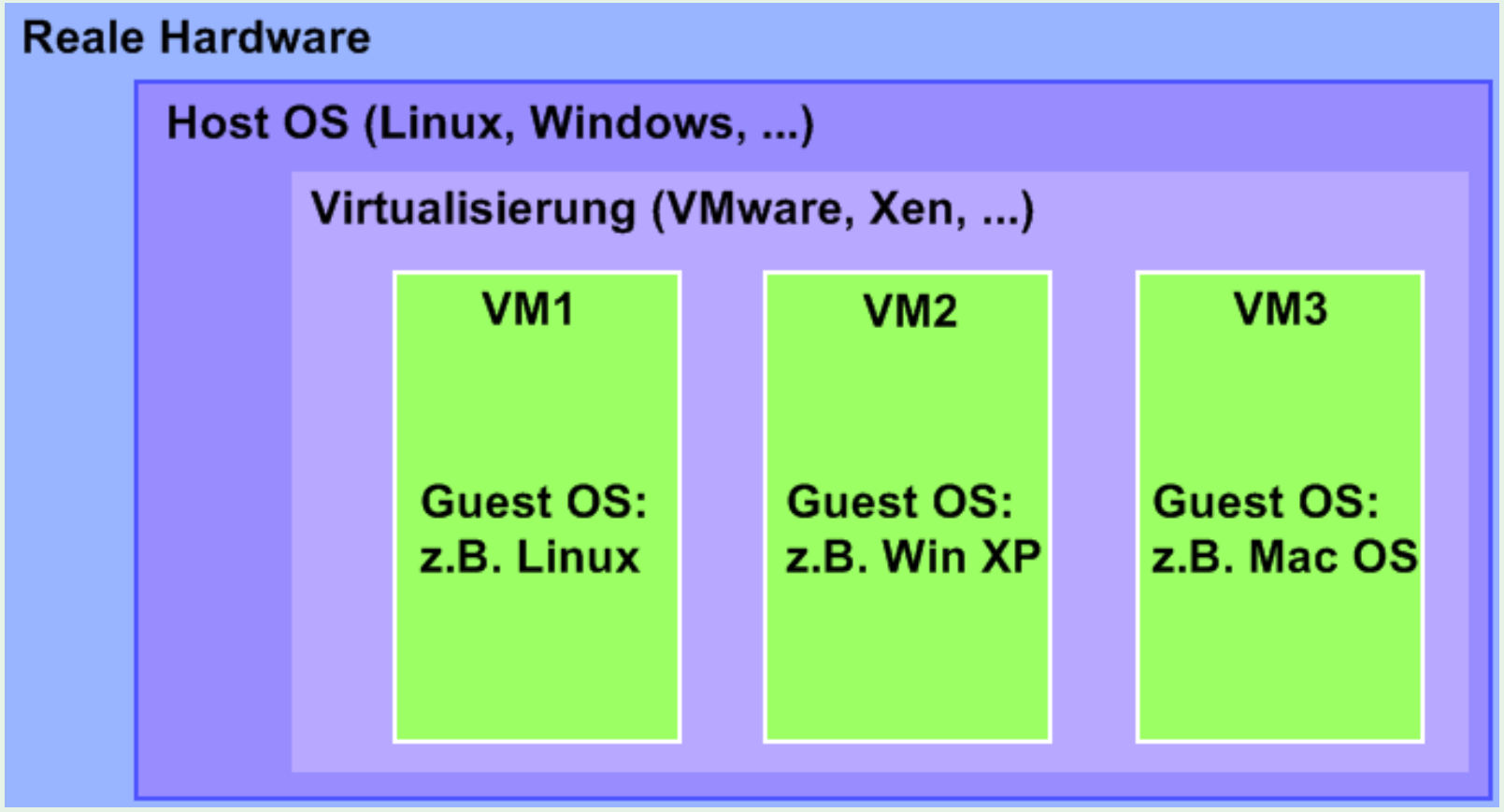
Keine echten „Hardware-Rechner“, sondern sog. „virtuelle Maschinen“!

Virtuelle Maschine (VM):

Was mietet man da eigentlich für Rechner?

Keine echten „Hardware-Rechner“, sondern sog. „virtuelle Maschinen“!

Virtuelle Maschine (VM):



wie benutzt man VMs bei EC2?

Begriff: bei EC2 heißt eine laufende VM **Instanz**

- wähle Instanztyp (virtuelle Hardware)
- starte beliebig viele Instanzen in **Elastic** Computing Cloud
- arbeite als *root* (Macht über System, Software)
- beende Instanz(en) zu beliebigem Zeitpunkt

wie benutzt man VMs bei EC2?

Begriff: bei EC2 heißt eine laufende VM **Instanz**

- wähle Instanztyp (virtuelle Hardware)
- starte beliebig viele Instanzen in **Elastic** Computing Cloud
- arbeite als *root* (Macht über System, Software)
- beende Instanz(en) zu beliebigem Zeitpunkt

wie benutzt man VMs bei EC2?

Begriff: bei EC2 heißt eine laufende VM **Instanz**

- wähle Instanztyp (virtuelle Hardware)
- starte beliebig viele Instanzen in **Elastic** Computing Cloud
- arbeite als *root* (Macht über System, Software)
- beende Instanz(en) zu beliebigem Zeitpunkt

wie benutzt man VMs bei EC2?

Begriff: bei EC2 heißt eine laufende VM **Instanz**

- wähle Instanztyp (virtuelle Hardware)
- starte beliebig viele Instanzen in **Elastic** Computing Cloud
- arbeite als *root* (Macht über System, Software)
- beende Instanz(en) zu beliebigem Zeitpunkt

wie benutzt man VMs bei EC2?

Begriff: bei EC2 heißt eine laufende VM **Instanz**

- wähle Instanztyp (virtuelle Hardware)
- starte beliebig viele Instanzen in **Elastic** Computing Cloud
- arbeite als *root* (Macht über System, Software)
- beende Instanz(en) zu beliebigem Zeitpunkt

wie benutzt man VMs bei EC2?

Begriff: bei EC2 heißt eine laufende VM **Instanz**

- wähle Instanztyp (virtuelle Hardware)
- starte beliebig viele Instanzen in **Elastic** Computing Cloud
- arbeite als *root* (Macht über System, Software)
- beende Instanz(en) zu beliebigem Zeitpunkt

Vorteile einer Computing Cloud mit VMs

- Benutzer berührt das *Host OS* nicht
- als *root* arbeiten ohne Sorge (kaputt? ups - weg und neu!)
- VMs auf derselben Hardware sind unabhängig

→ Läuft die Virtualisierung stabil (tut sie), läuft das *Host OS* stabil (praktisch wartungsfrei)

Vorteile einer Computing Cloud mit VMs

- Benutzer berührt das *Host OS* nicht
- als *root* arbeiten ohne Sorge (kaputt? ups - weg und neu!)
- VMs auf derselben Hardware sind unabhängig

→ Läuft die Virtualisierung stabil (tut sie), läuft das *Host OS* stabil (praktisch wartungsfrei)

Vorteile einer Computing Cloud mit VMs

- Benutzer berührt das *Host OS* nicht
- als *root* arbeiten ohne Sorge (kaputt? ups - weg und neu!)
- VMs auf derselben Hardware sind unabhängig

→ Läuft die Virtualisierung stabil (tut sie), läuft das *Host OS* stabil (praktisch wartungsfrei)

Vorteile einer Computing Cloud mit VMs

- Benutzer berührt das *Host OS* nicht
- als *root* arbeiten ohne Sorge (kaputt? ups - weg und neu!)
- VMs auf derselben Hardware sind unabhängig

→ Läuft die Virtualisierung stabil (tut sie), läuft das *Host OS* stabil (praktisch wartungsfrei)

Vorteile einer Computing Cloud mit VMs

- Benutzer berührt das *Host OS* nicht
- als *root* arbeiten ohne Sorge (kaputt? ups - weg und neu!)
- VMs auf derselben Hardware sind unabhängig

→ Läuft die Virtualisierung stabil (tut sie), läuft das *Host OS* stabil (praktisch wartungsfrei)

Wie wählt man das OS und die Software?

Konzept: Amazon Machine Image (**AMI**)

Was ist ein AMI?

- Abbild des Dateisystems einer Instanz
- gespeichert auf S3

Ich möchte ein spezielles System! Wie?

- es gibt viele öffentliche AMIs
- verschiedene Distris und Software
- man kann komplett eigene AMIs von Linuxsystemen erstellen

Wie wählt man das OS und die Software?

Konzept: Amazon Machine Image (**AMI**)

Was ist ein AMI?

- Abbild des Dateisystems einer Instanz
- gespeichert auf S3

Ich möchte ein spezielles System! Wie?

- es gibt viele öffentliche AMIs
- verschiedene Distris und Software
- man kann komplett eigene AMIs von Linuxsystemen erstellen

Wie wählt man das OS und die Software?

Konzept: Amazon Machine Image (**AMI**)

Was ist ein AMI?

- Abbild des Dateisystems einer Instanz
- gespeichert auf S3

Ich möchte ein spezielles System! Wie?

- es gibt viele öffentliche AMIs
- verschiedene Distris und Software
- man kann komplett eigene AMIs von Linuxsystemen erstellen

Wie wählt man das OS und die Software?

Konzept: Amazon Machine Image (**AMI**)

Was ist ein AMI?

- Abbild des Dateisystems einer Instanz
- gespeichert auf S3

Ich möchte ein spezielles System! Wie?

- es gibt viele öffentliche AMIs
- verschiedene Distris und Software
- man kann komplett eigene AMIs von Linuxsystemen erstellen

Wie wählt man das OS und die Software?

Konzept: Amazon Machine Image (**AMI**)

Was ist ein AMI?

- Abbild des Dateisystems einer Instanz
- gespeichert auf S3

Ich möchte ein spezielles System! Wie?

- es gibt viele öffentliche AMIs
- verschiedene Distris und Software
- man kann komplett eigene AMIs von Linuxsystemen erstellen

Wie wählt man das OS und die Software?

Konzept: Amazon Machine Image (**AMI**)

Was ist ein AMI?

- Abbild des Dateisystems einer Instanz
- gespeichert auf S3

Ich möchte ein spezielles System! Wie?

- es gibt viele öffentliche AMIs
- verschiedene Distris und Software
- man kann komplett eigene AMIs von Linuxsystemen erstellen

Wie wählt man das OS und die Software?

Konzept: Amazon Machine Image (**AMI**)

Was ist ein AMI?

- Abbild des Dateisystems einer Instanz
- gespeichert auf S3

Ich möchte ein spezielles System! Wie?

- es gibt viele öffentliche AMIs
- verschiedene Distris und Software
- man kann komplett eigene AMIs von Linuxsystemen erstellen

Wie wählt man das OS und die Software?

Konzept: Amazon Machine Image (**AMI**)

Was ist ein AMI?

- Abbild des Dateisystems einer Instanz
- gespeichert auf S3

Ich möchte ein spezielles System! Wie?

- es gibt viele öffentliche AMIs
- verschiedene Distris und Software
- man kann komplett eigene AMIs von Linuxsystemen erstellen

Der Nutzen von EC2 für ATLAS Computing

Kosten:

EC2 deutlich teurer, als wenn man eigene Hardware voll auslastet
$\rightarrow$ ATLAS Computing komplett auf EC2 natürlich nicht sinnvoll

Aber:

Zum Abfangen von Spitzenlasten ideal (Deadlines, Konferenzen, ...)

Der Nutzen von EC2 für ATLAS Computing

Kosten:

EC2 deutlich teurer, als wenn man eigene Hardware voll auslastet
$\rightarrow$ ATLAS Computing komplett auf EC2 natürlich nicht sinnvoll

Aber:

Zum Abfangen von Spitzenlasten ideal (Deadlines, Konferenzen, ...)

Idee / Wunschvorstellung

Computing Clouds mit eigener Hardware:

- Aufbau analog zu EC2
- ein Jobsystem **vollständig kompatibel** zu EC2

Ergebnis:

- RZ müssen deutlich weniger Systempflege betreiben
- wenn erforderlich, einfach bei EC2 rechnen

$\longrightarrow$ einfacheres und robusteres verteiltes Computing als LCG

Idee / Wunschvorstellung

Computing Clouds mit eigener Hardware:

- Aufbau analog zu EC2
- ein Jobsystem **vollständig kompatibel** zu EC2

Ergebnis:

- RZ müssen deutlich weniger Systempflege betreiben
- wenn erforderlich, einfach bei EC2 rechnen

→ einfacheres und robusteres verteiltes Computing als LCG

Idee / Wunschvorstellung

Computing Clouds mit eigener Hardware:

- Aufbau analog zu EC2
- ein Jobsystem **vollständig kompatibel** zu EC2

Ergebnis:

- RZ müssen deutlich weniger Systempflege betreiben
- wenn erforderlich, einfach bei EC2 rechnen

→ einfacheres und robusteres verteiltes Computing als LCG

Idee / Wunschvorstellung

Computing Clouds mit eigener Hardware:

- Aufbau analog zu EC2
- ein Jobsystem **vollständig kompatibel** zu EC2

Ergebnis:

- RZ müssen deutlich weniger Systempflege betreiben
- wenn erforderlich, einfach bei EC2 rechnen

$\longrightarrow$ einfacheres und robusteres verteiltes Computing als LCG

Idee / Wunschvorstellung

Computing Clouds mit eigener Hardware:

- Aufbau analog zu EC2
- ein Jobsystem **vollständig kompatibel** zu EC2

Ergebnis:

- RZ müssen deutlich weniger Systempflege betreiben
- wenn erforderlich, einfach bei EC2 rechnen

$\longrightarrow$ einfacheres und robusteres verteiltes Computing als LCG

Idee / Wunschvorstellung

Computing Clouds mit eigener Hardware:

- Aufbau analog zu EC2
- ein Jobsystem **vollständig kompatibel** zu EC2

Ergebnis:

- RZ müssen deutlich weniger Systempflege betreiben
- wenn erforderlich, einfach bei EC2 rechnen

→ einfacheres und robusteres verteiltes Computing als LCG

Idee / Wunschvorstellung

Computing Clouds mit eigener Hardware:

- Aufbau analog zu EC2
- ein Jobsystem **vollständig kompatibel** zu EC2

Ergebnis:

- RZ müssen deutlich weniger Systempflege betreiben
- wenn erforderlich, einfach bei EC2 rechnen

$\longrightarrow$ einfacheres und robusteres verteiltes Computing als LCG

Übersicht

Welche Möglichkeiten für AC mit AWS?

AWS verstehen:

gute „Developer Guides“ zu den AWS, tutorials, ...

ATLAS Softwarepaket verstehen:

- Ansammlung von C++, Python, Fortran, bashskripten, configfiles, Detektordaten
- schwer überschaubare eigentümliche Dinge
- fast 10 *GB*!
- fehlerbehaftet, sehr umständlich zu bedienen
- schlecht portierbar: **nur** entwickelt Scientific Linux 4 (alt!)
- sehr unübersichtlich dokumentiert

→ eine Wissenschaft für sich

Welche Möglichkeiten für AC mit AWS?

AWS verstehen:

gute „Developer Guides“ zu den AWS, tutorials, ...

ATLAS Softwarepaket verstehen:

- Ansammlung von C++, Python, Fortran, bashskripten, configfiles, Detektordaten
- schwer überschaubare eigentümliche Dinge
- fast 10 *GB*!
- fehlerbehaftet, sehr umständlich zu bedienen
- schlecht portierbar: **nur** entwickelt Scientific Linux 4 (alt!)
- sehr unübersichtlich dokumentiert

$\longrightarrow$ eine Wissenschaft für sich

Welche Möglichkeiten für AC mit AWS?

AWS verstehen:

gute „Developer Guides“ zu den AWS, tutorials, ...

ATLAS Softwarepaket verstehen:

- Ansammlung von C++, Python, Fortran, bashskripten, configfiles, Detektordaten
- schwer überschaubare eigentümliche Dinge
- fast 10 *GB*!
- fehlerbehaftet, sehr umständlich zu bedienen
- schlecht portierbar: **nur** entwickelt Scientific Linux 4 (alt!)
- sehr unübersichtlich dokumentiert

$\longrightarrow$ eine Wissenschaft für sich

Welche Möglichkeiten für AC mit AWS?

AWS verstehen:

gute „Developer Guides“ zu den AWS, tutorials, ...

ATLAS Softwarepaket verstehen:

- Ansammlung von C++, Python, Fortran, bashskripten, configfiles, Detektordaten
- schwer überschaubare eigentümliche Dinge
- fast 10 *GB*!
- fehlerbehaftet, sehr umständlich zu bedienen
- schlecht portierbar: **nur** entwickelt Scientific Linux 4 (alt!)
- sehr unübersichtlich dokumentiert

→ eine Wissenschaft für sich

Welche Möglichkeiten für AC mit AWS?

AWS verstehen:

gute „Developer Guides“ zu den AWS, tutorials, ...

ATLAS Softwarepaket verstehen:

- Ansammlung von C++, Python, Fortran, bashskripten, configfiles, Detektordaten
- schwer überschaubare eigentümliche Dinge
- fast 10 *GB*!
- fehlerbehaftet, sehr umständlich zu bedienen
- schlecht portierbar: **nur** entwickelt Scientific Linux 4 (alt!)
- sehr unübersichtlich dokumentiert

→ eine Wissenschaft für sich

Welche Möglichkeiten für AC mit AWS?

AWS verstehen:

gute „Developer Guides“ zu den AWS, tutorials, ...

ATLAS Softwarepaket verstehen:

- Ansammlung von C++, Python, Fortran, bashskripten, configfiles, Detektordaten
- schwer überschaubare eigentümliche Dinge
- fast 10 *GB*!
- fehlerbehaftet, sehr umständlich zu bedienen
- schlecht portierbar: **nur** entwickelt Scientific Linux 4 (alt!)
- sehr unübersichtlich dokumentiert

→ eine Wissenschaft für sich

Welche Möglichkeiten für AC mit AWS?

AWS verstehen:

gute „Developer Guides“ zu den AWS, tutorials, ...

ATLAS Softwarepaket verstehen:

- Ansammlung von C++, Python, Fortran, bashskripten, configfiles, Detektordaten
- schwer überschaubare eigentümliche Dinge
- fast 10 *GB*!
- fehlerbehaftet, sehr umständlich zu bedienen
- schlecht portierbar: **nur** entwickelt Scientific Linux 4 (alt!)
- sehr unübersichtlich dokumentiert

$\longrightarrow$ eine Wissenschaft für sich

Welche Möglichkeiten für AC mit AWS?

AWS verstehen:

gute „Developer Guides“ zu den AWS, tutorials, ...

ATLAS Softwarepaket verstehen:

- Ansammlung von C++, Python, Fortran, bashskripten, configfiles, Detektordaten
- schwer überschaubare eigentümliche Dinge
- fast 10 *GB*!
- fehlerbehaftet, sehr umständlich zu bedienen
- schlecht portierbar: **nur** entwickelt Scientific Linux 4 (alt!)
- sehr unübersichtlich dokumentiert

$\longrightarrow$ eine Wissenschaft für sich

erste Hürde

ein OS so wählen und modifizieren, dass...

- ein AMI davon auf EC2 halbwegs gut startet (Ziel: ssh login)
- ein ATLAS Software Release darauf korrekt läuft

Laut Doku **SL4 Software für EC2 deutlich zu alt**!

→ ATLAS Software auf neueren Distris testen:

- versch. moderne Distributionen installiert, modifiziert und getestet
  - Fedora 9, Fedora 8, Fedora 7, Scientific Linux 5
  - irgendwann aufgegeben

→ nur auf Scientific Linux 4 alle Tests erfolgreich

erste Hürde

ein OS so wählen und modifizieren, dass...

- ein AMI davon auf EC2 halbwegs gut startet (Ziel: ssh login)
- ein ATLAS Software Release darauf korrekt läuft

Laut Doku **SL4 Software für EC2 deutlich zu alt**!

→ ATLAS Software auf neueren Distris testen:

- versch. moderne Distributionen installiert, modifiziert und getestet
- Fedora 9, Fedora 8, Fedora 7, Scientific Linux 5
- irgendwann aufgegeben

→ nur auf Scientific Linux 4 alle Tests erfolgreich

erste Hürde

ein OS so wählen und modifizieren, dass...

- ein AMI davon auf EC2 halbwegs gut startet (Ziel: ssh login)
- ein ATLAS Software Release darauf korrekt läuft

Laut Doku **SL4 Software für EC2 deutlich zu alt**!

→ ATLAS Software auf neueren Distris testen:

- versch. moderne Distributionen installiert, modifiziert und getestet
- Fedora 9, Fedora 8, Fedora 7, Scientific Linux 5
- irgendwann aufgegeben

→ nur auf Scientific Linux 4 alle Tests erfolgreich

erste Hürde

ein OS so wählen und modifizieren, dass...

- ein AMI davon auf EC2 halbwegs gut startet (Ziel: ssh login)
- ein ATLAS Software Release darauf korrekt läuft

Laut Doku **SL4 Software für EC2 deutlich zu alt**!

→ ATLAS Software auf neueren Distris testen:

- versch. moderne Distributionen installiert, modifiziert und getestet
- Fedora 9, Fedora 8, Fedora 7, Scientific Linux 5
- irgendwann aufgegeben

→ nur auf Scientific Linux 4 alle Tests erfolgreich

erste Hürde

ein OS so wählen und modifizieren, dass...

- ein AMI davon auf EC2 halbwegs gut startet (Ziel: ssh login)
- ein ATLAS Software Release darauf korrekt läuft

Laut Doku **SL4 Software für EC2 deutlich zu alt**!

→ ATLAS Software auf neueren Distris testen:

- versch. moderne Distributionen installiert, modifiziert und getestet
- Fedora 9, Fedora 8, Fedora 7, Scientific Linux 5
- irgendwann aufgegeben

→ nur auf Scientific Linux 4 alle Tests erfolgreich

erste Hürde

ein OS so wählen und modifizieren, dass...

- ein AMI davon auf EC2 halbwegs gut startet (Ziel: ssh login)
- ein ATLAS Software Release darauf korrekt läuft

Laut Doku **SL4 Software für EC2 deutlich zu alt**!

→ ATLAS Software auf neueren Distris testen:

- versch. moderne Distributionen installiert, modifiziert und getestet
- Fedora 9, Fedora 8, Fedora 7, Scientific Linux 5
- irgendwann aufgegeben

→ nur auf Scientific Linux 4 alle Tests erfolgreich

erste Hürde

ein OS so wählen und modifizieren, dass...

- ein AMI davon auf EC2 halbwegs gut startet (Ziel: ssh login)
- ein ATLAS Software Release darauf korrekt läuft

Laut Doku **SL4 Software für EC2 deutlich zu alt**!

→ ATLAS Software auf neueren Distris testen:

- versch. moderne Distributionen installiert, modifiziert und getestet
- Fedora 9, Fedora 8, Fedora 7, Scientific Linux 5
- irgendwann aufgegeben

→ nur auf Scientific Linux 4 alle Tests erfolgreich

erste Hürde

ein OS so wählen und modifizieren, dass...

- ein AMI davon auf EC2 halbwegs gut startet (Ziel: ssh login)
- ein ATLAS Software Release darauf korrekt läuft

Laut Doku **SL4 Software für EC2 deutlich zu alt**!

→ ATLAS Software auf neueren Distris testen:

- versch. moderne Distributionen installiert, modifiziert und getestet
- Fedora 9, Fedora 8, Fedora 7, Scientific Linux 5
- irgendwann aufgegeben

→ nur auf Scientific Linux 4 alle Tests erfolgreich

Ein lauffähiges AMI von SL4 erstellen

mit „AMI Tools: *ec2-bundle-vol*“

kann laufendes Linuxsystem in ein AMI „bundlen“

2 Wochen Zusammengefasst:

- Software auf unkonventionellen Wegen installiert/modifiziert
- Systemkomponenten ausgetrickst
- Amazon Tools ausgetrickst
- Skripte geschrieben, um vieles zu vereinfachen

→ vom lokalen SL4 über ein wenig funktionierendes SL4 AMI zu einem benutzbaren SL4 AMI, dass man simpel weiterentwickeln kann

Detaillierte Dokumentation:

http://gehrcke.de/awsac/

Ein lauffähiges AMI von SL4 erstellen

mit „AMI Tools: *ec2-bundle-vol*“

kann laufendes Linuxsystem in ein AMI „bundlen“

2 Wochen Zusammengefasst:

- Software auf unkonventionellen Wegen installiert/modifiziert
- Systemkomponenten ausgetrickst
- Amazon Tools ausgetrickst
- Skripte geschrieben, um vieles zu vereinfachen

→ vom lokalen SL4 über ein wenig funktionierendes SL4 AMI zu einem benutzbaren SL4 AMI, dass man simpel weiterentwickeln kann

Detaillierte Dokumentation:

http://gehrcke.de/awsac/

Ein lauffähiges AMI von SL4 erstellen

mit „AMI Tools: *ec2-bundle-vol*“

kann laufendes Linuxsystem in ein AMI „bundlen“

2 Wochen Zusammengefasst:

- Software auf unkonventionellen Wegen installiert/modifiziert
- Systemkomponenten ausgetrickst
- Amazon Tools ausgetrickst
- Skripte geschrieben, um vieles zu vereinfachen

→ vom lokalen SL4 über ein wenig funktionierendes SL4 AMI zu einem benutzbaren SL4 AMI, dass man simpel weiterentwickeln kann

Detaillierte Dokumentation:

http://gehrcke.de/awsac/

Ein lauffähiges AMI von SL4 erstellen

mit „AMI Tools: *ec2-bundle-vol*“

kann laufendes Linuxsystem in ein AMI „bundlen“

2 Wochen Zusammengefasst:

- Software auf unkonventionellen Wegen installiert/modifiziert
- Systemkomponenten ausgetrickst
- Amazon Tools ausgetrickst
- Skripte geschrieben, um vieles zu vereinfachen

→ vom lokalen SL4 über ein wenig funktionierendes SL4 AMI zu einem benutzbaren SL4 AMI, dass man simpel weiterentwickeln kann

Detaillierte Dokumentation:

http://gehrcke.de/awsac/

Ein lauffähiges AMI von SL4 erstellen

mit „AMI Tools: *ec2-bundle-vol*“

kann laufendes Linuxsystem in ein AMI „bundlen“

2 Wochen Zusammengefasst:

- Software auf unkonventionellen Wegen installiert/modifiziert
- Systemkomponenten ausgetrickst
- Amazon Tools ausgetrickst
- Skripte geschrieben, um vieles zu vereinfachen

→ vom lokalen SL4 über ein wenig funktionierendes SL4 AMI zu einem benutzbaren SL4 AMI, dass man simpel weiterentwickeln kann

Detaillierte Dokumentation:

http://gehrcke.de/awsac/

Ein lauffähiges AMI von SL4 erstellen

mit „AMI Tools: *ec2-bundle-vol*“

kann laufendes Linuxsystem in ein AMI „bundlen“

2 Wochen Zusammengefasst:

- Software auf unkonventionellen Wegen installiert/modifiziert
- Systemkomponenten ausgetrickst
- Amazon Tools ausgetrickst
- Skripte geschrieben, um vieles zu vereinfachen

→ vom lokalen SL4 über ein wenig funktionierendes SL4 AMI zu einem benutzbaren SL4 AMI, dass man simpel weiterentwickeln kann

Detaillierte Dokumentation:

http://gehrcke.de/awsac/

Ein lauffähiges AMI von SL4 erstellen

mit „AMI Tools: *ec2-bundle-vol*“

kann laufendes Linuxsystem in ein AMI „bundlen“

2 Wochen Zusammengefasst:

- Software auf unkonventionellen Wegen installiert/modifiziert
- Systemkomponenten ausgetrickst
- Amazon Tools ausgetrickst
- Skripte geschrieben, um vieles zu vereinfachen

→ vom lokalen SL4 über ein wenig funktionierendes SL4 AMI zu einem benutzbaren SL4 AMI, dass man simpel weiterentwickeln kann

Detaillierte Dokumentation:

http://gehrcke.de/awsac/

Ein lauffähiges AMI von SL4 erstellen

mit „AMI Tools: *ec2-bundle-vol*“

kann laufendes Linuxsystem in ein AMI „bundlen“

2 Wochen Zusammengefasst:

- Software auf unkonventionellen Wegen installiert/modifiziert
- Systemkomponenten ausgetrickst
- Amazon Tools ausgetrickst
- Skripte geschrieben, um vieles zu vereinfachen

→ vom lokalen SL4 über ein wenig funktionierendes SL4 AMI zu einem benutzbaren SL4 AMI, dass man simpel weiterentwickeln kann

Detaillierte Dokumentation:

http://gehrcke.de/awsac/

Läuft AC auf EC2? → ATLAS Software installieren

Vorüberlegungen:

- AMI maximal 10 *GB*
- häufig neues Software Release
- versch. Leute arbeiten mit versch. Releases

→ Software nicht in AMI bundeln

Hat man nur diese 10 *GB* zum Arbeiten? Nein - Instanzspeicher:

groß, aber geht bei Instanzterminierung verloren

Läuft AC auf EC2? → ATLAS Software installieren

Vorüberlegungen:

- AMI maximal 10 *GB*
- häufig neues Software Release
- versch. Leute arbeiten mit versch. Releases

→ Software nicht in AMI bundeln

Hat man nur diese 10 *GB* zum Arbeiten? Nein - Instanzspeicher:

groß, aber geht bei Instanzterminierung verloren

Läuft AC auf EC2? → ATLAS Software installieren

Vorüberlegungen:

- AMI maximal 10 *GB*
- häufig neues Software Release
- versch. Leute arbeiten mit versch. Releases

→ Software nicht in AMI bundeln

Hat man nur diese 10 *GB* zum Arbeiten? Nein - Instanzspeicher:

groß, aber geht bei Instanzterminierung verloren

Läuft AC auf EC2? → ATLAS Software installieren

Vorüberlegungen:

- AMI maximal 10 *GB*
- häufig neues Software Release
- versch. Leute arbeiten mit versch. Releases

→ Software nicht in AMI bundeln

Hat man nur diese 10 *GB* zum Arbeiten? Nein - Instanzspeicher:

groß, aber geht bei Instanzterminierung verloren

Läuft AC auf EC2? → ATLAS Software installieren

Vorüberlegungen:

- AMI maximal 10 *GB*
- häufig neues Software Release
- versch. Leute arbeiten mit versch. Releases

→ Software nicht in AMI bundeln

Hat man nur diese 10 *GB* zum Arbeiten? Nein - Instanzspeicher:

groß, aber geht bei Instanzterminierung verloren

Läuft AC auf EC2? → ATLAS Software installieren

Vorüberlegungen:

- AMI maximal 10 *GB*
- häufig neues Software Release
- versch. Leute arbeiten mit versch. Releases

→ Software nicht in AMI bundeln

Hat man nur diese 10 *GB* zum Arbeiten? Nein - Instanzspeicher:

groß, aber geht bei Instanzterminierung verloren

Lösung: Elastic Block Storages

EC2-API Update 20. August 2008:

Einführung der „Elastic Block Storages“ (**EBS**)

- EBS lassen sich starten/beenden wie Instanzen
- permanente Speicher, unabhängig von Instanzen
- EBS kann an genau eine Instanz gekoppelt und ins Dateisystem eingebunden werden
- EBS existiert bei Instanzbeendigung weiter
- Backup: „EBS-Snapshot“ auf S3 speichern

EBS erstellt, ATLAS Software (mühsam) installiert, Tests gestartet $\rightarrow$ erfolgreich!

Lösung: Elastic Block Storages

EC2-API Update 20. August 2008:

Einführung der „Elastic Block Storages“ (**EBS**)

- EBS lassen sich starten/beenden wie Instanzen
- permanente Speicher, unabhängig von Instanzen
- EBS kann an genau eine Instanz gekoppelt und ins Dateisystem eingebunden werden
- EBS existiert bei Instanzbeendigung weiter
- Backup: „EBS-Snapshot“ auf S3 speichern

EBS erstellt, ATLAS Software (mühsam) installiert, Tests gestartet $\rightarrow$ erfolgreich!

Lösung: Elastic Block Storages

EC2-API Update 20. August 2008:

Einführung der „Elastic Block Storages“ (**EBS**)

- EBS lassen sich starten/beenden wie Instanzen
- permanente Speicher, unabhängig von Instanzen
- EBS kann an genau eine Instanz gekoppelt und ins Dateisystem eingebunden werden
- EBS existiert bei Instanzbeendigung weiter
- Backup: „EBS-Snapshot“ auf S3 speichern

EBS erstellt, ATLAS Software (mühsam) installiert, Tests gestartet $\rightarrow$ erfolgreich!

Lösung: Elastic Block Storages

EC2-API Update 20. August 2008:

Einführung der „Elastic Block Storages“ (**EBS**)

- EBS lassen sich starten/beenden wie Instanzen
- permanente Speicher, unabhängig von Instanzen
- EBS kann an genau eine Instanz gekoppelt und ins Dateisystem eingebunden werden
- EBS existiert bei Instanzbeendigung weiter
- Backup: „EBS-Snapshot“ auf S3 speichern

EBS erstellt, ATLAS Software (mühsam) installiert, Tests gestartet $\rightarrow$ erfolgreich!

Lösung: Elastic Block Storages

EC2-API Update 20. August 2008:

Einführung der „Elastic Block Storages“ (**EBS**)

- EBS lassen sich starten/beenden wie Instanzen
- permanente Speicher, unabhängig von Instanzen
- EBS kann an genau eine Instanz gekoppelt und ins Dateisystem eingebunden werden
- EBS existiert bei Instanzbeendigung weiter
- Backup: „EBS-Snapshot“ auf S3 speichern

EBS erstellt, ATLAS Software (mühsam) installiert, Tests gestartet $\rightarrow$ erfolgreich!

Lösung: Elastic Block Storages

EC2-API Update 20. August 2008:

Einführung der „Elastic Block Storages“ (**EBS**)

- EBS lassen sich starten/beenden wie Instanzen
- permanente Speicher, unabhängig von Instanzen
- EBS kann an genau eine Instanz gekoppelt und ins Dateisystem eingebunden werden
- EBS existiert bei Instanzbeendigung weiter
- Backup: „EBS-Snapshot“ auf S3 speichern

EBS erstellt, ATLAS Software (mühsam) installiert, Tests gestartet $\rightarrow$ erfolgreich!

Lösung: Elastic Block Storages

EC2-API Update 20. August 2008:

Einführung der „Elastic Block Storages“ (**EBS**)

- EBS lassen sich starten/beenden wie Instanzen
- permanente Speicher, unabhängig von Instanzen
- EBS kann an genau eine Instanz gekoppelt und ins Dateisystem eingebunden werden
- EBS existiert bei Instanzbeendigung weiter
- Backup: „EBS-Snapshot“ auf S3 speichern

EBS erstellt, ATLAS Software (mühsam) installiert, Tests gestartet $\rightarrow$ erfolgreich!

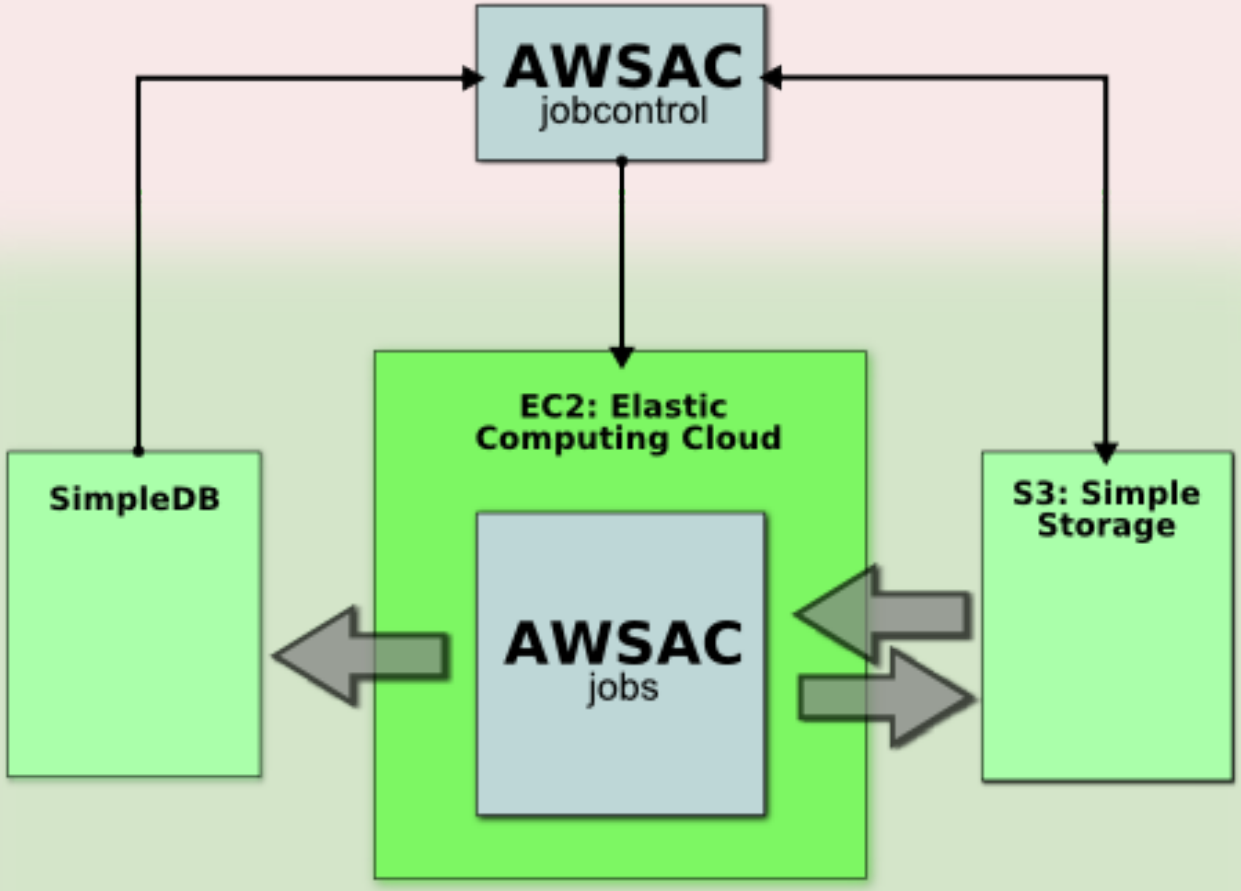
Übersicht

flexibles, benutzerfreundliches Jobsystem für AC mit AWS

Was brauche ich? Einen Plan:

flexibles, benutzerfreundliches Jobsystem für AC mit AWS

Was brauche ich? Einen Plan:



Wie den Plan umsetzen?

Wie steuert man alle AWS (EC2, S3, SDB, ...)?

AWS werden über HTTP Protokoll gesteuert:

- spezieller *HTTP request* an https://aws.amazon.com
- in *HTTP response* ist u.a. Bestätigung oder Fehler notiert

Einfachste Umsetzung: *HTTP GET Request*

Schematischer Beispiel-Request:
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-60a54009&InstanceCount=1&InstanceType=c1.middle ...
&...auth parameters...

Wie den Plan umsetzen?

Wie steuert man alle AWS (EC2, S3, SDB, ...)?

AWS werden über HTTP Protokoll gesteuert:

- spezieller *HTTP request* an https://aws.amazon.com
- in *HTTP response* ist u.a. Bestätigung oder Fehler notiert

Einfachste Umsetzung: *HTTP GET Request*

Schematischer Beispiel-Request:
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-60a54009&InstanceCount=1&InstanceType=c1.middle ...
&...auth parameters...

Wie den Plan umsetzen?

Wie steuert man alle AWS (EC2, S3, SDB, ...)?

AWS werden über HTTP Protokoll gesteuert:

- spezieller *HTTP request* an https://aws.amazon.com
- in *HTTP response* ist u.a. Bestätigung oder Fehler notiert

Einfachste Umsetzung: *HTTP GET Request*

Schematischer Beispiel-Request:
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-60a54009&InstanceCount=1&InstanceType=c1.middle ...
&...auth parameters...

Wie den Plan umsetzen?

Wie steuert man alle AWS (EC2, S3, SDB, ...)?

AWS werden über HTTP Protokoll gesteuert:

- spezieller *HTTP request* an https://aws.amazon.com
- in *HTTP response* ist u.a. Bestätigung oder Fehler notiert

Einfachste Umsetzung: *HTTP GET Request*

Schematischer Beispiel-Request:
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-60a54009&InstanceCount=1&InstanceType=c1.middle ...
&...auth parameters...

Wie den Plan umsetzen?

Wie steuert man alle AWS (EC2, S3, SDB, ...)?

AWS werden über HTTP Protokoll gesteuert:

- spezieller *HTTP request* an https://aws.amazon.com
- in *HTTP response* ist u.a. Bestätigung oder Fehler notiert

Einfachste Umsetzung: *HTTP GET Request*

Schematischer Beispiel-Request:
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-60a54009&InstanceCount=1&InstanceType=c1.middle ...
&...auth parameters...

Wahl der Programmiersprache

Jede moderne Programmiersprache kann HTTP
(Java, Ruby, PHP, Python, ...)

Meine Wahl: Python

- meine Lieblingsprogrammiersprache
- opensource Modul, welches fast alle AWS recht stabil unterstützt:
  boto

Wahl der Programmiersprache

Jede moderne Programmiersprache kann HTTP
(Java, Ruby, PHP, Python, ...)

Meine Wahl: Python

- meine Lieblingsprogrammiersprache
- opensource Modul, welches fast alle AWS recht stabil unterstützt: **boto**

Wahl der Programmiersprache

Jede moderne Programmiersprache kann HTTP
(Java, Ruby, PHP, Python, ...)

Meine Wahl: Python

- meine Lieblingsprogrammiersprache
- opensource Modul, welches fast alle AWS recht stabil unterstützt: **boto**

Wahl der Programmiersprache

Jede moderne Programmiersprache kann HTTP
(Java, Ruby, PHP, Python, ...)

Meine Wahl: Python

- meine Lieblingsprogrammiersprache
- opensource Modul, welches fast alle AWS recht stabil unterstützt: **boto**

Was ist ein **Job**?

Meine Definition:

Ein Job ist ein als *root* ausgeführtes Shellscript, welches als eigener Prozess auf einer Instanz läuft. Das Script wird vom Benutzer selbst geliefert. Innerhalb dieses Scriptes können die Befehle eines selbstgewählten ATLAS Software Releases benutzt werden.

zusätzlich möglich:

- user kann beliebige Dateien in das System geben
- user kann definieren, welche Dateien er zurückbekommen möchte

→ Der Benutzer kann auf der VM alles machen, was er möchte.

Was ist ein **Job**?

Meine Definition:

Ein Job ist ein als *root* ausgeführtes Shellscript, welches als eigener Prozess auf einer Instanz läuft. Das Script wird vom Benutzer selbst geliefert. Innerhalb dieses Scriptes können die Befehle eines selbstgewählten ATLAS Software Releases benutzt werden.

zusätzlich möglich:

- user kann beliebige Dateien in das System geben
- user kann definieren, welche Dateien er zurückbekommen möchte

$\longrightarrow$ Der Benutzer kann auf der VM alles machen, was er möchte.

Was ist ein **Job**?

Meine Definition:

Ein Job ist ein als *root* ausgeführtes Shellscript, welches als eigener Prozess auf einer Instanz läuft. Das Script wird vom Benutzer selbst geliefert. Innerhalb dieses Scriptes können die Befehle eines selbstgewählten ATLAS Software Releases benutzt werden.

zusätzlich möglich:

- user kann beliebige Dateien in das System geben
- user kann definieren, welche Dateien er zurückbekommen möchte

→ Der Benutzer kann auf der VM alles machen, was er möchte.

Was ist ein **Job**?

Meine Definition:

Ein Job ist ein als *root* ausgeführtes Shellscript, welches als eigener Prozess auf einer Instanz läuft. Das Script wird vom Benutzer selbst geliefert. Innerhalb dieses Scriptes können die Befehle eines selbstgewählten ATLAS Software Releases benutzt werden.

zusätzlich möglich:

- user kann beliebige Dateien in das System geben
- user kann definieren, welche Dateien er zurückbekommen möchte

→ Der Benutzer kann auf der VM alles machen, was er möchte.

Was ist ein **Job**?

Meine Definition:

Ein Job ist ein als *root* ausgeführtes Shellscript, welches als eigener Prozess auf einer Instanz läuft. Das Script wird vom Benutzer selbst geliefert. Innerhalb dieses Scriptes können die Befehle eines selbstgewählten ATLAS Software Releases benutzt werden.

zusätzlich möglich:

- user kann beliebige Dateien in das System geben
- user kann definieren, welche Dateien er zurückbekommen möchte

$\rightarrow$ Der Benutzer kann auf der VM alles machen, was er möchte.

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

Entstanden sind die **AWSACtools**

awsac-session (-start | -check | -getresults):

- benutzt der Anwender
- bekommt Job-Shellscripts und andere individuelle Daten
- Aufgaben: Jobsessions starten, Job-Monitoring (Informationen aus SimpleDB), Resultate empfangen (von S3)

awsac-autorun:

- in SL4 AMI eingebunden und in Systemstart jeder Instanz integriert
- Aufgaben: empfängt Informationen über Jobsession und startet lange Befehlskette

awsac-processjobs:

- ist Teil der Befehlskette in den Instanzen einer Jobsession
- Aufgaben: führt Jobs aus, setzt Stati (in SimpleDB), verwaltet EBS, speichert Resultatdaten (auf S3), beendet Instanzen

einige Bemerkungen zu **AWSACtools**

Quelltexte und genaue Funktionsweise: http://gehrcke.de/awsac
Testläufe (Event Simulation) funktionierten reibungslos!

erwähnenswert:

- *awsac-session* braucht nur Python → portabel
- Anzahl Jobs auf einer Instanz entspricht Anzahl virtueller Cores
- *awsac-processjobs* nicht in AMI hardgecoded
- *awsac-session* empfängt Logs von Job-Shellscripten und von *awsac-processjobs*

einige Bemerkungen zu **AWSACtools**

Quelltexte und genaue Funktionsweise: http://gehrcke.de/awsac
Testläufe (Event Simulation) funktionierten reibungslos!

erwähnenswert:

- *awsac-session* braucht nur Python → portabel
- Anzahl Jobs auf einer Instanz entspricht Anzahl virtueller Cores
- *awsac-processjobs* nicht in AMI hardgecoded
- *awsac-session* empfängt Logs von Job-Shellscripten und von *awsac-processjobs*

einige Bemerkungen zu **AWSACtools**

Quelltexte und genaue Funktionsweise: http://gehrcke.de/awsac
Testläufe (Event Simulation) funktionierten reibungslos!

erwähnenswert:

- *awsac-session* braucht nur Python → portabel
- Anzahl Jobs auf einer Instanz entspricht Anzahl virtueller Cores
- *awsac-processjobs* nicht in AMI hardgecoded
- *awsac-session* empfängt Logs von Job-Shellscripten und von *awsac-processjobs*

einige Bemerkungen zu **AWSACtools**

Quelltexte und genaue Funktionsweise: http://gehrcke.de/awsac
Testläufe (Event Simulation) funktionierten reibungslos!

erwähnenswert:

- *awsac-session* braucht nur Python → portabel
- Anzahl Jobs auf einer Instanz entspricht Anzahl virtueller Cores
- *awsac-processjobs* nicht in AMI hardgecoded
- *awsac-session* empfängt Logs von Job-Shellscripten und von *awsac-processjobs*

einige Bemerkungen zu **AWSACtools**

Quelltexte und genaue Funktionsweise: http://gehrcke.de/awsac
Testläufe (Event Simulation) funktionierten reibungslos!

erwähnenswert:

- *awsac-session* braucht nur Python → portabel
- Anzahl Jobs auf einer Instanz entspricht Anzahl virtueller Cores
- *awsac-processjobs* nicht in AMI hardgecoded
- *awsac-session* empfängt Logs von Job-Shellscripten und von *awsac-processjobs*

einige Bemerkungen zu **AWSACtools**

Quelltexte und genaue Funktionsweise: http://gehrcke.de/awsac
Testläufe (Event Simulation) funktionierten reibungslos!

erwähnenswert:

- *awsac-session* braucht nur Python → portabel
- Anzahl Jobs auf einer Instanz entspricht Anzahl virtueller Cores
- *awsac-processjobs* nicht in AMI hardgecoded
- *awsac-session* empfängt Logs von Job-Shellscripten und von *awsac-processjobs*

Übersicht

Zusammenfassung und Fazit

Zusammenfassung

- Cloud Computing mit VMs einfacher/robuster als LCG-Ansatz
- nach einiger Vorarbeit: EC2 ist für ATLAS Computing zu benutzen
- **AWSACtools** zeigen: mit AWS lässt sich ein flexibles, robustes und benutzerfreundliches Jobsystem aufbauen

Fazit

$\longrightarrow$ Die Idee „Cloud Computing mit VMs für ATLAS/LHC“ sollte weitergedacht werden.

Zusammenfassung und Fazit

Zusammenfassung

- Cloud Computing mit VMs einfacher/robuster als LCG-Ansatz
- nach einiger Vorarbeit: EC2 ist für ATLAS Computing zu benutzen
- **AWSACtools** zeigen: mit AWS lässt sich ein flexibles, robustes und benutzerfreundliches Jobsystem aufbauen

Fazit

→ Die Idee „Cloud Computing mit VMs für ATLAS/LHC“ sollte weitergedacht werden.

Zusammenfassung und Fazit

Zusammenfassung

- Cloud Computing mit VMs einfacher/robuster als LCG-Ansatz
- nach einiger Vorarbeit: EC2 ist für ATLAS Computing zu benutzen
- **AWSACtools** zeigen: mit AWS lässt sich ein flexibles, robustes und benutzerfreundliches Jobsystem aufbauen

Fazit

→ Die Idee „Cloud Computing mit VMs für ATLAS/LHC“ sollte weitergedacht werden.

Zusammenfassung und Fazit

Zusammenfassung

- Cloud Computing mit VMs einfacher/robuster als LCG-Ansatz
- nach einiger Vorarbeit: EC2 ist für ATLAS Computing zu benutzen
- **AWSACtools** zeigen: mit AWS lässt sich ein flexibles, robustes und benutzerfreundliches Jobsystem aufbauen

Fazit

→ Die Idee „Cloud Computing mit VMs für ATLAS/LHC“ sollte weitergedacht werden.

Zusammenfassung und Fazit

Zusammenfassung

- Cloud Computing mit VMs einfacher/robuster als LCG-Ansatz
- nach einiger Vorarbeit: EC2 ist für ATLAS Computing zu benutzen
- **AWSACtools** zeigen: mit AWS lässt sich ein flexibles, robustes und benutzerfreundliches Jobsystem aufbauen

Fazit

→ Die Idee „Cloud Computing mit VMs für ATLAS/LHC“ sollte weitergedacht werden.

Ausblick

nächste Schritte

- Online-Dokumentation fertigstellen (für München, ATLAS Computing-Leute am CERN, andere ATLAS-Gruppen, AWS-Verantwortliche, Nimbus-Leute)
- Weiterentwicklung der **AWSACtools** und Erstellung eines public AWSAC-AMIs
- $\longrightarrow$ ermöglicht Nutzung des Jobsystems mit verschiedenen AWS-Accounts

Ausblick

nächste Schritte

- Online-Dokumentation fertigstellen (für München, ATLAS Computing-Leute am CERN, andere ATLAS-Gruppen, AWS-Verantwortliche, Nimbus-Leute)
- Weiterentwicklung der **AWSACtools** und Erstellung eines public AWSAC-AMIs
- $\longrightarrow$ ermöglicht Nutzung des Jobsystems mit verschiedenen AWS-Accounts

Ausblick

nächste Schritte

- Online-Dokumentation fertigstellen (für München, ATLAS Computing-Leute am CERN, andere ATLAS-Gruppen, AWS-Verantwortliche, Nimbus-Leute)
- Weiterentwicklung der **AWSACtools** und Erstellung eines public AWSAC-AMIs
- $\longrightarrow$ ermöglicht Nutzung des Jobsystems mit verschiedenen AWS-Accounts

Ausblick

nächste Schritte

- Online-Dokumentation fertigstellen (für München, ATLAS Computing-Leute am CERN, andere ATLAS-Gruppen, AWS-Verantwortliche, Nimbus-Leute)
- Weiterentwicklung der **AWSACtools** und Erstellung eines public AWSAC-AMIs
- $\longrightarrow$ ermöglicht Nutzung des Jobsystems mit verschiedenen AWS-Accounts

Ausblick

Nimbus:

- „**Nimbus** provides a free, open source infrastructure [...], allowing you to [...] make your own EC2 style service“
- entwickelt von einer Arbeitsgruppe am Argonne National Laboratory, Mathematics and Computer Science Division
- Kontakt über AWS hergestellt
- Gruppe interessiert an **AWSACtools**

→ EC2 zum Preis eigener Rechner kann mit **Nimbus** bald Realität sein!
AWSACtools können mit **Nimbus** zusammenarbeiten!

Ausblick

Nimbus:

- „**Nimbus** provides a free, open source infrastructure [...], allowing you to [...] make your own EC2 style service“
- entwickelt von einer Arbeitsgruppe am Argonne National Laboratory, Mathematics and Computer Science Division
- Kontakt über AWS hergestellt
- Gruppe interessiert an **AWSACtools**

→ EC2 zum Preis eigener Rechner kann mit **Nimbus** bald Realität sein!
AWSACtools können mit **Nimbus** zusammenarbeiten!

Ausblick

Nimbus:

- „**Nimbus** provides a free, open source infrastructure [...], allowing you to [...] make your own EC2 style service“
- entwickelt von einer Arbeitsgruppe am Argonne National Laboratory, Mathematics and Computer Science Division
- Kontakt über AWS hergestellt
- Gruppe interessiert an **AWSACtools**

→ EC2 zum Preis eigener Rechner kann mit **Nimbus** bald Realität sein!
AWSACtools können mit **Nimbus** zusammenarbeiten!

Ausblick

Nimbus:

- „**Nimbus** provides a free, open source infrastructure [...], allowing you to [...] make your own EC2 style service“
- entwickelt von einer Arbeitsgruppe am Argonne National Laboratory, Mathematics and Computer Science Division
- Kontakt über AWS hergestellt
- Gruppe interessiert an **AWSACtools**

→ EC2 zum Preis eigener Rechner kann mit **Nimbus** bald Realität sein!
AWSACtools können mit **Nimbus** zusammenarbeiten!

Ausblick

Nimbus:

- „**Nimbus** provides a free, open source infrastructure [...], allowing you to [...] make your own EC2 style service“
- entwickelt von einer Arbeitsgruppe am Argonne National Laboratory, Mathematics and Computer Science Division
- Kontakt über AWS hergestellt
- Gruppe interessiert an **AWSACtools**

→ EC2 zum Preis eigener Rechner kann mit **Nimbus** bald Realität sein!
AWSACtools können mit **Nimbus** zusammenarbeiten!

Ausblick

Nimbus:

- „**Nimbus** provides a free, open source infrastructure [...], allowing you to [...] make your own EC2 style service“
- entwickelt von einer Arbeitsgruppe am Argonne National Laboratory, Mathematics and Computer Science Division
- Kontakt über AWS hergestellt
- Gruppe interessiert an **AWSACtools**

→ EC2 zum Preis eigener Rechner kann mit **Nimbus** bald Realität sein!
AWSACtools können mit **Nimbus** zusammenarbeiten!

Literaturverzeichnis


AWS
AWS documentation
http://aws.amazon.com/documentation/


Jan-Philip Gehrcke
AWSAC documentation
http://gehrcke.de/atlas/awsacdev/ (now)
http://gehrcke.de/awsac/ (soon)


ATLAS Groups
ATLAS Computing TWiki
https://twiki.cern.ch/twiki/bin/view/Atlas/AtlasComputing