



## **SAS in Action** - Scheduling von SAS Jobs mit dem Platform LSF Scheduler beim LVM

Daniel Hakenjos  
DV-Organisation/Data Warehouse

Plattform Netzwerk Treffen  
05.10.2010

## ■ Der Konzern

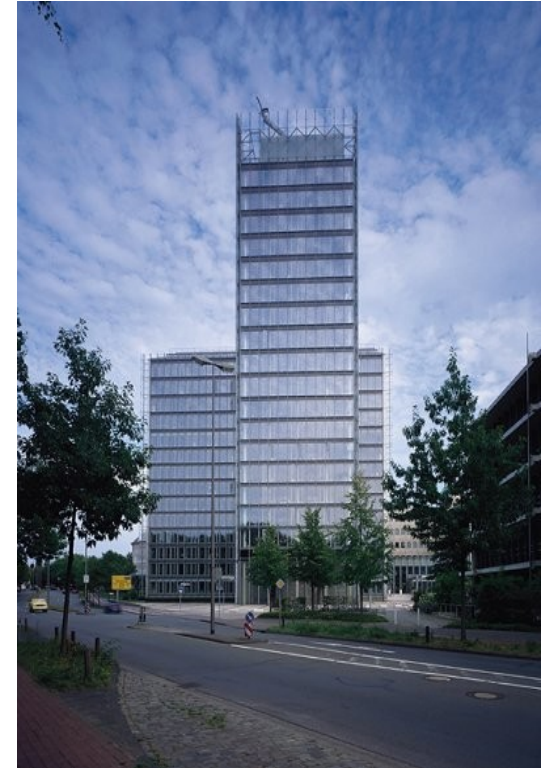
- LVM Landwirtschaftlicher Versicherungsverein a.G.
- LVM Rechtsschutzversicherungs-AG
- LVM Rechtsschutzservice GmbH
- LVM Lebensversicherungs-AG
- LVM Krankenversicherungs-AG
- FSM Fonds-Service Münster GmbH
- LVM Pensionsfonds-AG
- dibera GmbH

## ■ Die Bank

- Augsburger Aktienbank AG

## ■ Unsere Kooperationspartner

- Aachener Bausparkasse AG
- Federated Investors - World-Class Investment Manager
- hkk Erste Gesundheit



# LVM in Zahlen

## ■ Die LVM Versicherung:

- Über 3 Millionen Kunden
- 9,44 Millionen Verträge
- Beitragseinnahmen über 2,46 Milliarden Euro
- 12,4 Milliarden Euro Kapitalanlage
- Einer der Top 5 Kfz-Versicherer bundesweit
- In Deutschland rund 2.100 Agenturen
- In der Unternehmenszentrale in Münster arbeiten über 3.000 Mitarbeiter sowie über 100 Azubis und Praktikanten



# Inhalt

- **SAS-Produktion beim LVM**
- Herausforderungen
- Vorstellung Platform LSF und Process Manager
- Platform LSF beim LVM
- Praxis-Beispiel
- Ausblick

# SAS-Produktion beim LVM

- Data Warehouse beim LVM
  - Konzeptentwurf aus 1990
  - Spartenbezogenes, technisches Modell
  - keine Historisierung
  - fehlende Standards
  - Analysen und Reports laufen auf Snap-Shots
  - Unterschiedliche Plattformen:
    - z/OS und AIX
- Aufbau eines neuen spartenübergreifenden, historisierten, tagesaktuellen Data Warehouses
  - Entscheidung IIS – Insurance Intelligence Solution
  - Standardisierter Detail Data Store als Datenmodell

## SAS-Produktion beim LVM (2)

- Batch-Verarbeitung unter z/OS
  - ca. 2000 einzelne SAS-Programme
  - JCL ruft Script auf
- Batch-Verarbeitung unter AIX
  - ca. 800 SAS-Programme zur Verarbeitung
  - Aufruf erfolgt über Shell-Scripte
  - Einbindung in Host-Scheduler: JCL ruft Script auf
  - Kein Workload-Balancing
  - Logs über stdout an Host zurückgeliefert
- Komplexer Prozess zur Bereitstellung eines SAS-Programms
  - Fest definiertes Antrags- und Freigabeverfahren

# Inhalt

- SAS-Produktion beim LVM
- **Herausforderungen**
- Vorstellung Plattform LSF und Process Manager
- Plattform LSF beim LVM
- Praxis-Beispiel
- Ausblick

# Herausforderungen im Rahmen der IIS

- Aufbau eines spartenübergreifenden,
  - **historisierten**,
  - **tagesaktuellen** Data Warehouses
  
- Implementierung der ETL-Programme als DI-Studio-Jobs
  - Erwartet ca. 1000 Jobs
  - Sehr hoher Aufwand für die Produktionsplanung
  - Lastverteilung?
  
- Integration in SAS-Metadaten
  - Bereitstellung der DIS-Jobs
  - Planung der Jobs in der SAS Management Console
  - Übergabe der Jobs-Flows an den Scheduler direkt möglich
  
- Q4/'08: Evaluierung zur Nutzung eines Schedulers unter AIX

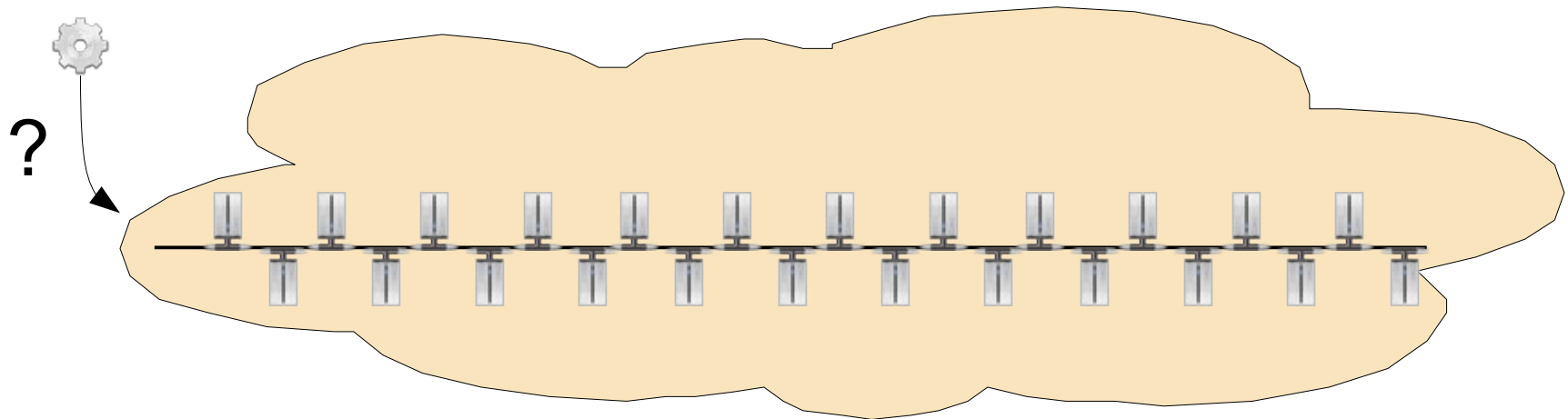


# Inhalt

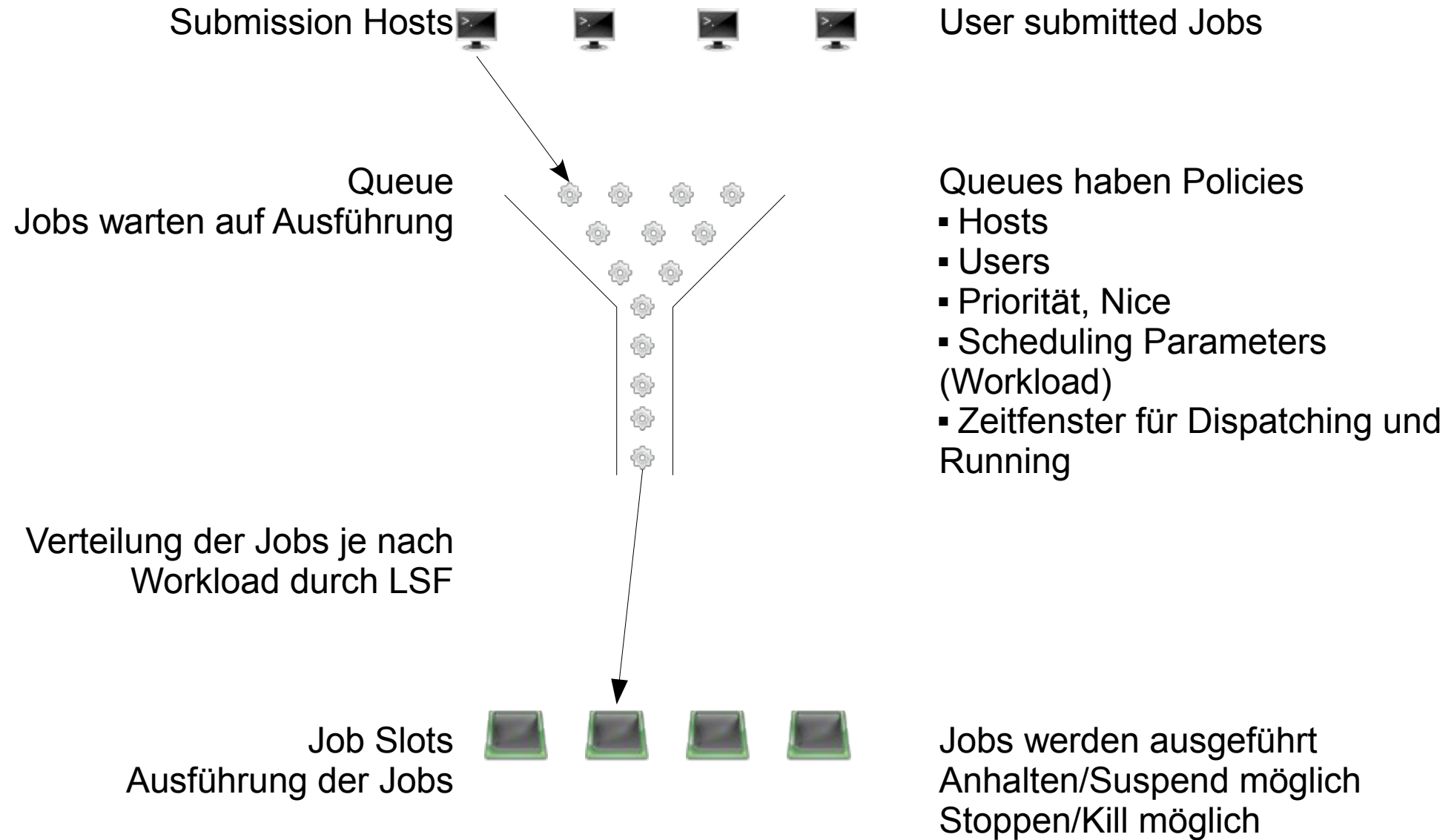
- SAS-Produktion beim LVM
- Herausforderungen
- **Vorstellung Platform LSF und Process Manager**
- Platform LSF beim LVM
- Praxis-Beispiel
- Ausblick

# Was ist Plattform LSF?

- Plattform LSF (Load Sharing Facility) ist eine Kollektion von verteilten Resource-Management-Produkten, die
  - Hosts/Computer in einem Cluster/Grid zusammenfasst
  - den Workload der Systeme monitort/analysiert
  - Jobs verteilt, scheduled und ausbalanciert
- Zentrale Fragestellung:
  - Auf welchem Host/Computer kann mein Job gefahren werden?

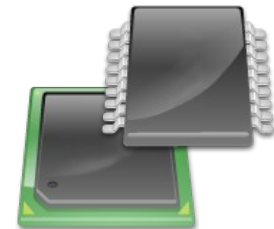


# Wie funktioniert LSF?



# Ressourcen-Management

- Aktueller Stand der Ressourcen auf den Rechnern werden fortlaufend an den Management-Host übertragen
- Eigene Parameter z.B. Auslastung eines Filesystems
- Messbare Ressourcen für Hosts definierbar/vorkonfiguriert
  - Bedarf bei Queues bzw. Jobs definierbar
- Statische Infos vom Host:  
Typ, Model, Name, CPU factor,  
ncpus, ndisks, maxmem. Maxswp, maxtmp
- Dynamische Infos:
  - Run queue length 1 Minute, CPU-Auslastung, Paging Rate, verfügbarer Speicher, verfügbarer Swap, Disk I/O Rate, ...



# Was ist Platform Process Manager?

- Im Platform Process Manager können Jobs in Flows/Netzen organisiert werden
  - Definition der Jobs inkl. Queue und Resource
  - Abhängigkeiten der Jobs sehr fein definierbar
  - NICHT: Abhängigkeiten zwischen Job-Flows
- Ausführung des Job-Flows im Process Manager: wartende Jobs, je nach Abhängigkeiten über LSF submitted
  - Laufender Job im Process Manager läuft nicht sofort auf einem Host, sondern wird erst vom LSF verteilt



# Inhalt

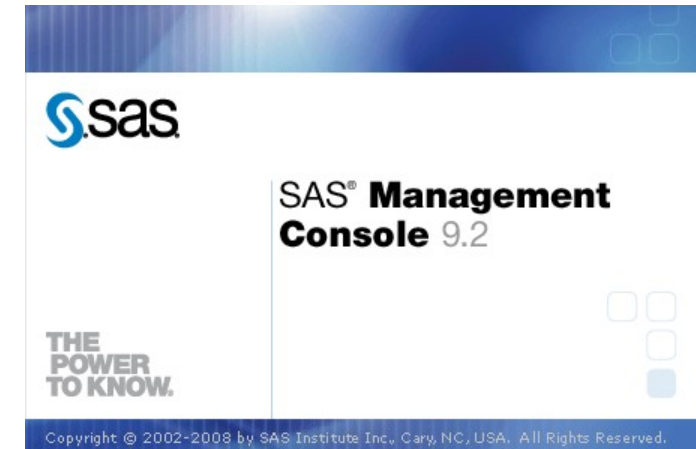
- SAS-Produktion beim LVM
- Herausforderungen
- Vorstellung Platform LSF und Process Manager
- **Platform LSF beim LVM**
- Praxis-Beispiel
- Ausblick

# Einsatz des LSF Schedulers beim LVM

- Plattform LSF und Process Manager sind über SAS in Form einer OEM-Version lizenziert
  - Nachteil: Monitoring nicht möglich da die Plattform Management Console nicht Bestandteil der OEM-Version sind.
- ein Cluster mit genau einem Host:
  - Management Host, Submission Host und Execution Host gleichzeitig
- 4 Queues mit unterschiedlichen Prioritäten definiert:
  - r1m (Run Queue Length 1-minute): 0.8/2.5
  - Priority 30/35/40/45, Nice 10
  - Anzahl Job-Slots entspricht 1,5-fache Anzahl der CPU-Cores
    - 8 CPU-Cores → 12 Job-Slots definiert

# Job-Flows und Jobs

- 80 Job-Flows mit ca. 1700 Jobs
- Seit August 2009 ca. 200.000 Job-Läufe im LSF
- seit SAS 9.2 deutlich bessere Integration in der Management Console
  - Grafische Darstellung der Job-Flows
  - Filter/Suchmasken





# Triggern der Job-Flows

- Jeder Job-Flow wird durch einen Job im Host-Scheduler getriggert
  - Eigenentwicklung: Script zum Triggern und Überwachen eines Job-Flows
  - Erfolgreiche Anbindung des LSF-Schedulers an den Host-Scheduler
  - Keine Event-Steuerung im Process Manager sondern über den Host-Scheduler
    - Abhängigkeiten zwischen Job-Flows im Host-Scheduler hinterlegt

# Inhalt

- SAS-Produktion beim LVM
- Herausforderungen
- Vorstellung Platform LSF und Process Manager
- Platform LSF beim LVM
- **Praxis-Beispiel**
- Ausblick

# Praxis-Beispiel

- In Q1/2010
  - Projekt erhält Anforderung zum Aufbau eines bankfachlichen Data Warehouses
  - Data Marts umfassen ca. 450 Tabellen
    - Davon 270 Referenztabellen
  - Tägliche Lieferung im CSV-Format
    - Datenmenge ca. 1 GB
  - Service-Level
    - Lieferung erfolgt bis 6 Uhr
    - Bereitstellung/Aktualisierung Data Marts bis max. 9 Uhr

## Praxis-Beispiel (2)

- Strategische Entscheidung: Einsatz DI-Studio
- Beladung/Aktualisierung als Einzel-Job je Tabelle
- Zweistufiger Aufbau
  - Data Marts der aktuellen Lieferung (ca. 10-15 GB täglich)
  - Historisierte Data Marts
    - Historienbildung mit aktueller Lieferung
- Datenvolumen
  - Aktuelle Data Marts ca. 10-15 GB
  - Historisierte Data Marts
    - Wachstum ca. 1-2 GB täglich
    - Mittelfristig 600 GB



## Praxis-Beispiel (3)

- Aufbau der Job-Flows
  - Je Data Mart ein Job-Flow
  - Je Tabelle ein DI-Studio-Job
  - ca. 1000 Jobs in insgesamt 21 Job-Flows organisiert
    - mit Vor- und Nachverarbeitung
  - Wenig Abhängigkeiten innerhalb der Job-Flows
  - Herausforderung für Scheduler:
    - sehr viele Jobs mit kleiner Laufzeit
    - Wenige Jobs mit sehr langer Laufzeit (Faktentabellen)

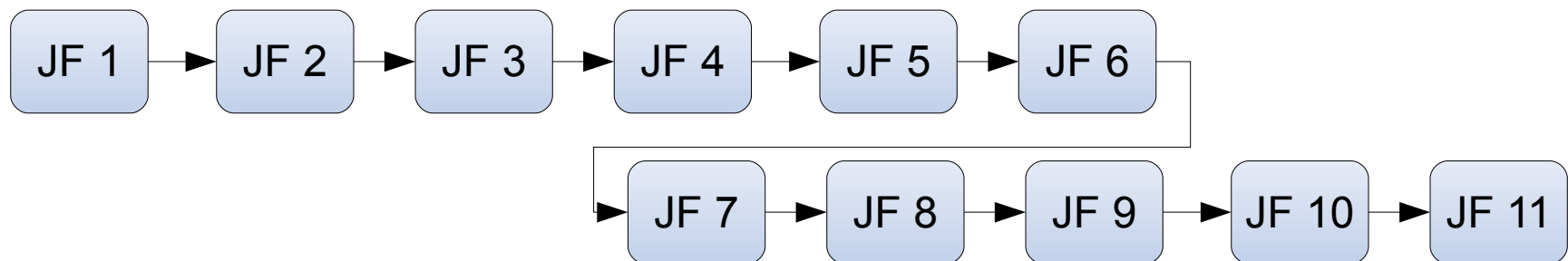
# Praxis-Beispiel (4)

## ■ Laufzeitverhalten:

- Abarbeitung der Jobs nacheinander
  - Keine parallele Verarbeitung!
  - 3 Stunden Laufzeit zum Aufbau der aktuellen Data Marts

## ■ Platform LSF

- Abarbeitung der Job-Flows nacheinander
  - Jobs innerhalb eines Job-Flows werden parallel abgearbeitet
- Laufzeit 1,5 Stunden



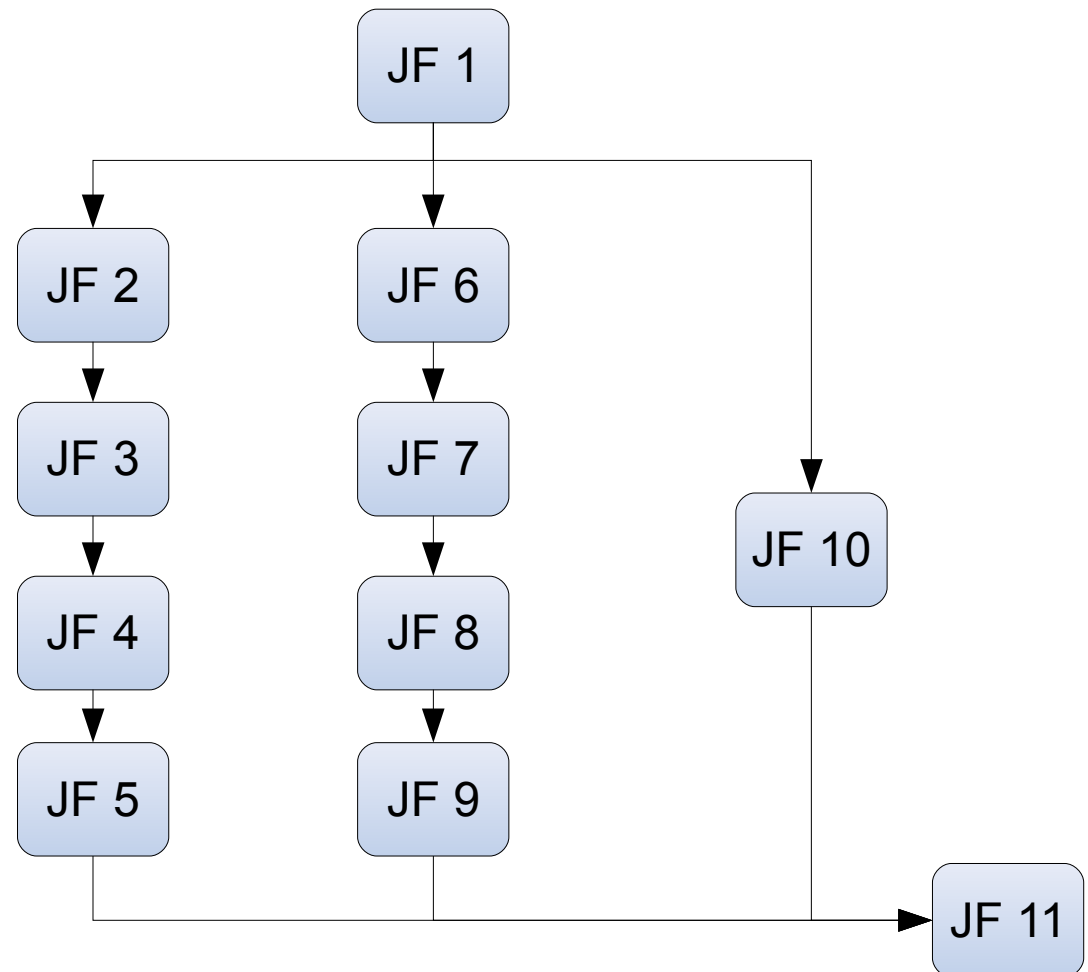
# Praxis-Beispiel (5)

## ■ Parallelisierung der Job-Flows

- Jobs mit kleiner Laufzeit erhalten geringe Priorität
- Job-Flows werden teilweise gleichzeitig gestartet
- Laufzeit ca. 1 Stunde

## ■ Fazit:

- Service-Level konnte mit LSF gehalten werden
- Geringer Aufwand in Prod.-planung



# Ausblick

- SAS-Rechner zukünftig im Cluster betreiben
  - Kapazitäten optimal ausschöpfen
  - Insbesondere „Spitzen“ besser ausbalancieren
- Ausbau des Ressourcen-Management
- ggf. Prüfung einer Voll-Lizensierung
  - Nutzung weiterer Monitoring-Funktionalitäten von LSF

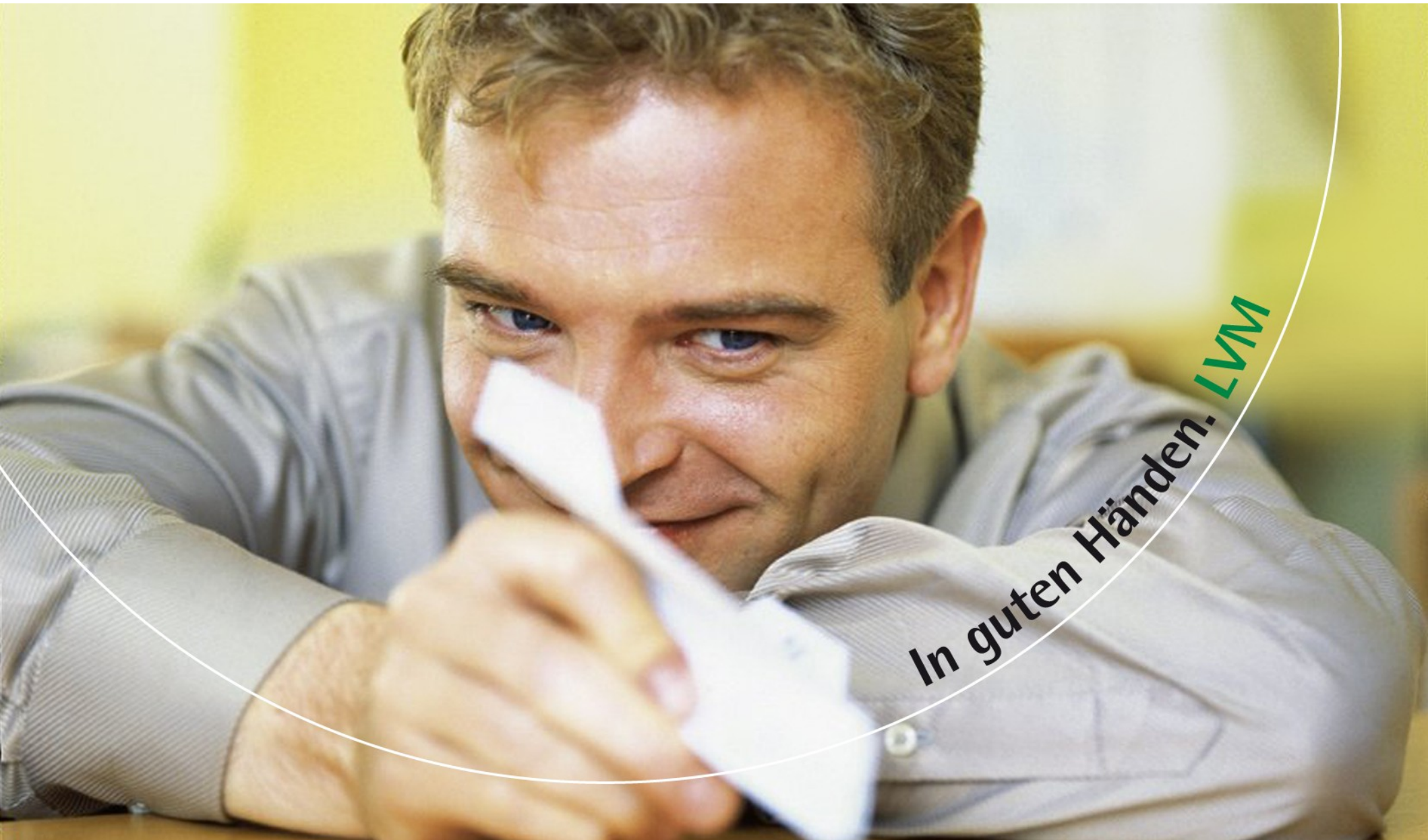






Fragen und Anregungen...

# Vielen Dank für Ihre Aufmerksamkeit!



In guten Händen. **LVM**