

## ***Cluster4All***

-

# **Erweiterung des Linux-basierten Clustercomputings in Microsoft Umgebungen**

Ferdinand Geier, Michael Rauh

ParTec Cluster Competence Center GmbH, München, Deutschland

### **Abstract:**

Linux compute clusters are an accepted platform for parallel applications. This paper describes how to temporarily transform an already existing PC based office environment into a powerful and cost-effective compute cluster environment, based on Linux.

### **Keywords:**

Compute cluster, Linux, Windows, parallel Applications

## 1 Motivation

Rechencluster sind heutzutage preiswerte und trotzdem leistungsfähige Plattformen für rechenintensive Anwendungen. Viele große und mittelständische Firmen setzen diese Technologie mittlerweile ein. Aus Kostengründen basieren diese Cluster typischerweise auf Standard Server Hardware (COTS) und Linux als Betriebssystem. Eingebunden in den komplexen Workflow großer Firmen sind auch Zulieferfirmen bis hin zum kleinen Ingenieurbüro heute gezwungen, entsprechende Berechnungsumgebungen zu betreiben.

Der Grundgedanke hinter *Cluster4All* ist es, dem Kunden die Rechenleistung, die er eigentlich schon in Form von Arbeitsplatzrechnern zur Verfügung hat, effizient nutzen zu lassen. Typischerweise werden von Firmen im CAD-/CAE- Umfeld Windows basierte Workstations mit hoher Leistung benutzt. Diese sind notwendig, um die komplexen Modelle und Simulationen in endlicher Zeit zu designen, zu berechnen und/oder zu visualisieren. Zusätzlich existieren auch noch Windows Computer, die für generelle Büroaufgaben eingesetzt werden.

Alle diese Systeme sind vornehmlich nur während der regulären Arbeitszeit in Benutzung. Nachts, an Wochenenden und an Feiertagen bleiben Sie unbenutzt. Dabei könnten all diese Computer, wenn sie zu einem Cluster verbunden werden, ihre kumulierte Rechenleistung dazu nutzen, komplexe Simulationen zu berechnen.

Das Ziel von *Cluster4All* ist es aber keineswegs, einen „echten“ Cluster mit spezieller Hardware zu ersetzen. Die Performance einer *Cluster4All*-basierten Lösung wird nicht mit der eines dedizierten Clusters vergleichbar sein. Vielmehr soll dem Benutzer ein einfacher, und kostengünstiger Einstieg in die Welt des Clustercomputing ermöglicht werden. Vor allem für diejenigen Kunden, die bisher die Anschaffung eines „echten“ kostenintensiven Clusters gescheut haben, soll dies eine interessante Alternative darstellen. Die Kostenersparnis beim Einsatz von *Cluster4All* ist deshalb so groß, weil sich der Kunde keine neue Hardware anschaffen muss, sondern seine bestehende Infrastruktur mit Hilfe von *Cluster4All* besser ausnutzen kann. Ergänzend können die Arbeitsplatzrechner in ein bereits bestehendes Rechencluster integriert werden.

Im Rahmen einer Diplomarbeit wurde bei der Fa. ParTec Cluster Competence Center GmbH ein Konzept erarbeitet, das es erlaubt, die Anforderungen des Clustercomputings mit den Vorgaben einer typischen Büro-Arbeitsplatzumgebung zu kombinieren. Wesentliche Teile des Konzepts wurden evaluiert und als Prototyp realisiert. Mittels einiger typischer LS-Dyna Jobs wurden erste Erfahrungen, auch bezüglich der erreichbaren Performance, gesammelt.

## 2 Konzept

Das generelle Konzept von *Cluster4All* besteht nun darin, ein Linux-basiertes Cluster auf die installierten Workstations oder Arbeitsplatzrechner auszudehnen. Generell übernehmen dabei die Workstations die Rolle von Rechenknoten, haben also keinerlei administrative oder verwaltungstechnische Aufgaben.

### 2.1 Aufgaben und Probleme

Bei der Realisierung ergeben sich schnell einige prinzipielle Aufgabenstellungen und Probleme, die es zu betrachten gilt:

#### 2.1.1 Betriebssystem

Die einzelnen Prozesse eines parallelen Jobs müssen aus mehrerlei Gründen mit identischen Binärprogrammen gestartet werden. Die installierten Arbeitsplatzrechner verwenden nun typischerweise Windows als Betriebssystem, wohin gegen die Rechenknoten unter Linux laufen sollen<sup>1</sup>.

#### 2.1.2 Benutzung als Arbeitsplatzrechner

Die Verwendung als Arbeitsplatzrechner ist die primäre Aufgabe des jeweiligen Rechners. Der Start von Rechenjobs, also die Benutzung des Arbeitsplatzrechners als Rechenknoten innerhalb eines

---

<sup>1</sup> Linux ist heutzutage die bevorzugte Plattform für Rechencluster und wird in der überwiegenden Mehrzahl der installierten Systeme benutzt.

Clusters, sollte nur erfolgen, falls das System nicht benutzt wird. Der Term „nicht benutzt“ muss hierbei noch genauer definiert werden.

### 2.1.3 Heterogene Umgebung

Sehr wahrscheinlich sind die verfügbaren Arbeitsplatzrechner sehr heterogen in Bezug auf Prozessorleistung und –Architektur, Arbeitsspeicher, verfügbarer Plattenplatz, usw. Bei der Verteilung von parallelen Prozessen muss diese Heterogenität berücksichtigt werden, ansonsten können Inkompatibilitäten und inakzeptable Leistungsprobleme auftreten. Das System muss also, im Gegensatz zu reinen Clusterlösungen, mit einer sehr inhomogenen Rechnerlandschaft zurechtkommen.

### 2.1.4 Interconnect

Typischerweise sind die Arbeitsplatzrechner mit 100MB Ethernet verbunden. Dies kann zu einem gravierenden Missverhältnis zwischen Rechenleistung und Netzwerkleistung führen. Rechencluster erfordern aus diesem Grund häufig ein schnelleres Netzwerk.

### 2.1.5 Vorhandene Cluster-Infrastruktur

In vielen Fällen ist bereits ein Rechencluster vorhanden. *Cluster4All* sollte die zusätzlichen Rechenknoten transparent in dieses Cluster integrieren. Das Job-Management sollte – zumindest aus der Sicht des Anwenders – in beiden Fällen identisch sein. Er sollte also mit denselben Tools und Programmen seine Rechenjobs starten können, unabhängig davon, wo sie letztendlich ablaufen.

Auch aus Administratorsicht wäre es sehr wünschenswert, *Cluster4All* nur als Erweiterung eines bestehenden Clusters zu sehen. Gerade im betrachteten Marktsegment stehen häufig keine Vollzeit-Administratoren zur Verfügung, so dass mit möglichst wenig Spezialwissen –die Administration möglich sein sollte.

Dies alles zeigt, dass ein System wie *Cluster4All* in einem sehr dynamischen Umfeld zurechtkommen muss, ganz im Gegensatz zu einem typischen Rechencluster. Bei diesem ist die Konfiguration statisch vorgegeben und Knoten können höchstens aufgrund von Hardware-Ausfällen verschwinden. Bei *Cluster4All* ist jedoch jederzeit damit zu rechnen, dass Rechenknoten plötzlich nicht mehr zur Verfügung stehen oder wieder hinzukommen.

## 3 Realisierung

Für die Realisierung wurden die oben genannten Anforderungen und Probleme analysiert und die Lösungen prototypisch realisiert

### 3.1 Betriebssystem

Generell sollte eine Linux-Umgebung auf den Rechenknoten bereitgestellt werden. Dazu sind verschiedene Ansätze denkbar:

- Parallele Installation auf einer separaten Partition,
- Cygwin (eine Linux-ähnlichen Umgebung für Windows), oder eine
- Virtuelle Maschine

Da der Computer jederzeit und ohne lange Wartezeiten sofort wieder für den Benutzer verfügbar sein soll, ist die Reboot-Lösung nicht zufriedenstellend. Bevor der Benutzer wieder mit dem Computer arbeiten kann, müsste hier Linux heruntergefahren werden und Windows erneut gestartet werden. Das nimmt selbst bei schnellen Computern zu viel Zeit in Anspruch, als das diese Lösung akzeptabel wäre. Umgekehrt muss der Anwender nach getaner Arbeit daran denken, den Computer mit Linux neu zu starten, so dass *Cluster4All* zur Verfügung steht. Dies bedeutet zusätzlichen Aufwand für den Benutzer und Nachteile für *Cluster4All*, da immer die Möglichkeit besteht, dass der Benutzer den Neustart des Systems vergisst. Auch braucht man für die separate Linux-Installation eine eigene Partition. Zusätzlich ist bei diesen Rechnern typischerweise die gesamte Festplatte in Benutzung und somit müsste der PC erweitert werden. Der Einbau einer neuen Festplatte kostet jedoch zusätzlich Geld und stellt einen Eingriff in das bestehende System dar. Der Vorteil dieser Lösung läge allerdings darin, dass die bestehende Software-Installation (Microsoft Windows) unverändert bleibt.

Die zweite Möglichkeit wäre, die Anwendung mit Hilfe von Cygwin unter Windows auszuführen. Die Installation von Cygwin greift aber tief in das Betriebssystem ein. Auch ist der Installationsaufwand hoch und die Installation selbst umständlich. Es gibt keinen vergleichbaren Support wie bei einer großen Linux Distribution.

Auch ist nur eine eingeschränkte Auswahl von Programmen verfügbar. Vor allem kommerzielle Anwendungen sind für dieses Umfeld nicht erhältlich. Aus diesen Gründen scheidet Cygwin als Plattform aus.

Die von uns gewählte Lösung benutzt eine virtuelle Maschine, in der Linux gleichzeitig neben Windows läuft. Diese virtuelle Maschine stellt für das Gastbetriebssystem einen kompletten Rechner dar. Dabei ist Windows das Hostbetriebssystem und Linux das Gastbetriebssystem. Es gibt verschiedene Lösungen auf dem Markt, die diesen Zweck erfüllen:

- VMware Produkte
- Windows Virtual PC
- Parallels Workstation
- QEMU
- Bochs

Diese Programme emulieren einen kompletten PC und bieten so die Möglichkeit, dass die Betriebssysteme Microsoft Windows und Linux parallel auf einen Computer laufen. Der Vorteil dieser Lösung ist, dass das bestehende Betriebssystem Windows nicht verändert wird. Lediglich die Software für die virtuelle Maschine muss installiert werden. Der Kunde bekommt ein fertiges Linux Image und muss sich nicht um die Konfiguration von Linux kümmern. Auch wird hierbei keine eigene Festplatte oder eine eigene Partition benötigt, die Festplatte wird im Dateisystem des Hostbetriebssystems emuliert.

Auch die Hardware-Kompatibilität spielt dabei eine untergeordnete Rolle. Die Hardware sieht für das Gastbetriebssystem immer gleich aus, egal auf welcher physikalischen Hardware diese läuft. Allein die VM muss die Hardware des Host (Windows) PCs unterstützen. Die somit vereinfachte Installation und Konfiguration von Linux für den Kunden greift auch bei späteren Updates der Linux Umgebung, da nur ein neues Image aufgespielt werden muss.

Für den ersten Prototyp fiel die Entscheidung nach längeren Recherchen auf den VMware Player.

### 3.2 Wann kann der Rechner für parallele Rechenjobs verwendet werden?

Nutzt *Cluster4All* einen Computer für Berechnungen von Simulationen, ist dieser für andere Aufgaben praktisch nicht nutzbar. Solche Simulationen benötigen typischerweise sehr viel Arbeitsspeicher und erzeugen eine hohe CPU-Last. *Cluster4All* darf deshalb nur Rechenprozesse auf diesem Rechner starten, wenn dieser sonst unbenutzt ist. *Cluster4All* reserviert dabei fast den gesamten Arbeitsspeicher des Computers, und erzwingt vor dem Start, dass andere Programme des Gastbetriebssystems auf die Festplatte ausgelagert werden. Die virtuelle Maschine verhindert aber aus Performancegründen, dass Sie selbst ausgelagert wird. Deshalb muss eine sichere Methode entwickelt werden, den PC als unbenutzt erkennen zu können.

In der heutigen Zeit kann nicht mehr davon ausgegangen werden, dass die Arbeitszeit zu festen Zeiten beginnt und endet. In vielen Firmen ist heutzutage Gleitzeit üblich, bei der jeder Mitarbeiter selbst bestimmen kann, wann, und wie lange er heute arbeitet. Aber selbst ohne Gleitzeit sind viele Szenarien denkbar, welche die die Arbeitszeit beeinflussen. Überstunden, Krankheit, Verspätungen, usw. lassen das Zeitfenster, in der ein Computer sicher nicht in Benutzung ist, sehr klein werden. Ein starrer Zeitraum ist daher ungeeignet. Das System sollte nach Möglichkeit selbstständig entscheiden, ob ein Computer für Berechnungen benutzt werden darf.

Es gibt nun verschiedene Möglichkeiten, einen Computer als unbenutzt erkennen zu können. Die einfachste Methode ist es, die CPU-Auslastung zu überwachen. Wenn die CPU-Auslastung einen definierten Grenzwert übersteigt, kann man davon ausgehen, dass der PC in Benutzung ist.

Leider bedeutet dies im Umkehrschluss nicht, dass der PC frei ist, falls die (mittlere) Auslastung unter den Schwellwert gefallen ist. Moderne PCs sind durch die typischen Aufgaben wie das Lesen von Emails oder das Bearbeiten von Texten nicht ausgelastet und befinden sich die meiste Zeit im Leerlauf. *Cluster4All* kann diese leeren Rechenzyklen aber aus oben genannten Gründen nicht nutzen, da es andere Programme zu stark beeinträchtigt. Eine Überwachung der CPU-Last reicht deshalb nicht aus.

Ein alternativer Weg besteht darin, zu überprüfen, ob ein Benutzer im System angemeldet ist. Sollte dies nicht der Fall sein, kann man mit Sicherheit davon ausgehen, dass *Cluster4All* nun seine Arbeit aufnehmen darf. Das setzt aber voraus, dass jeder Mitarbeiter sich bei Ende der Arbeit abmeldet. Sollte dies vergessen werden steht der Computer für *Cluster4All* damit nicht zur Verfügung.

Die sicherste Methode ist daher, die Maus und die Tastatur auf Aktivität zu überwachen. Wenn nach einer definierten Zeitspanne keine Eingaben mehr stattgefunden haben, wird der PC als unbenutzt angesehen. Natürlich wertet diese Überwachung einzig und alleine aus, ob ein neues Ereignis stattgefunden hat. Es wird nicht überprüft oder gespeichert, welche Taste gedrückt worden ist. Da dies trotzdem einen schweren Eingriff darstellt, kann man bei *Cluster4All* wählen, ob die Tastatur/Maus Überwachung genutzt wird, oder ob nur die Anmeldung eines Benutzers überprüft wird.

### 3.3 Clusterbetrieb der Rechner

Für den Start und das Management von Rechenprozessen und Jobs auf dem virtuellen Cluster wird ParaStation, eine bewährte und robuste Middleware für Linux Rechencluster verwendet. Wie andere Lösungen geht auch ParaStation von einer statischen Clusterkonfiguration aus, erwartet also, dass alle bekannten Knoten immer zur Verfügung stehen und nur aufgrund von Hardwareausfällen oder Administratoreingriffen nicht verwendbar sind. Wie wird aber nun ein Rechner diesem virtuellen Cluster hinzugefügt und auch wieder entnommen?

In der Konfiguration des zentralen *Cluster4All* Servers wird eine Zeitspanne als reguläre Arbeitszeit definiert. Innerhalb dieser Zeitspanne wird generell nicht versucht, Jobs auf dem virtuellen Cluster zu starten. Am Ende der konfigurierten Arbeitszeit versucht der *Cluster4All* Server alle ihm bekannten Rechner in das Cluster zu integrieren. Dazu schickt er Befehle an die Klienten, um die lokale virtuelle Maschine zu starten. Sollte zu diesem Zeitpunkt noch ein Benutzer aktiv sein, erkennt dies der Client und verzögert den Start der VM. Über den Server kann der Status jedes einzelnen Klienten jederzeit abrufen werden.

Parastation erkennt nun automatisch, wenn ausreichend viele Knoten verfügbar sind, um geplante Rechenprozesse zu starten. Parastation sieht die virtuellen Maschinen als „normale“ Rechenknoten an. Ein neuer Job kann wie bei einem gewöhnlichen Cluster über die Kommandozeile gestartet werden.

Die Programme für die Berechnungen werden dabei jeder virtuellen Maschine mit Hilfe eines Netzlaufwerks über SMB zur Verfügung gestellt. Das bedeutet, man muss das Programm für die Berechnungen nur einmal auf dem *Cluster4All* Server kopieren. Es ist denkbar, dass in einer späteren Version des *Cluster4All* Klienten eine Möglichkeit geschaffen wird, um neue Rechenjobs bequem aus Windows heraus zu starten.

Das Konzept der Rechenknoten muss für *Cluster4All* erweitert werden. Wenn in einem normalen Cluster ein Knoten plötzlich nicht mehr erreichbar ist, bedeutet dies, dass ein Problem aufgetreten ist. Bei *Cluster4All* kann es durchaus vorkommen, dass ein Knoten nicht mehr antwortet und trotzdem alles in Ordnung ist. Wenn ein Mitarbeiter seinen PC unvorhergesehen benutzt, dann müssen laufende Jobs unterbrochen bzw. beendet werden. Der *Cluster4All* Client erkennt Benutzeraktivität und sendet in diesem Fall eine Nachricht an den Server. Dieser veranlasst Parastation alle Jobs, welche in dieser virtuellen Maschine laufen, anzuhalten. Wenn sicher festgestellt wurde, dass der Benutzer seinen Computer in Anspruch nehmen will, muss die virtuelle Maschine komplett beendet werden.

Auch wenn die VM nicht mehr aktiv rechnet, so gibt sie reservierten Arbeitsspeicher nicht mehr frei, und verhindert zusätzlich, dass sie selbst auf die Festplatte ausgelagert wird. Nur durch ein Beenden der VM hat der Benutzer wieder den gesamten Arbeitsspeicher für seine Programme verfügbar. Die

VM besitzt jedoch einen Mechanismus, mit dem man einen Snapshot des gerade laufenden Systems erzeugen kann, der auf der Festplatte gespeichert wird. Dieser ist mit dem Ruhezustand von Windows vergleichbar. Wenn das System wieder „aufgeweckt“ wird, läuft das System und alle Programme in dem gleichen Zustand wie vorher weiter.

Das suspendieren des gesamten virtuellen Knotens führt nun dazu, dass die in ParaStation integrierte Prozessüberwachung nicht mehr greift, da hierbei der dafür zuständige Dämonprozess ebenfalls nicht mehr reagiert. Dies würde dazu führen, dass die Rechenprozesse auf allen anderen Knoten abgebrochen würden, da von einem fatalen Hardwareproblem ausgegangen wird. Als Lösung ist eine Erweiterung von ParaStation vorgesehen, die diese Situation entsprechend erkennt und reagiert.

#### 4 Erste Erfahrungen, Performance, Benchmarks

Für die Tests und erste Performancemessungen wurden einige typische Arbeitsplatz-PCs verwendet, wie sie auch in Firmen anzutreffen sind. Die Testsysteme hatten folgende Ausstattung:

- Prozessor: AMD Athlon 3200+ (2 GHz)
- Hauptspeicher: 960 MB
- Betriebssystem: Windows XP Professional SP 2

Die PCs waren über einen kostengünstigen Gigabit Ethernet Switch vernetzt, in den Rechnern selbst wurden auch Gigabit Ethernet Karten installiert. In einer virtuellen Maschine (VMware Player) wurde als Gastbetriebssystem Fedora Core5 in einer sehr abgespeckten Version verwendet.

Für die ersten Low-Level Benchmarks wurde der Pallas MPI Benchmark verwendet, welcher eine gute Übersicht über erreichbare Latenzen und Netzwerkdurchsatz liefert. Gemessen wurde mit Standard TCP und dem optimierten Netzwerkprotokoll von ParaStation:

Benchmark PingPong ( Anzahl Prozesse = 2, TCP)

Bytes	repetitions	t[usec]	Mbytes/sec
0	1000	51.66	0.00
1	1000	51.62	0.02
2	1000	51.46	0.04
4	1000	51.57	0.07
8	1000	51.54	0.15
16	1000	51.81	0.29
32	1000	51.58	0.59
64	1000	51.96	1.17
128	1000	58.46	2.09
256	1000	63.07	3.87
512	1000	72.40	6.74
1024	1000	91.11	10.72
2048	1000	128.75	15.17
4096	1000	171.00	22.84
8192	1000	243.32	32.11
16384	1000	439.18	35.58
32768	1000	727.43	42.96
65536	640	1567.42	39.87
131072	320	2727.49	45.83
262144	160	5084.59	49.17
524288	80	9720.70	51.44
1048576	40	18690.95	53.50
2097152	20	36448.90	54.87
4194304	10	72278.64	55.34

Benchmark PingPong (Anzahl Prozesse = 2, Optimierungen mit ParaStation Netzwerkprotokoll)

Bytes	Repetition	T[usec]	Mbytes/sec
0	1000	41.28	0.00
1	1000	41.00	0.02
2	1000	41.42	0.05
4	1000	41.63	0.09
8	1000	41.33	0.18
16	1000	41.57	0.37
32	1000	43.66	0.70
64	1000	43.67	1.40
128	1000	47.92	2.55
256	1000	51.81	4.71
512	1000	61.60	7.93
1024	1000	87.71	11.13
2048	1000	104.86	18.63
4096	1000	161.87	24.13
8192	1000	202.29	38.62
16384	1000	420.55	37.15
32768	1000	676.45	46.20
65536	640	1095.36	57.06
131072	320	1916.28	65.23
262144	160	3554.51	70.33
524288	80	6790.96	73.63
1048576	40	13247.04	75.49
2097152	20	29243.20	68.39
4194304	10	79467.75	50.33

Wie sich zeigt, sind die auf dedizierten Rechenclustern mit ParaStation erreichbaren Latenzen von ca. 20µsec in so einer Umgebung nicht realisierbar. Auch die erzielbare Bandbreite liegt deutlich unter den entsprechenden Zahlen eines Rechenclusters. Die Emulation einer Netzwerkkarte durch die virtuelle Maschine ist hierbei der begrenzende Faktor.

High-Level Benchmarks mit LS-Dyna werden derzeit durchgeführt, Ergebnisse liegen momentan noch nicht vor.

## 5 Voraussetzungen

Wie sich gezeigt hat, sind für die effiziente Nutzung eines Konzeptes wie *Cluster4All* einige Voraussetzungen bezüglich der verfügbaren Infrastruktur notwendig.

Entscheidende Faktoren sind dabei einigermaßen aktuelle PC Hardware, die mit ausreichendem Hauptspeicher ausgestattet sein muss. Als Hostbetriebssystem ist eine aktuelle Windows-Variante notwendig. Für die Installation der für *Cluster4All* notwendigen Software ist auf den jeweiligen PC ca. 1GB Festplattenplatz vorzusehen.

Als Interconnect ist unbedingt Gigabit Ethernet anzustreben, eine Kopplung über Fast Ethernet (100MB) ist wegen der dadurch bedingten deutlich höheren Latenzzeiten, vor allem aber wegen der massiv geringeren Bandbreite, nicht ausreichend.

Für die Verwaltung des virtuellen Clusters ist ein eigener Linux-basierter Server vorgesehen, welcher gleichzeitig entsprechenden Plattenplatz und Fileshares für *Cluster4All*-spezifische Daten bereitstellen kann.

Er übernimmt dabei folgende Dienste:

- *Cluster4All* Server (steuert Start und Beendigung von virtuellen Maschinen, Suspendierung von Jobs)
- Parastation (Prozess- und Jobverwaltung, rechnet im *Cluster4All* Cluster mit)
- Samba Server (Bereitstellung von Images, Konfigurationsdateien)
- DHCP Server für die virtuellen Maschinen
- Genereller Fileserver

Eine weitere Voraussetzung ist die „Kooperation“ der Benutzer. Die Arbeitsplatzrechner dürfen natürlich nicht ausgeschaltet werden, sondern müssen permanent laufen. Der Benutzer sollte sich am besten vom System abmelden. Jedwede aktive Bildschirmschoner verbrauchen natürlich CPU-Zeit und be- oder verhindern sogar, dass der jeweilige Knoten innerhalb von *Cluster4All* verwendet werden kann.

## 6 Ausblick

Derzeit ist *Cluster4All* als Konzeptstudie und als Prototyp verfügbar. Bei entsprechendem Interesse wird es weiterentwickelt werden und als supportete Lösung für kostengünstige, virtuelle Clusterlösungen dienen. Dies dürfte besonders als Einsteigerlösung für Ingenieurbüros, CAD- und CAE-Anwender attraktiv sein.

## 7 Zusammenfassung

Mit dem Konzept *Cluster4All* steht eine interessante Alternative oder Ergänzung zur Installation eines dedizierten Rechenclusters zur Verfügung. Vorhandene Infrastruktur kann zusätzlich genutzt werden und bietet dadurch eine sehr kostengünstige Plattform für die Abarbeitung von parallelen Rechenjobs. Eine Integration in bestehende oder zukünftig geplante Rechencluster ist jederzeit möglich.