# Scheduling in Rechenzentren

**Seminar „Scheduling in verschieden Produktionsbereichen“**
**ZHAW - Zürcher Hochschule für Angewandte Wissenschaften**
**SoE - School of Engineering**

**Miro Ljubicic**
ljubimir@students.zhaw.ch

12. Juni 2013

# Inhaltsverzeichnis

# 1 Abstract

Diese Arbeit befasst sich mit Scheduling in Rechenzentren und ist an der Zürcher Hochschule für angewandte Wissenschaften (ZHAW) im Rahmen des Seminars „Scheduling in verschiedenen Produktionsbereichen“ entstanden. Ziel ist eine Vertiefung des Seminarteilnehmers in das gewählte Thema inkl. Ausarbeitung eines technischen Berichts (dieses Dokument), eines einseitigen Handouts (separates Dokument) sowie einer kurzen Präsentation im Klassenplenum.

Die Arbeit ist wie folgt aufgebaut: In Abschnitt 2 wird die Aufgabenstellung kurz erläutert. In Abschnitt 3 werden die notwendigen Grundlagen zu Rechenzentren sowie die verwendete Notation für Schedulingprobleme erläutert. Abschnitt 4 widmet sich der Betrachtung konkreter (realitätsnaher) Schedulingprobleme und stellt Lösungsvarianten vor. Abschliessend folgt in Abschnitt 5 ein Fazit des Autors zur weiteren Entwicklung des Schedulings in Rechenzentren.

# 2 Aufgabenstellung

## 2.1 Ausgangslage

Scheduling befasst sich mit der Zuteilung (resp. Beanspruchung) von Ressourcen (machines) um Aufgaben (jobs) in Zeit (time) auszuführen. Die Lösungsfindung orientiert sich zudem an Zielkriterien (objectives) sowie optionalen Nebenbedingungen (costraints). Beim Einsatz in Rechenzentren sind solche Nebenbedingungen häufig Grenzen von finanziellen Mitteln, physischer Kapazität (Platz, Strom, Kühlung, Kommunikation, etc.) sowie der maximalen Ausführungszeit einer Informationsverabeitung (z.B. eines Batch-Jobs).

## 2.2 Ziel der Arbeit

Die Darstellung des Zuordnungsproblems von Rechenjobs zu physischen oder virtuellen Maschinen in Rechenzentren ist Gegenstand dieser Arbeit. Dabei soll sowohl auf das eigentliche Schedulingproblem, der Zuordnung der Rechenjobs zu Maschinen im Allgemeinen, als auch von virtuellen zu physischen Maschinen im Speziellen eingegangen werden. Neben den reinen Jobzuordnungen sollen auch die Effekte auf den Betrieb des Rechenzentrums erfasst werden. Dabei sind speziell Energiebedarf und Kühlungsaspekte zu berücksichtigen. Ein wesentlicher Teil dieses Seminarthemas besteht auch darin, dass man dieses Schedulingproblem dynamisch anpassen kann und damit keine vollständige Lösung von Beginn an benötigt. Auch dieser Aspekt soll entsprechend hervorgehoben werden. Wie in allen anderen Arbeiten sollen die vorhandenen Nebenbedingungen und die möglichen Optimierungsziele hervorgehoben werden, kombiniert mit existierenden Lösungsansätzen.

## 2.3 Aufgabenstellung

Vertiefung ins Thema „Scheduling in Rechenzentren“. Ausarbeitung repräsentativer Lösungsmöglichkeiten für vielfältige Probleme sowie deren Übertragbarkeit für praktische Anwendungen.

## 2.4 Sprachliche und typografische Konventionen

**Sprache** Diese Seminararbeit ist gemäss reglementarischer Vorgaben in deutscher Sprache verfasst. Da die zu Grunde liegenden Publikationen im Bereich „Scheduling“ jedoch zum überwiegenden Teil ausschliesslich auf englisch zur Verfügung stehen, bleiben etablierte englische Begriffe mit hoher Akzeptanz und Verbreitung unübersetzt.

**Typografie** Nachfolgend die wichtigsten typografischen Konventionen, welche in diesem Dokument verwendet werden:

- *„Kursive Schrift in Anführungszeichen“*
  Zitat (Quellenangabe am Ende des Zitats)
- *Kursive Schrift ohne Anführungszeichen*
  Wichtiger Fachbegriff
- $f(x) = x^2$
  Mathematische Formel

# 3 Grundlagen

## 3.1 Definition

*„Scheduling beschreibt die Planung der Ressourcenbelegung durch Aufgaben über der Zeit. Wesentliche Elemente sind also:*

- *Ressourcen (machines): z.B. Maschinen, Menschen*
- *Aufgaben (jobs): z.B. Fertigungsaufträge*
- *Zeit (es wird in der Zukunft geplant und eventuell dynamisch angepasst)*

*Beim Erstellen von Lösungen für das jeweilige Schedulingproblem gibt es:*

- *Zielkriterien (objectives): z.B. schnellstmögliche Erledigung eines Auftrags*
- *Nebenbedingungen (constraints): z.B. Kostenobergrenzen, Deadlines*

“ [2]

## 3.2 Notation

Nachfolgend die in diesem Dokument verwendete fachliche Notation für Schedulingprobleme:

$$\alpha|\beta|\gamma$$

- $\alpha$ repräsentiert die *Maschinenumgebung* (machine)
  (spezifiziert die funktionalen und organisatorischen Möglichkeiten, welche auf Maschinenseite gegeben sind)
- $\beta$ sind die *Nebenbedingungen (constraints)*
  (diese sind durch die Verarbeitung des Jobs gegeben)
- $\gamma$ sind die *Optimierungsziele*
  (Typischerweise umfassen diese eine Minimierung von Laufzeit, Verspätungen, etc.)

[2]

### 3.2.1 Jobs

- $r_j$
  Zeitpunkt der Bekanntmachung (release time) eines Jobs $j$
- $d_j$
  Der angestrebte Beendigungszeitpunkt (due date) eines Jobs $j$ (nachträgliche Fertigstellung zieht Strafen nach sich)
- $\overline{d_j}$
  Der letztmögliche Beendigungszeitpunkt (deadline) eines Jobs $j$. Danach ist das Resultat wertlos.

### 3.2.2 Machines

- $1$
  Einmaschinenmodell (eine einzelne Ressource)
- $P_m$
  $m$ parallele, identische Machines (z.B. ein Cluster aus m identischen Rechnern)
- $FF_c$
  Flexibler Flowshop mit $c$ Schritten, wobei für jedem Schritt mehrere identische Machines zur Verfügung stehen. Jeder Job absolviert die gleiche Reihenfolge an Verarbeitungen
- $Q_m$
  $m$ parallele Machines, die jedoch unterschiedlich schnell sind (z.B. ein Cluster mit unterschiedlichen schnellen Rechnern)

### 3.2.3 Constraints

- $brkdwn$
  Machines sind nicht permanent verfügbar (z.B. Wartung)
- $M_j$
  Der Job $j$ kann nur auf einer Untermenge $M_j$ aller Machines ausgeführt werden
- $block$
  Ein Job nur dann von einer Machine auf die andere verlagert werden, wenn die entsprechenden Puffer vorhanden sind. Bis zur Verlagerung ist die vorherige Machine blockiert.

### 3.2.4 Optimierungsziele

- Minimierung des Fertigstellungszeitpunkts $C$ (completion time):

$$\min \sum w_j * C_j$$

- Minimierung der Verspätung $L$ (lateness):

$$L_j = C_j - d_j$$

$$\min \sum w_j * L_j$$

Weiterführende Informationen zur Notation in [2]

## 3.3 Rechenzentren

### 3.3.1 Definition

*„Ein Rechenzentrum ist ein Bereich, ein Raum, eine Einrichtung oder ein Standort einer zentralen Datenverarbeitung. Das Rechenzentrum ist ein Dienstleistungsunternehmen oder eine Abteilung eines Unternehmens in dem die Massendatenverarbeitung in Programmläufen und der Betrieb von Mainframes und anderer zentraler Systemkomponenten für netzorientierte Datenverarbeitungssysteme erfolgt. Rechenzentren stellen ihre Rechenleistung der eigenen oder fremden Firmen gegen Entgelt zur Verfügung.“* [6]

### 3.3.2 Betrieb von Rechenzentren

Ein Rechenzentrum ist in der Regel in mehrere Bereiche (z.B. separate Räume) unterteilt. Dies dient mehreren Zielen:

- Zutrittsautorisierung
- Unterschiede bei der benötigten Aussstattung (z.B. Netzwerkanbindung)
- topologische Nähe zu weiteren Systemen, mit welchen interagiert wird
  (Systeme, welche zeitkritische Services anbieten, sollen eine möglichst niedrige Latenz und hohe Bandbreite zu unmittelbaren Kommunikatonspartnern haben)
- Brandschutz
  (Bei Ausbruch eines Feuers bleibt dieses zunächst auf einen kleinen Raum beschränkt)
- Kühlung
  (Mehrere kleinere Räume, welche Systeme unterschiedlicher Klassen[1] beherbergen, sind einfacher und gezielter zu kühlen als ein einzelner grosser Raum)

Jeder Raum besitzt zur Regelung der Raumtemperatur sowie der Menge und Qualität zugeführter Frischluft eine oder mehrere Kühleinheiten (Computing Room Air Conditioner - CRAC) [11]. Um einen möglichst energieffizienten, kostengünstigen und stabilen Betrieb zu gewährleisten, müssen folgende Aspekte berücksichtigt werden:

- Einhaltung der maximal zulässigen Temperatur, für welche ein einzelnes System im worst-case ohne Abstriche bei Leistungsfähigkeit und Stabilität ausgelegt ist ($T_{current} < T_{emergency}$)
- maximale Kühlleistung des CRAC

[1] Der Begriff *Klasse* bezieht sich hierbei primär auf Unterscheidungsmerkmale im Bereich Rechenkapazität und Abwärme/Kühlungsbedarf

- maximale Luftmenge pro Zeiteinheit, welche das CRAC umzuwälzen (resp. frisch zuzuführen) vermag
- zeitliche Dauer bis die Luft vom CRAC beim zu kühlenden System ankommt
- geometrische Ausgestaltung der Luftströmung

### 3.3.3 Ressourcenverbrauch

Durch die räumliche Konzentration vieler Geräte (Server, Netzwerkkomponenten, etc.) auf engstem Platz entsteht zwangsläufig eine Problematik bei der Versorgung mit elektrischer Energie sowie dem Abtransport der entstehenden Wärme.

**Energiebedarf** Server in der Verbrauchklasse <1000 Watt hatten im vierten Quartal 2012 gemäss SPEC[2] im Durchschnitt einen Verbrauch (bei 100% Auslastung) von 276 Watt. Eine exemplarische Hochrechnung auf grösseres Rechenzentrum von 1000 Servern ergibt eine Leistungsaufnahme von 276 KW - hierbei sind zusätzliche Verbraucher wie Netzwerkkomponenten noch nicht einmal berücksichtigt. Zum Vergleich: Dies entspricht fast 1% der Nettoleistung eines kleineren Schweizer Atomkraftwerks wie MÜHLEBERG[3]. Die Bereitstellung der elektrischen Leistung ist nur eine Herausforderung; diese über Leitungen ins Gebäude zu führen und zu verteilen allerdings auch. Ausserdem muss ein Teil dieser Leistung bei einem Stromausfall über die USV-Architektur (unterbrechungsfreie Stromversorgung) für kritische Systeme weiterhin aufrecht erhalten werden.

**Kühlung** Aus dem hohen Energiebedarf und der damit verbundenen Abwärme ergeben sich Herausforderungen, diese Abwärme konstant abzuführen resp. durch gekühlte Luft zu ersetzen. Hierbei entstehen durch vergleichsweise niedrige Wirkungsgrade zusätzlicher Stromverbrauch und erneut Abwärme.

[2] `http://www.spec.org/power_ssj2008/results/res2012q4/`
[3] `http://www.iaea.org/pris/CountryStatistics/CountryDetails.aspx?current=CH`

# 4 Schedulingprobleme

## 4.1 Jobs zu Maschinen - Definition der Umgebung

Die Menge an möglichen Schedulingproblemen ist - je nach Umfang der verwendeten Notation - sehr gross. Im Rahmen dieser Arbeit wird aus Gründen der Vereinfachung folgende Umgebung definiert:

### 4.1.1 Jobs

- Gewöhnlicher Job
  Ein gewöhnlicher Job, welcher keine besonderen Anforderungen an die Verarbeitung stellt:

$$r_j$$

- Echtzeitjob
  Ein Job, welcher nahezu in Echtzeit ausgeführt werden soll, eine spätere Ausführung zieht Strafen (penalties) nach sich (z.B: eine Börsentransaktion):

$$r_j \sim d_j$$

- Job mit Deadline
  Ein Job, bis zur Deadline ausgeführt werden muss, da das Ergebnis ansonsten wertlos wird (meteorologische Berechnungen, Prognose von Verkehrsflüssen basierend auf aktuellen Daten, etc.):

$$\overline{d_j}$$

### 4.1.2 Maschinen-Umgebung (machine)

Rechenzentren weisen je nach Einsatzgebiet verschiedene Ausprägungen auf. In dieser Arbeit werden folgende Machines betrachtet:

- Einmaschinenmodell
  Dieser Typ ist bildet eine einzelne Maschine (i.d.R. mit einer CPU) und dient als Referenz für die nachfolgenden Typen:

$$1$$

- Homogene Rechenzentren mit Maschinen gleichen Typs (parallele Verarbeitung möglich):

$$P_m$$

- Flexibler Flowshop mit $c$ Schritten, wobei für jedem Schritt mehrere identische Maschinen zur Verfügung stehen. Jeder Job absolviert die gleiche Reihenfolge an Maschinenverarbeitungen:

$$FF_c$$

- Grids
  Global vernetzte, einzelne Maschinen verschiedener Typen und Leistungsklassen:

$$Q_m$$

### 4.1.3 Nebenbedingungen (constraints)

Für das Scheduling in einem Rechenzentrum sind vor allem folgende Nebenbedingungen relevant:

- Breakdown
  Maschinen sind nicht permanent verfügbar (z.B. Wartung):

$$brkdwn$$

- Maschinenrestiktionen
  Der Job $j$ kann nur auf einer Untermenge $M_j$ aller Machines ausgeführt werden. In einem Rechenzentrum wird dies durch verschiedene Maschinenkonfigurationen, Plattformen, Betriebssysteme, oder vorhandene Software manifestiert.

$$M_j$$

- Blocking
  Ein Job nur dann von einer Maschinen auf die andere verlagert werden, wenn die entsprechenden Puffer vorhanden sind. Bis zur Verlagerung ist die vorherige Maschine blockiert. Im heutigen Rechenzentren, welche eine dezentrale interne Struktur aufweisen (dedizierte Systeme für Applikationen, Datenbanken, Reporting, etc.), ist dies der Regelfall

$$block$$

### 4.1.4 Optimierungsziele

- Minimierung der Completion Time:

$$\min \sum w_j * C_j$$

  Diese Vorgabe gilt im Prinzip für jedes Schedulingproblem.

- Minimierung der Lateness:

$$\min \sum w_j * L_j$$

  Für Jobs mit Due Date gilt es, diesen Wert möglichst gegen Null zu optimieren, da ansonsten Strafen (penalties) anfallen. Für Jobs mit Deadline gilt hingegen immer:

$$\sum w_j * L_j = 0$$

### 4.1.5 Realitätsnahe Schedulingprobleme

Aus den obigen Jobs, Machines, Nebenbedingungen und Optimierungszielen werden folgende - möglichst realitätsnahe - konkreten Schedulingprobleme kombiniert.

- Börsentransaktion:

$$P_m|r_j \sim d_j|\sum w_j * L_j$$

  Diese Jobs laufen auf einem homogenen parallelen Menge an Maschinen, haben eine hohe Anforderung im Bezug auf die Verzögerung, welche möglichst niedrig sein soll, um Strafen in Form von Spekulationsverlusten zu minimieren.

- Rechnungsstellung:

$$FF_c|r_j, block, brkdwn|\sum w_j * C_j$$

  Diese Arbeiten sind in der Regel hochgradig parallelisiert, da die Rechnungsstellung mit hohen Volumen gleichartiger Schritte arbeitet, welche nacheinander ablaufen (z.B. Applikation -> Datenbank -> Applikation -> Rendering). Gleichzeitig sind Nebenbedingungen relevant, welche Durchsatz und allgemeine Verfügbarkeit der Maschinen beeinträchtigen. Optimierungsziel ist eine Reduktion der Durchlaufzeit, wobei für verspätete Rechnungsstellung im Normalfall keine konkreten Strafen anfallen.

- Meteorologische Berechnung für die nächsten drei Tage: Hierbei existieren mehrere Möglichkeiten, die Berechnung durchzuführen:
  - Berechnung in einer dedizierten Simulationsumgebung (z.B: „NEC Earth Simulator“[1])

$$P_m, |\overline{d_j}, brkdwn|\sum w_j * Lj = 0$$

  - Berechnung in einem dezentralen Grid (z.B. weltweiter Verbund von unabhängigen Systemen in verschiedenen Standorten)

$$Q_m, |\overline{d_j}, brkdwn|\sum w_j * Lj = 0$$

  Um eine meteorologische Vorhersage aufgrund komplexer Eingabedaten nutzen zu können, darf deren Berechnung logischerweise nicht länger dauern als der Zeitrahmen, für den sie gilt. Solche Berechnungen werden in der Regel auf dedizierten Umgebungen ausgeführt. Es besteht prinzipiell aber auch die Möglichkeit, diese an ein global verteiltes Grid auszulagern.

### 4.1.6 Lösungsansätze

#### 4.1.6.1 Dynamische Anpassung

Die in Abschnitt 4.1.5 aufgeführten Schedulingprobleme sind relativ statisch definiert und erlauben a priori nur beschränktes Wissen über Komplexität und/oder erwartete Laufzeit. Um eine dynamische Anpassung der Schedulingstrategien zu ermöglichen, können verschiedene Verfahren eingesetzt werden. Nachfolgend beschränken wir uns auf:

[1] http://www.jamstec.go.jp/es/en/index.html

### Klassische Verfahren

- Market-oriented scheduling
  Dieses Modell beruht auf dem Market-Prinzip: Alle Teilnehmer modellieren ihr zu lösendes Problem als Nutzenfunktion und reichen sie im „Markt“ ein. Daraus wird ein Gleichgewichtspreis (equilibrium price) ermittelt und für jedes angeforderte Gut ein separate Auktion gestartet. Anschliessend erhalten alle Teilnehmer das Resultat und können bei Bedarf ihre Nutzenfunktion anpassen, um eine neue Auktion auszulösen. Dies wird wiederholt, bis sich der Gleichgewichtspreis stabilisiert hat. Das System *WALRAS* [4] verwendet dieses Modell.

- NWIRE
  Franke et al. haben in [4] ein weiteres Modell eingeführt, welches auf einer verteilten Suche im *NWIRE-Netzwerk* nach freien Ressourcen sucht. Hierbei werden die Ressourcen auf einzelne Domains (jede mit eigenen Teilressourcen und eigenem Scheduling) verteilt. Jede Domain wird durch einen *Meta-Manager* verwaltet, welcher Ressourcenanfragen anderer Meta-Manager empfangen und verarbeiten kann, aber auch eigene Anfragen an andere Meta-Manager versenden kann.

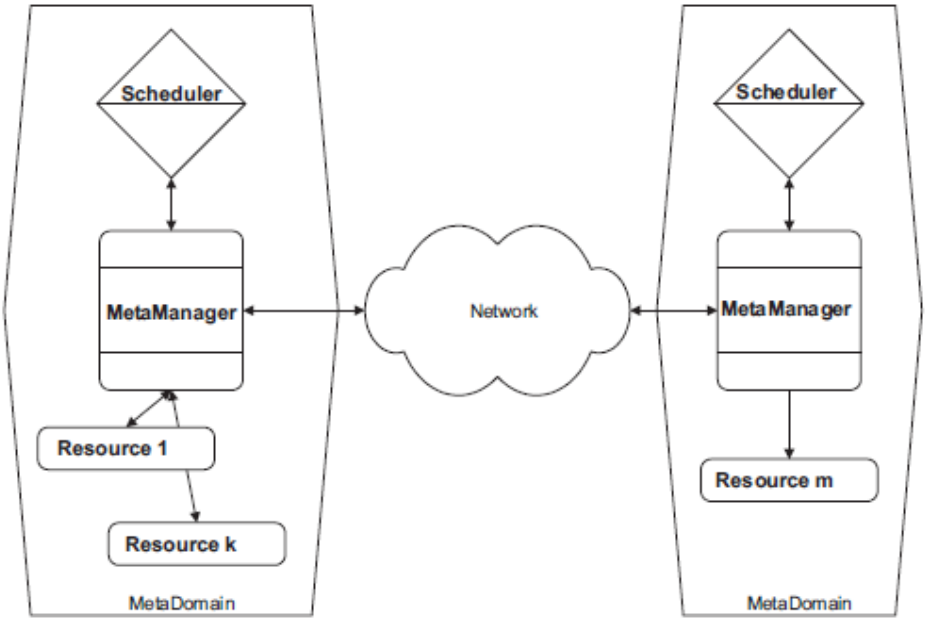


Abbildung 4.1: Aufbau von NWIRE, Quelle: [4]

### Verfahren unter Einbezug von Energie-/Kühlungsaspekten

- Thermal-aware scheduling
  Dieses Modell legt den Fokus primär auf die Optimierung des Energieverbrauchs heutiger Rechenzentren. Berral et al. haben in [7] ein Modell vorgestellt, welches basierend auf maschinellem Lernen eine Jobzuteilung aufgrund energetischer und thermischer Gesichtspunkte ausführen kann. Wang et al. haben in [9] ebenfalls mit *TASA (Thermal-aware Scheduling Algorithm)* einen Algorithmus vorgestellt, welcher das Scheduling auf Basis des resultierenden Temperaturanstiegs

steuert, woraus eine signifikante Energieeinsparung resultieren kann (s. Abb. 4.2). Hierzu bedient er sich am Anfang gemessener Daten über Auslastung und Dauer einzelner Jobs (s. Abb. 4.3).

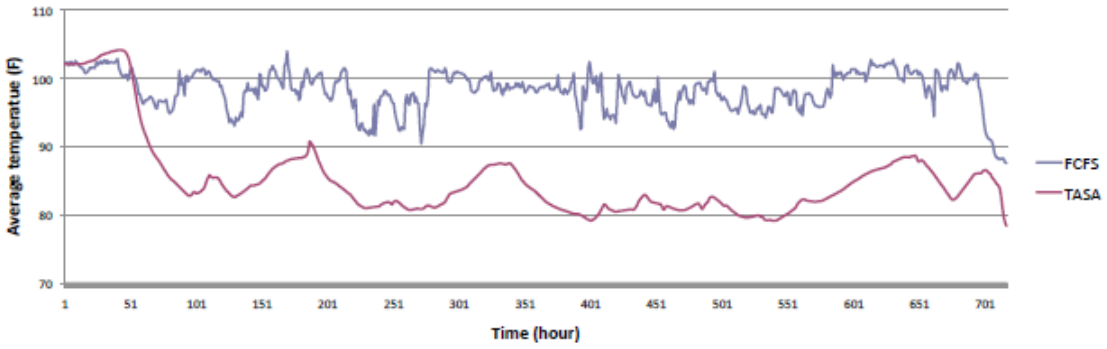


Abbildung 4.2: Reduktion des Energieverbrauchs durch Einsatz von TASA gegenüber FCFS, Quelle: [9]

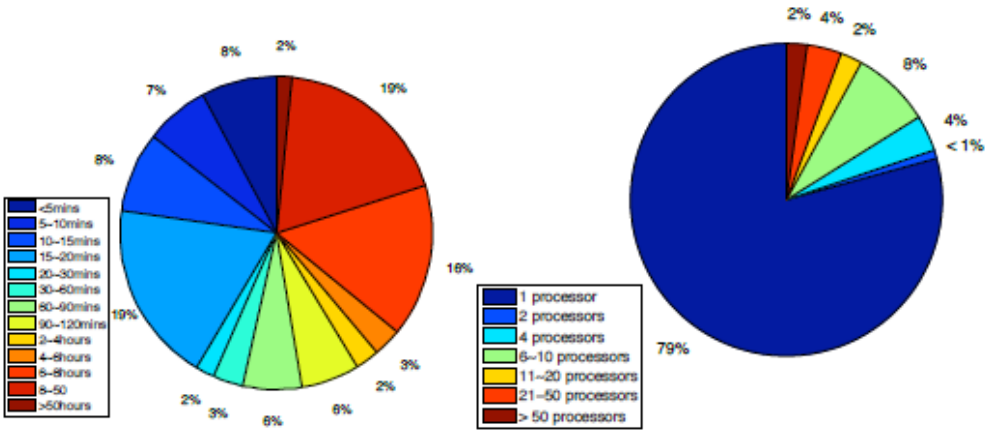


Abbildung 4.3: Durchschnittliche Dauer und Ressourcenverbrauch von Jobs (gemessen durch TASA), Quelle: [9]

Die zuvor beschriebenen Verfahren zur dynamischen Anpassungen wirken primär auf Formulierung der Jobs in 4.1.5.

#### 4.1.6.2 Design von Rechenzentren

Neben Scheduling-basierten Lösungen für eine Optimierung des Energieverbrauchs und Kühlungsbedarfs von Rechenzentren, sind durch geschicktes Design von Rechenzentren ebenfalls signifikante Einsparungen möglich:

- Kühltemperatur
  Viele Rechenzentren kühlen die Serverräume aufgrund der heterogenen Durchmischung verschiedenster Serversysteme zu stark ab (Kühlung auf die niedrigste Toleranztemperatur). Es sollte angestrebt werden, Systeme mit ähnlicher Leistungsaufnahme und Temperaturbandbreiten zu gruppieren.

Umgekehrt wird oft versucht, die mittlere Raumtemperatur durch Absenkung der Kühlleistung zu erhöhen, um die für das CRAC verwendete Energie zu minimieren. Allerdings versuchen die meisten Server, dies durch Erhöhung der Umdrehungszahl der eingebauten Lüfter zu kompensieren (die gleiche Wärme soll durch grösseres Luftvolumen pro Zeiteinheit abgeführt werden). Dies kann die angestrebte Verbrauchsreduktion verringern oder gar ins Gegenteil umkehren. Yeo et al. [11] haben durch Simulationen nachgewiesen, dass im Prinzip für jedes Rechenzentrum ein optimales Gleichgewicht zwischen Kühlleistung, thermischen Anforderungen der Systeme sowie dem Stromverbrauch von eingebauten Lüftern gefunden werden kann.

- Entfernung zwischen Kühlanlage und zu kühlenden Systemen
  Je weiter die Kühlanlage von den Servern entfernt, umso höher ist die Verzögerung, bis kühle Umgebungsluft die Komponenten des Servers wirkungsvoll kühlt. Dadurch wird unnötig Kühlleistung bereitgestellt.

- Steuerung des Luftstroms (Route und Geschwindigkeit)
  Durch effektive Steuerung des Luftstroms in Route und Geschwindigkeit kann die Kühlung optimiert werden. Dieser Aspekt wurde durc Yeo et al. [11] ebenfalls analysiert.

- Platzierung der Systeme
  Wie in Abb. 4.4 gezeigt, erwärmen sich in einem typischen Rechenzentrum (bedingt durch aufsteigende Warmluft) die obersten Geräte in einem Serverrack am meisten. Es macht daher Sinn, Systeme mit der geringsten Wärmetoleranz so zu platzieren, dass sie durch einströmende Kühlluft zuerst erreicht werden - im Umkehrschluss sollten Systeme mit hoher thermischer Toleranz so platziert werden, dass sie die Kühlluft zuletzt erreicht.

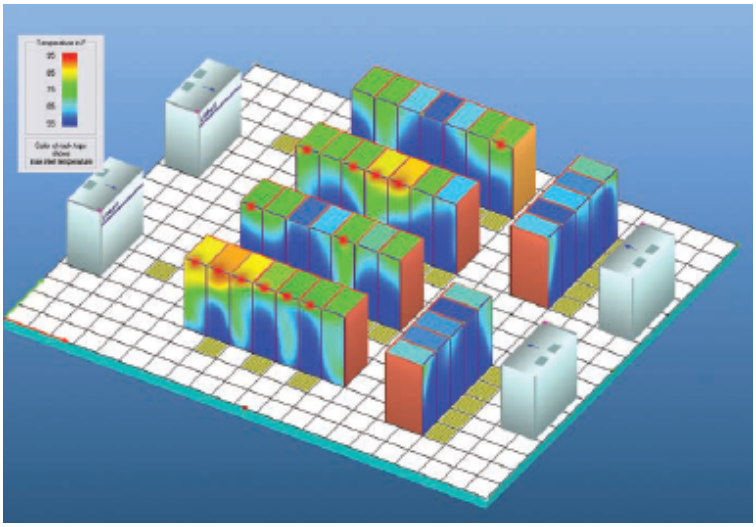

Abbildung 4.4: Typische Wärmeverteilung in einem Serverraum, Quelle: [9]

## 4.2 Virtuelle zu physischen Maschinen (VM-to-PM)

Ähnlich wie Jobs anhand ihrer Charakteristiken und des zu Grunde liegenden Schedulingproblems zu Machines zugewiesen werden, sind Zuweisungen von virtuellen zu physischen Maschinen möglich. Heutige Virtualisierungslösungen ermöglichen eine unterbrechungsfreie (Re-)Allokation von Ressourcen (Speicher, CPU, I/O) in einem dynamischen Umfeld. Die Vorteile sind hierbei

- Minimierung der Kosten (eine dezentrale Serverlandschaft ist gemessen an der Leistung teurer als wenige grosse Virtualisierungshosts),
- einfachere Verwaltung (zentrales Management und Monitoring),
- Maximierung der Verfügbarkeit (Failover, Clustering, Load Balancing),
- Maximierung des Durchsatzes (parallele Verarbeitung), sowie
- ausgeglichenere Auslastung (Ad-hoc-Verarbeitung zu Geschäftszeiten, Batch-Verarbeitung ausserhalb von Geschäftszeiten)

Im Gegensatz zum Job-zu-Maschine-Scheduling aus 4.1 geht es hier jedoch darum, einzelne Virtualisierungshosts über die gesamte Einsatzdauer maximal auszulasten. Dies wird auf verschiedene Arten erreicht:

- Analyse des Auslastungszyklus und Ableitung zukünftiger Entwicklungen (Capacity Management)
- Vorgaben seitens der Kunden/Benutzer bezüglich Kosten und erwarteter Leistung und Verfügbarkeit (Financial Management, Service Level Management, Availabilty Management)
- Abschaltung nicht ausgelasteter VMs und Verteilung ihrer Jobs auf andere VMs

Aus technischer Sicht gibt es diverse Methoden, über die Zeit ein optimales VM-to-PM-Scheduling zu etablieren. Beloglazov et al. haben in [1] einen performanten Algorithmus zur energieeffizienten VM-Selektion sowie VM-Allozierung innerhalb eines Rechenzentrums vorgestellt und in Simulationen bestätigt, dass dadurch der Energieverbrauch signifikant gesenkt werden kann, ohne die Quality-of-Service (QoS) nachteilig zu beeinflussen.

# 5 Fazit

## 5.1 Scheduling

Noch vor 25 Jahren umfasste Scheduling in Rechenzentren primär grosse Rechenanlagen von Regierungen, Universitäten und Firmen, welche weitestgehend isoliert von der Aussenwelt vorab geplante Probleme berechneten. Ein automatisiertes, autonomes Scheduling auf Ebene Rechenzentrum war faktisch nicht vorhanden. Mit dem technologischen Fortschritt (inbesondere im Bereich Netzwerk und Performance von Rechenanlagen) ergeben sich neue Möglichkeiten, Rechenkapazität zu vernetzen und verfügbar zu machen. Gleichzeitig wuchsen auch die Herausforderungen, Rechenkapazität möglichst effizient und effektiv zu nutzen. Scheduling in Rechenzentren ist heute eine vielseitige Disziplin, welche Modelle entwickelt, die ungebremst wachsende, global verteile Rechenleistung in einer konsolidierten Form dem Nutzer zur Verfügung zu stellen. Ein Ende der Entwicklung ist nicht abzusehen.

## 5.2 Leistungsaufnahme und Kühlung

Während sich das klassische Scheduling in Rechenzentren an den Problemen/Jobs und deren möglichst schnellen Abarbeitung orientierte, wächst seit einigen Jahren das Bewusstsein, dass die riesigen IT-Installationen auf der ganzen Welt einen erheblichen negativen Einfluss auf unsere Umwelt haben (Stichwort „Green IT“) Dies ist hauptsächlich auf die gigantische benötigte elektrische Energie für die eingesetzte Hardware zurückzuführen. Gleichzeitig erzeugt diese Hardware enorme Mengen an Abwärme, welche - wiederrum unter grossem Verbrauch elektrischer Energie - an die Umwelt abgeführt werden muss.
Die Zeiten, in denen Rechenleistung einfach mit höherer elektrischer Leistungsaufnahme erkauft werden konnte, sind glücklicherweise vorbei. Heutige Modelle setzen einerseits auf effizientere Hardware und verbessertes Design von Rechenzentren. Gleichzeitig leistet Scheduling in Rechenzentren einen wertvollen Beitrag, die vorhandene Hardware unter Gesichtspunkten von elektrischem Verbrauch und Kühlbedarf bestmöglich auszulasten.

# 6 Glossar

Nachfolgend die wichtigsten Fachbegriffe und Fremdwörter

**Constraint** Menge von Nebenbedingungen, welche bei der Abarbeitung von Jobs durch Maschinen eingehalten werden muss.

**CRAC (Computing Room Air Conditioner)** Kühleinheit, welche die dem Serverraum zugeführte Luft überwacht und den Geräten kontrolliert zuführt. Die wichtigsten Parameter sind hierbei Luftmenge, -temperatur, -feuchtigkeit sowie Staubkonzentration. Üblicherweise als geschlossener Regelkreis betrieben und aus der Ferne überwacht. [10]

**Grid** Vereinigung mehrerer unabhängiger Computersysteme, um die gemeinsamen Ressourcen gleichzeitig zur Berechnung eines - normalerweise wissenschaftlichen oder technischen - Problems - zu nutzen. Dies erfodert in der Regel viele Prozessoren und hohe I/O-Volumen.

**I/O (Input/Output)** Ein-/Ausgabe von Daten auf Computersystemen.

**Job** Arbeitsanweisung resp. Abfolge von Arbeitsanweisungen, welche durch eine Machine bearbeitet werden soll.

**Machine** Entspricht einer Menge an (nicht notwendigerweise physischen) Maschinen, auf welchen *Jobs* ausgeführt werden.

**USV** Unterbrechungsfreie Stromversorgung, um bei einem Stromausfall kritische Systeme eines Rechenzentrums weiterhin zu betreiben.

# 7 Abbildungsverzeichnis

# 8 Literaturverzeichnis

[1] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing, May 2011.

[2] Carsten Franke. Seminareinführung - Schedulingansätze in unterschiedlichen Produktionsbereichen, 2013.

[3] Carsten Franke, Joachim Lepping, Uwe Schwiegelshohn. Genetic Fuzzy Systems applied to Online Job Scheduling.

[4] Carsten Franke, Volker Hamscher, Ramin Yahyapour. Economic Scheduling in Grid Computing. Computer Engineering Institute, University of Dortmund, 44221 Dortmund, Germany.

[5] Erhan Kutanoglu, S. David Wu. Improving scheduling robustness via preprocessingand dynamic adaptation, 2003. Department of Mechanical Engineering, University of Texas at Austin – Department of Industrial and Systems Engineering, Lehigh University.

[6] ITWissen.info. Definition: RZ (Rechenzentrum). `http://www.itwissen.info/definition/lexikon/Rechenzentrum-computer-centre-RZ.html`. aufgerufen am 10. Mai 2013.

[7] Josep Ll. Berral, Íñigo Goiri, Ramón Nou, Ferran Julià, Jordi Guitart, Ricard Gavaldà, Jordi Torres. Towards energy-aware scheduling in data centers using machine learning. Computer Architecture Department, Department of Software (Universitat Politècnica de Catalunya) – Barcelona Supercomputing Center.

[8] Justin Moore, Jeff Chase, Parthasarathy Ranganathan, Ratnesh Sharma. Making Scheduling "Cool-Temperature-Aware Workload Placement in Data Centers. Department of Computer Science, Duke University – Internet Systems and Storage Lab, Hewlett Packard Labs.

[9] Lizhe Wang, Gregor von Laszewski, Jai Dayal, Xi He, Andrew J. Younge, Thomas R. Furlani. Towards Thermal Aware Workload Scheduling in a Data Center. Rochester Institute of Technology – State University of New York at Buffalo.

[10] SearchDataCenter. Definition: Computer Room Air Condition. `http://searchdatacenter.techtarget.com/definition/computer-room-air-conditioning-unit`. aufgerufen am 15.5.2013.

[11] Sungkap Yeo, Hsien-Hsin S. Lee. SimWare: A Holistic Warehouse-Scale Computer Simulator, September 2012. Georgia Insitute of Technology.