

Informationssysteme II

Übungsblatt 1

Christine Pries, Mareike Wagner, Felix Oppermann (Gruppe 18)

20. April 2005

Aufgabe 1 - Nachbereitung der Veranstaltung Informationssysteme I (40 Punkte)

Alles nachgearbeitet!!!

Aufgabe 2 - DWH (20 Punkte)

Im Gegensatz zu operativen Datenbanken enthält ein Data Warehouse nicht nur aktuelle, sondern auch historische Daten. Es kann damit als Grundlage von Analysen, z.B. auch von zeitlichen Verläufen, angewendet werden. Data Warehouses können somit als Unterstützung bei Entscheidungen des Managements dienen. Ein Data Warehouse wird im Gegensatz zu operativen Datenbanken nicht mehr direkt im Produktivbetrieb eingesetzt.

Operative Datenbanken stellen dabei eine der Datenquellen des Data Warehouse dar. Hierbei wird in der Regel eine größere Zahl an Datenquellen verwendet, so dass das Data Warehouse Daten zu verschiedenen Applikationen und Datenbanken enthält. In operativen Datenbanken sind in der Regel nur Daten zu einer Anwendung abgelegt. Die Daten im Data Warehouse sind zudem bereinigt und vorverarbeitet um einen konsistenten und zuverlässigen Datenbestand sicherzustellen. Operative Datenbanken enthalten oft noch fehlerhafte Daten. Die Datenbanken zu verschiedenen Anwendungen haben oft verschiedene Formate. Das Data Warehouse stellt eine integrierte Sammlung von Daten da, wobei die Daten so aufbereitet sein sollten, dass auch der Vergleich von Informationen aus verschiedenen Quellen leicht möglich ist. Durch die große Zahl der Quellen, die meist Informationen aus allen Bereichen eines Unternehmens umfassen besitzen Data Warehouses meist einen größeren Datenbestand als operative Datenbanken.

Aufgabe 3 - Architekturen von Informationssystemen (40 Punkte)

	monolithisch	ausgelagerte Präsentation	Fernzugriff auf Datenbestände
Interaktivität	sehr gut (1)	mittel (2)	schlecht (3)
Datentransfer	gut (7)	mittel (8)	schlecht (9)
Skalierbarkeit	sehr schlecht (13)	schlecht (14)	mittel (15)
Offenheit	schlecht (19)	schlecht (20)	gut (21)
Serverleistung	hoch (25)	hoch (26)	hoch (27)
Clientleistung	kein Client (31)	gering (32)	gering (33)
Testbarkeit	gut (37)	gut (38)	mittel (39)
Portierbarkeit	schlecht (43)	mittel (44)	gut (45)

	Fernzugriff auf verteilte Datenbestände	verteilte Applikation
Interaktivität	schlecht (4)	schlecht (5)
Datentransfer	schlecht (10)	schlecht (11)
Skalierbarkeit	gut (16)	gut (17)
Offenheit	gut (22)	gut (23)
Serverleistung	mittel (28)	mittel (29)
Clientleistung	gering (34)	mittel (35)
Testbarkeit	mittel (40)	mittel (41)
Portierbarkeit	gut (46)	gut (47)

	verteilte Applikation (eigenes DBMS)
Interaktivität	mittel (6)
Datentransfer	mittel (12)
Skalierbarkeit	gut (18)
Offenheit	gut (24)
Serverleistung	mittel - gering (30)
Clientleistung	mittel - hoch (36)
Testbarkeit	schlecht (42)
Portierbarkeit	gut (48)

1. Direkter Datenfluss zwischen allen Komponenten. Keine Netzwerkkommunikation nötig.
2. Update des Client ggf. schwierig. Jede Änderung benötigt Netzwerkkommunikation.
3. Änderungen im Datenbestand müssen über zwei Netzwerkschnittstellen zum Client propagiert werden.
4. Änderungen im Datenbestand müssen über zwei Netzwerkschnittstellen zum Client propagiert werden. Zusätzlich Synchronisation der Datenbanken nötig.
5. Wie bei (4)
6. Manche Operationen können vom Client unabhängig bearbeitet werden. Änderungen im lokalen Datenbestand können direkt propagiert werden.
7. Keine Netzwerkkommunikation nötig.
8. Nur Transfer der Ergebnisdaten zum Client.
9. Doppelte Netzwerkkommunikation, zum einem zwischen Anwendung und Client und zum anderen zwischen Anwendung und DBMS.
10. Wie bei (9)
11. Ähnlich wie bei (9). Abhängig von der Anwendung, kann zwischen Client und Anwendungslogik ein geringeres oder höheres Transfervolumen nötig sein.
12. Durch Zwischenspeichern der Daten ist es möglich das Transfervolumen geringer zu halten.
13. Schlecht, da die ganze Anwendung auf einem Rechner laufen muss. Skalierung ist nur durch Erhöhung der Leistungsfähigkeit dieses Rechners möglich.
14. Ein Server kann mehrere Clients bedienen. Für den Server gilt das gleiche wie bei (13).
15. Etwas besser als (14), da die Datenbank und die Anwendungslogik auf verschiedenen Rechner laufen. Beide lassen sich bei Bedarf separat erweitern.
16. Das System kann bei Bedarf leicht durch weitere DBMS-Server erweitert werden.
17. wie bei (16)
18. wie bei (16)
19. DBMS ist fester Bestandteil der Anwendung und somit schwer austauschbar.
20. wie bei (19)
21. DBMS ist separater Systembestandteil und relativ leicht auszutauschen.
22. wie bei (21)
23. wie bei (21)
24. wie bei (21)
25. Wir betrachten das ganze System als Server. Die Anforderung an diesen ist hoch, da er alle Aufgaben übernehmen muss. Meist wird ein solches System jedoch nicht von mehreren Benutzern gleichzeitig verwendet.
26. Etwas geringer als bei (25), da die Präsentation ausgelagert ist.
27. Die verschiedenen Aufgaben sind auf zwei Rechner verteilt, so dass die Anforderungen an diese Systeme etwas geringer sind. Zudem ist es möglich jeweils auf die Aufgabe optimierte Systeme zu verwenden.
28. Zusätzlich zu (27) eine Verteilung der Datenbanken. Folglich sind die Anforderungen an die einzelnen Systeme geringer.
29. Wie bei (28). Zusätzlich wird ein Teil der Last vom Client übernommen.
30. Wie bei (28). Zusätzlich übernimmt der Teil einen Teil der Last und kann auch durch seine eigene Datenbank Serverzugriffe vermeiden.

31. Kein Client, wir betrachten das System hier als Server.
32. Sehr geringe Anforderungen an den Client, da dieser nur die Darstellung der Daten übernehmen muss. Für viele Anwendungen reicht z.B. ein Webbrowser und somit ein relativ leistungsschwaches System.
33. Wie bei (32)
34. Wie bei (32)
35. Der Client übernimmt einen kleinen Teil der Berechnungen, so dass er etwas leistungsfähiger als bei (32) sein sollte.
36. Wie bei (32). Zusätzlich etwas höhere Speicherplatzanforderungen.
37. Nur ein System, welches unabhängig zu testen ist. Ggf. ist der Test der einzelnen Komponenten schwieriger.
38. Wie bei (37). Client muss zusätzlich getestet werden.
39. Zusätzlich zu den Tests der einzelnen Komponenten muss auch die Kommunikation getestet werden. Die Komponenten für sich genommen sind ggf. leichter zu testen.
40. Wie bei (39)
41. Wie bei (39). Test des Clients ist aber aufwändiger.
42. Wie bei (39). Test des Clients ist aber aufwändiger.
43. Es muss immer das gesamte System portiert werden.
44. Portierung des Servers relativ aufwändig, da das gesamte System portiert werden muss. Für den Client sehr einfach, da dieser nur die Darstellung übernimmt. (Ggf. nur ein Webbrowser)
45. Portierung einfacher, da die einzelnen Komponenten separat portiert werden können. Die Trennung vereinfacht auch Anpassungen, sofern die Schnittstellen robust sind. Client wie bei (44).
46. Wie bei (45)
47. Wie bei (45). Portierung des Clients wieder schwieriger, da er auch Teile der Berechnungen durchführt.
48. Wie bei (45). Portierung des Clients wieder schwieriger, da er auch Teile der Berechnungen und Datenerhaltung übernimmt.