Vorlesungsreihe
Entwicklung webbasierter Anwendungen

Serverfarmen und Cloud Computing

Prof. Dr.-Ing. Thomas Wiedemann
email: wiedem@informatik.htw-dresden.de

HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT DRESDEN (FH)
Fachbereich Informatik/Mathematik

Gliederung

- Aufbau und Konfiguration von **Serverfarmen**
  - Notwendigkeit und historische Entwicklung des Load Balancing
  - Aktueller Stand
  - Neue Entwicklungen
- **Cloud Computing**
  - Arten und Prinzipien
  - Aktuelle Anbieter und einige Tests

Entwicklung von Serverfarmen und des Load Balancing

Typische Hardwareanforderung beim Aufbau von Webanwendungen

- zu Beginn steht (bei Startups?) das Preis/Leistungsverhältnis im Fokus
- bei Erfolg waren / sind dann schnelle Anpassungen an die höheren Zahlen der User-Requests notwendig
- **Probleme:**
  - **einfache PC-basierte Server können trotz Hauptspeicherausbaus nur in Grenzen in der Antwortkapazität erweitert werden**
  - **stärkere Workstations oder Supercomputer stellen wiederum spezielle Softwareanforderungen oder unterstützen nicht alle Optionen der Anwendungen**

Häufige (und heute typische) Lösung

- **Aufbau von größeren Rechnerclustern auf PC-Hardwarebasis und meist Linux-Betriebssystemen (Serverblades)**
- **Die Anfragen müssen dann möglichst gleichmässig auf die Rechner verteilt werden (= Load Balancing)**

Allgemeine Anforderungen an das Load Balancing

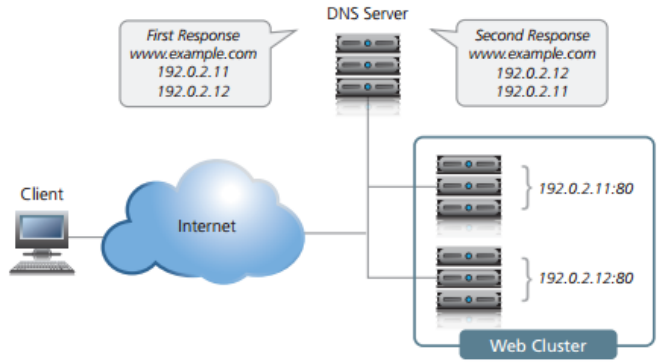
Im Rahmen der Lastverteilung sind folgende Anforderungen zu erfüllen:

- **Hochverfügbarkeit (highly available)**
  - auch bei Ausfall einzelner Server muss die Gesamtfunktionalität der Serverfarm erhalten bleiben
- **Skalierbarkeit (scalable)**
  - Neue Anforderungen an die Leistung müssen einfach und ohne Beeinträchtigung des vorhandenen Systems umgesetzt werden können
- **Voraussagbares, konsistentes Verhalten (predictable)**
  - Das Verhalten der Serverfarm muss aus Sicht der Webapplikationen ohne Unterschiede zu einem Einzelsystem funktionieren und konsistente Ergebnisse bei mehrfachen Aufrufen bringen.

Erste Optionen des Load Balancing

Load Distribution unter Nutzung der DNS-Server

- DNS-Round Robin (pro Aufruf werden mehrere Adressen vom DNS-Server zurückgegeben, dies ergibt ein eher zufälliger Verteilen, kein Balancing)
- funktioniert für kleinere Serverfarmen auch heute noch ganz gut, Probleme können auftreten bei long-term-Sessions und Serverausfällen

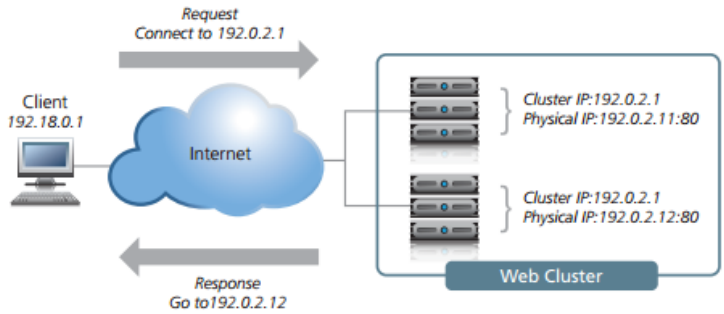


Quelle: http://www.f5.com/pdf/white-papers/

Optionen des Load Balancing

Load Balancing auf Softwarebasis

- Erster Client-Aufruf der Anwendung erfolgt über eine zentrale Cluster-IP,
- der Cluster entscheidet dann in der Anwendungssoftware (oder auf OS-Niveau), welcher Server die geringste Auslastung hat und initiiert einen REDIRECT auf diesen Server.
- Bei geringer Serverzahl (<10) sehr gut, dann anwachsende Probleme mit dem Serverabgleich.

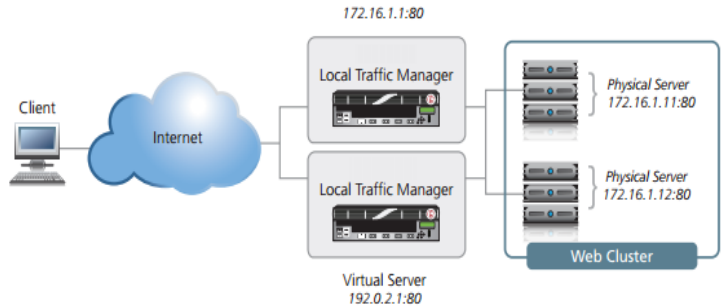


Quelle: http://www.f5.com/pdf/white-papers/

Optionen des Load Balancing

Netzwerk-Infrastruktur-basiertes Load Balancing

- Lastverteilung durch die Netzwerkrouter (spezielle Hardware >Performance)
- mit bidirektionalem Network Address Translation (NAT)
- Verteilung der Anfragen erfolgt durch ständiges Messen der Auslastung und Antwortzeiten und kann auch unterschiedliche Hardwareausbauten in der Serverfarm unterstützen



Quelle: http://www.f5.com/pdf/white-papers/

Aktuelle Entwicklungen

Weiterentwicklung des reinen Load Balancings zum Application Delivery Controller (ADN) mit folgenden Zusatzfunktionalitäten:

- **Health Monitoring:** Gesamtüberwachung der Serverfarm auf Funktionalität und Leistung
- **Dynamic provisioning** : Automatisches Verwalten der verfügbaren Rechnerresourcen (Erweiterung bei Last / Abschaltung zwecks Energiesparen)
- **Intelligentes Caching und Kompression** in Abhängigkeit von den Inhalten
- einheitliche **Authentifizierung** auf dem ADN (spart verteilte Auth. auf den Applikationsservern selbst
- **Location - basiertes Verteilen** der Anfragen (Suche nach dem netzwerkstechnisch "räumlich nächsten" Serverpool
- Bereitstellung von **Content Delivery Networks (CDN) (**vgl. Goggle-Jquery-Bereitstellung oder Ablage hochfrequentierter Presse-Seiten)
- Integration weiterer Services (wie Antiviruskontrolle etc.) in die Hardwareebene

Allgemeine Probleme beim Load Balancing

- Probleme mit Long-term http-Connections :
  - Bei Verwendung von long term Sessions oder größeren Datenoperationen (Download per ftp oder http) müssen die folgenden Requests wieder auf dem gleichem Server erfolgen, damit die Daten ab der aktuellen Stelle verfügbar sind
  - Auch bei permanenten Server-Datenobjekten (DLL oder Java Enterprise) muss eine konstante Zuordnung zum Server erfolgen !
- **Mögliche Lösung:** Speicherung der Quell-IP im Load Balancer und konstante Zuordnung des Clients zum Hosts (dies kann aber wiederum zu Problemen mit Clients hinter einem Proxy geben)
- Beim Ausfällen von Serverhardware muss dies erkannt und ggf. eine erneute Aussendung des Request erfolgen

Beispiele für aktuelle Serverfarmen

(Schätzungen nach http://storageservers.wordpress.com/2013/07/17/facts-and-stats-of-worlds-largest-data-centers/)

- **Google Data Center**
  - ca. 13 weltweit verteilte Data Center mit ca. 900,000 Servern
    (Verbrauch ca. **300 Megawatt** = 0.01% der Welt-Elektroenergie
    = ausreichend für ca. 200,000 Haushalte)
  - In der Regel werden neue Datacenter in der Nähe von Wasserkraftwerken (billige Elektroenergie + Kühlung) gebaut -> Colorado / Finnland
- **Microsoft Data Center**
  - ca. 1.000.000 Server weltweit verteilt (europäischer Node in Dublin Irland)
- **Amazon Data Center**
  - 450,000 Server, davon ca. 40.000 für Cloud-Nutzer (siehe Folgeseiten)

Andere Big-Player (Facebook, Domain-Hoster) verfügen über ähnliche Serverfarmen.

Cloud computing – Einführung und Historie

Entstehung des Cloud Computing

- Mit der Entwicklung großer Internetfirmen wie Amazon, Facebook oder Yahoo entstanden auch sehr große Serverfarmen und in der Folge Probleme bei deren Auslastung
  - Bei Internet-Shops wie Amazon wurde die Hauptlast nur im Weihnachtsgeschäft benötigt. Außerhalb dieser Zeit waren die Server meist nicht ausgelastet (tw. nur 10% Auslastung -> 90% freie Kapazitäten)
  - Idee einer Verwertung der freien Kapazitäten auf dem freien Markt unter dem Slogan „Cloud Server“ ab ca. 2006

- Ursache / Voraussetzung für den Erfolg von Cloud Computing und Cloud Services sind auch die schnellen Internetverbindungen, welche den Unterschied zu einem lokalen Speicher/Rechnersystem stark verringert haben (bzw. nicht mehr sichtbar für Endanwender)

Arten des Cloud Computing

Typen des Cloud Computing nach dem Servicetyp:

- **Cloud Software as a Service (SaaS)**

stellt eine Software zur Nutzung bereit und wird deshalb auch als Software on Demand bezeichnet. Der Anwender muss die Software nicht selbst kaufen und installieren, sondern nutzt diese auf Mietbasis fallweise.

- **Cloud Platform as a Service (PaaS)**

stellt eine Programmier- und/oder Laufzeitplattform zur Verfügung und erlaubt das Entwickeln und Ausführen von (Kunden-) Software auf dieser Plattform.

- **Cloud Infrastructure as a Service (IasS)**

stellt eine Hardwareplattform zur Verfügung. Der Anwender muss selbst die Laufzeit- und Anwendungssoftware installieren und administrieren.

Typen des CC nach Vertraulichkeit und Datenschutz :

- **Public Clouds - meist auf fremden (weit entfernten) Servern**
- **Private Clouds - die Cloud wird innerhalb der eigenen Firma aufgebaut und betrieben** (im Prinzip nur Nutzung der Cloud-Managementsoftware im firmeneigenen Rechenzentrum)
- **Hybrid Clouds - Mischform: unkritische Inhalte auf Public, kritische auf private Cloud**

Allgemeine Vorteile und Nachteile des Cloud Computing

Vorteile :

- sehr hohe Flexibilität bei Performance und techn. Parametern, starke Automat. Der technischen Abläufe
- starke Kosteneinsparungen bei Aufbau und Betrieb von Cloudcomputing-Kapazitäten (Skaleneffekte)
- Professionelle Datensicherheit und hohe Verfügbarkeit
- immer aktuelle Softwarestände

Nachteile :

- bei Public Clouds meist kein direkter (lokaler) Zugriff auf den Rechner, sondern nur über Webmasken oder Remote-Desktop etc.
- starke Abhängigkeit vom Anbieter (Pleite ?, Ausfälle [Blitzschlag Irland!])
- Bei Public clouds Probleme mit Sicherheit / Vertraulichkeit (Geheimnis-schutz nicht 100% gewährleistet !)

Ausführliche BITKOM - Bewertung unter :
http://www.bitkom.org/files/documents/BITKOM-Leitfaden-CloudComputing_Web.pdf

Test des Cloud Computing am Bsp. von Simulationsanw.

Historie und bisherige Ansätze vor ca. 10 Jahren : :

erste Konzepte zur fallweisen Nutzung von Simulationssoftware :

- SIMPC/ GPSS/H –Websimulator der Uni. Magdeburg,
- Lösungen des Fraunhofer Institut Stuttgart (Mail-basiert)
- Simsolution-System des Autors (Application Service Prov.-System)

Allgemeine Bedingungen bei heutigen Cloud-Systemen

- **Cloud-Masseneinsatz statt singulärer Anwendung – statt exotischer ASP-Ansätze nun allgemeine Akzeptanz des Cloud-Gedankens**
- **starke Kosteneinsparungen durch Skalierungseffekte (billigere HW)**
- **jetzt insgesamt höhere Akzeptanz in der Managementebene**
- **breite Bandbreite an technischen Einsatzoptionen und Quasistandards**
- **mit den neuen Angeboten (vgl. Microsoft Azure-Cloud / Amazon Cloud Computing) sind universelle IT-Clouds verfügbar**
- **Hinweis : Die nachfolgenden Tests beziehen sich auf die aktuellen Cloud-Systeme im Sommer 2011. Möglich sind entsprechende Änderungen durch neue Lösungsansätze !**

Potentiale und Grenzen der Simulationsanwendung auf der Cloud

a.) Einsatz von bekannten Simulatoren auf der Cloud

- **Problem: Cloud-Anbieter sind aufgrund von Sicherheitsbedenken sehr restriktiv : fremde EXE-Dateien sind i.d.R. NICHT zugelassen**
  - **damit entfallen fast alle bekannten Simulatoren, da diese nicht im Sourcecode, sondern als Binaries vorliegen**
  - **ggf. können bei intensiven Verhandlungen mit Cloud-Anbietern auch geprüfte Simulatoren als Binaries angeboten werden**
  - **Option B: Einrichtung einer Private Cloud mit Binaries (effektiv?)**

b) Verfügbare, alternative Lösungen auf Public Cloud:
auf Scriptbasis oder Zwischencode operierenden Simulatoren :

- **Java-basierte Simulatoren**
- **.NET – basierte Simulatoren**
- **PHP und Python u.a. (je nach Cloud)**

Test von Cloud Computing – Systemen: Microsoft Azure

Allgemeine Charakteristika :

- **starke Anlehnung auf bekannte Microsoft-Technologien**
  - **Windows Server 2008 R2 als Serverbasis**
  - **Visual Studio 2010 als Entwicklungsplattform**
  - **damit ALLE VS-Sprachen der .NET-Familie als Basis für Applikationen (C#, Visual Basic, ASP.NET, etc)**

Erste Einsatzerfahrungen

- **Schneller Einstieg durch gute Dokumentationen und Beispiele**
- **mittelmäßige Managementumgebung : intuitiv bedienbar, aber relativ langsam und nicht sehr komfortabel, kein Single-Site-Login zum Einblick in die Vertrags- und Rechnungsdaten**
- **unterdurchschnittliche Leistungswerte der einfachen Serveroptionen, vergleichbar mit ca. 1,6 Ghz – PC !**
- **Schlechtes Bezahlmodell : Berechnung auch im Standby-Modus**

Test von Cloud Computing – Systemen: Microsoft Azure II

Softwaremodule:

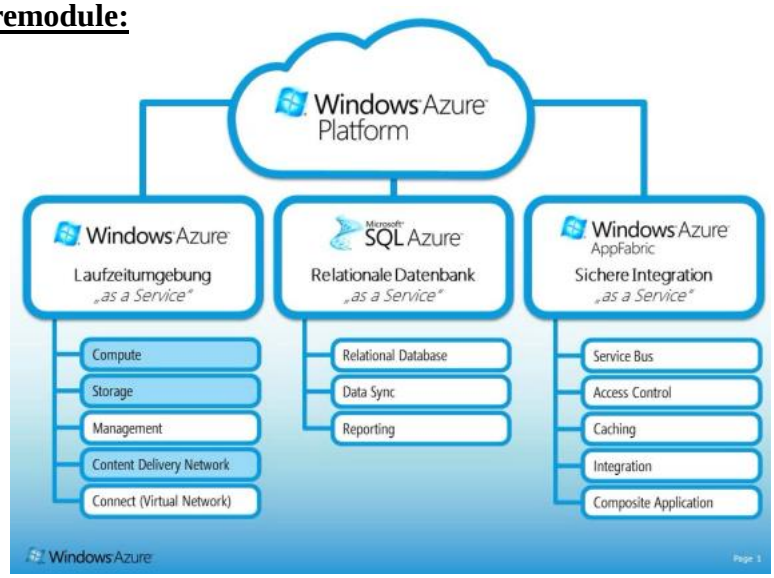


Abb. - Quelle: http://www.microsoft.com/de-de/azure/entwickeln/Optimierung-mit-CDN.aspx

Test von Cloud Computing – Systemen: Microsoft Azure II

Bereitstellung und Upload DIREKT aus VS 2010

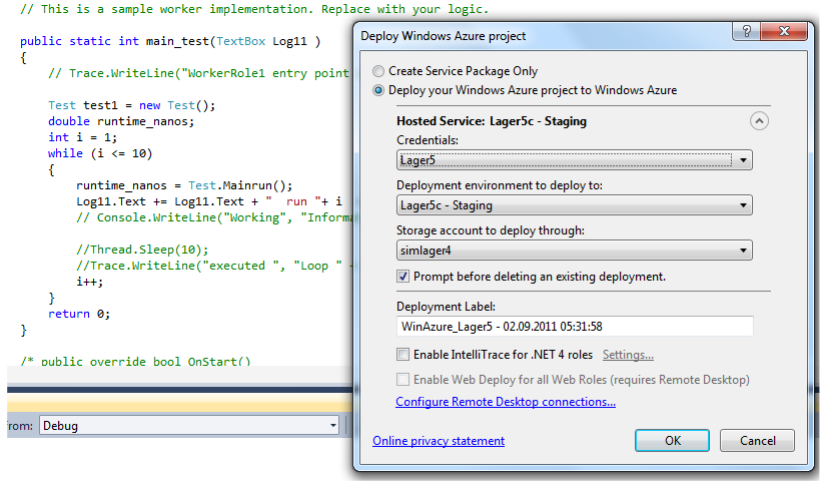
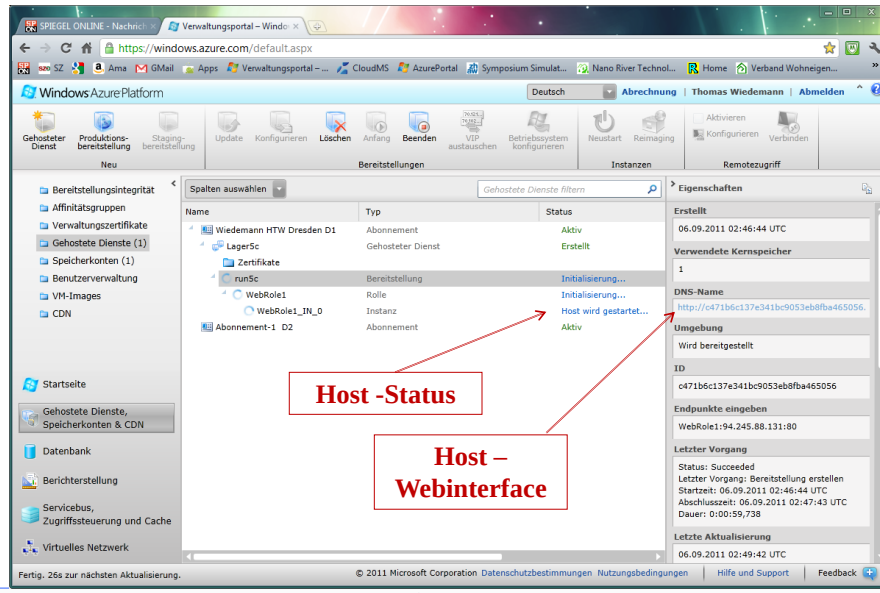


Abb.: Bereitstellung und Upload DIREKT aus VS 2010

Test von Cloud Computing – Systemen: Microsoft Azure III

Managementconsole



Test : Google App Engine

Allgemeine Charakteristika : :

- **starke Orientierung auf freie Tools**
  - **unterstützt Java, Python, PHP und GO**

Erste Einsatzerfahrungen

- **sehr schneller Einstieg durch gute Dokumentationen und Beispiele**
- **mittelmäßige Managementumgebung : intuitiv bedienbar, graphische Darstellung der Werte, teilweise sehr verzögert**
- **sehr schneller Upload der App durch Eclipse-Plugin**
- **(In 2011 tw. noch unterschiedliche Leistungswerte, da manchmal 200ms, teilweise aber auch 4fache Zeit → Performance sehr unterschiedlich)**

Test : Google App Engine

Aktuelle Preise (2015 / 01) :

Service	Kostenloses Kontingent pro App pro Tag	Preise bei Überschreitung des kostenlosen Kontingents
Instanzen	28 Instanzstunden	0,05 $/Instanz/Stunde
Cloud Datastore (NoSQL-Datenbank)	50.000 (Lesen/Schreiben/geringer Umfang) 1 GB Speicher	0,06 $/100.000 (Lesen oder Schreiben) Kleine Operationen kostenlos* 0,18 $/GB/Monat
Ausgehender Netzwerkverkehr	1 GB	0,12 $/GB
Eingehender Netzwerkverkehr	1 GB	Kostenlos
Cloud Storage	5 GB	0,026 $/GB/Monat
Memcache	Kostenlose Nutzung des freigegebenen Pools Kein kostenloses Kontingent für den dedizierten Pool	Kostenlose Nutzung des freigegebenen Pools Dedizierter Pool: 0,06 $/GB/Stunde
Suchen	1.000 Basisoperationen 0,01 GB Dokumentindexierung 0,25 GB Dokumentenspeicherung 100 Suchanfragen	0,50 $/10.000 Suchanfragen 2,00 $/GB Dokumentindexierung 0,18 $/GB/Monat Speicherung
Email API	100 Empfänger	Vertrieb kontaktieren
Logs API	100 MB	0,12 $ pro GB
Aufgaben-Warteschlange und Protokollspeicherung	5 GB 1 GB	0,026 $/GB/Monat
Virtuelle IP-Adressen (SSL)	Kein kostenloses Kontingent	39,00 $/virtuelle IP-Adresse pro Monat

Test : Amazon Elastic Compute Cloud

Allgemeine Charakteristika : :

- **starke Orientierung auf das Amazon-Shop-Geschäft**
  - **Linux und Windows-basiert, (und Amazon-eigenes Betriebssystem)**
  - **Unterstützt v.a. .NET, Java, aber auch Ruby, Python**

Erste Einsatzerfahrungen

- **sehr umfangreich, da keine Funktionseinschränkungen und komplettes Amazon Cloud Angebot nutzbar :**
  - **Amazon Elastic Cloud (variable Rechenleistung)**
  - **Amazon S3 Simple Storage Service**
  - **AWS Lambda on Demand Verarbeitungsservice**
    (nur die reine Laufzeit nach einem Ereignis wird berechnet)
- **Aktuelle Preise für alle Dienste (mit komplexem Rabattsystem unter http://aws.amazon.com/de/ec2/pricing/)**
- **gute Managementumgebung : intuitiv bedienbar, komfortabel, anfangs unübersichtlich, guter Einblick in Vertrags- und Rechnungsdaten**

Vergleich der Kosten

Generell gilt: **Die aktuellen Preismodelle sind stark im Wandel, teilweise undurchsichtig und schlecht vergleichbar und können sich je nach Konkurrenzsituation schnell ändern (gewisse Unsicherheit und Nachteil gegenüber selbst betriebenen Rechnercluster ..)**

- **Die Gesamtkosten ergeben sich aus**

  Preis_pro_h * Anzahl_Instanzen * Anzahl_h

  + Preis_pro_Gigabyte_Datenspeicher * Datenablage

  + Preis_pro_Gigabyte_Datentraffic * DatenIO

  + Preis_für_Sonderoptionen

Aktuelle Werte sind :	**Minimal**	**Typisch**	**Maximal**
Preis_pro_h :	**$0,095**	**$0,38**	**$2,28**
Preis_pro_Gbyte:	**<1Gb free**	**$0,11 / GB**	
Preis_Traffic	**<1Gb free**	**$0,12/ GB**	

Zum Vergleich : Die reinen Stromkosten für eine eigene Rechnerstunde (Annahme 500 W) liegen bei ca. 0,12€ / Stunde !

Einsatzerfahrungen zu aktuellen Cloud-Systemen

Cloud Computing im Simulationsbereich

- Prinzipiell sinnvoll aus Sicht von Kosteneinsparungen und zum Abdecken von Lastspitzen (z.B. 20 x Sim / Pause / 100 x Sim)
- Die allgemein wachsende Akzeptanz in der IT und im Management ist auch im Simulationsbereich positiv bemerkbar !

Technische Probleme und Restriktionen

- (noch) keine Lauffähigkeit von COTS-Systemen
- scriptbasierte Simulatoren (.NET / Java) möglich
- Clouds von MS / Amazon und Google anwendbar, aber optimierungswürdig !I
- **Datenschutz, Sicherheit und Vertrauenswürdigkeit noch nicht abschliessend geklärt** -> Private oder Hybrid Cloud als Lösungsalternative !!?
- NSA-Affäre w#re zumindest in D. /Europa kontraproduktiv !

Nach Klärung der offenen technischen Fragen der Sicherheitsprobleme könnten Cloud-Lösungen eine interessante Option für die Modellierung und Simulation sein !