

ppi

systematisches Performancetuning als Projektaufgabe

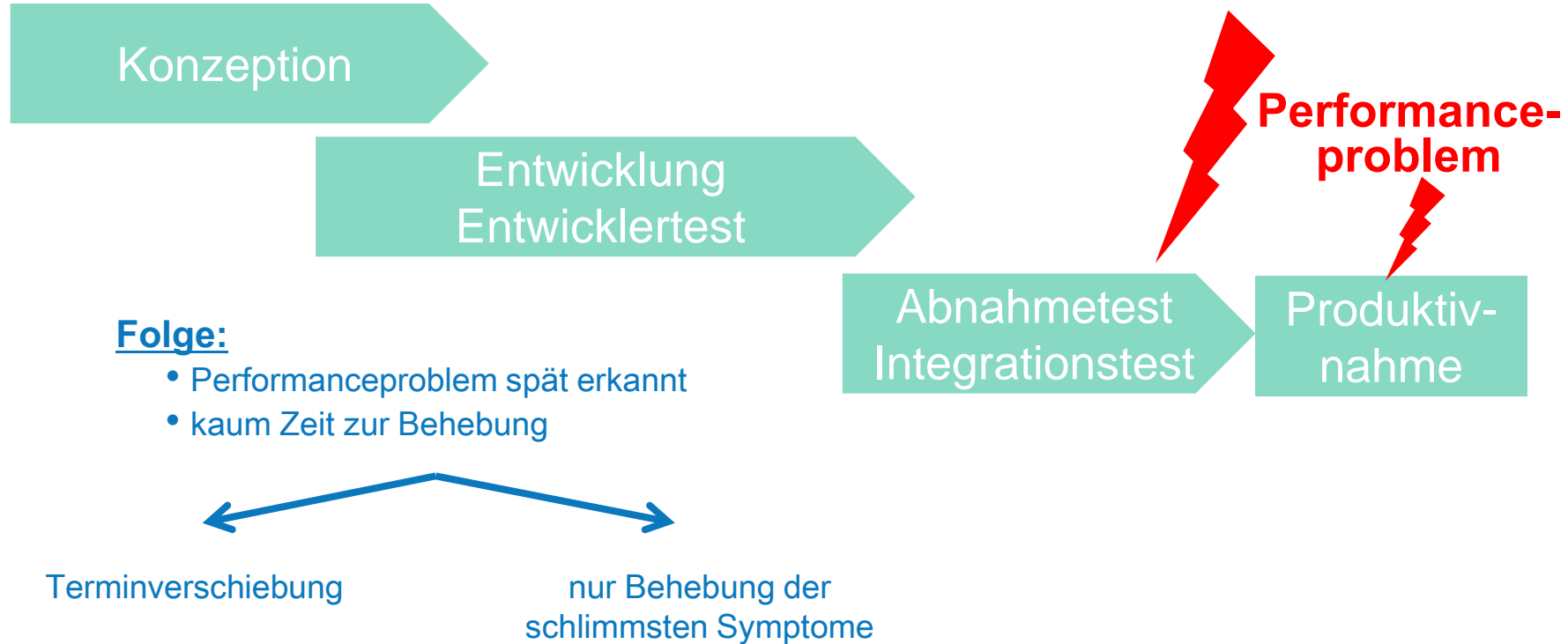
Konferenz Data Analytics März 2019

Agenda

- 1** Projektvorgehen
- 2** konkrete Umsetzung
- 3** Empfehlung
- 4** Beispiele

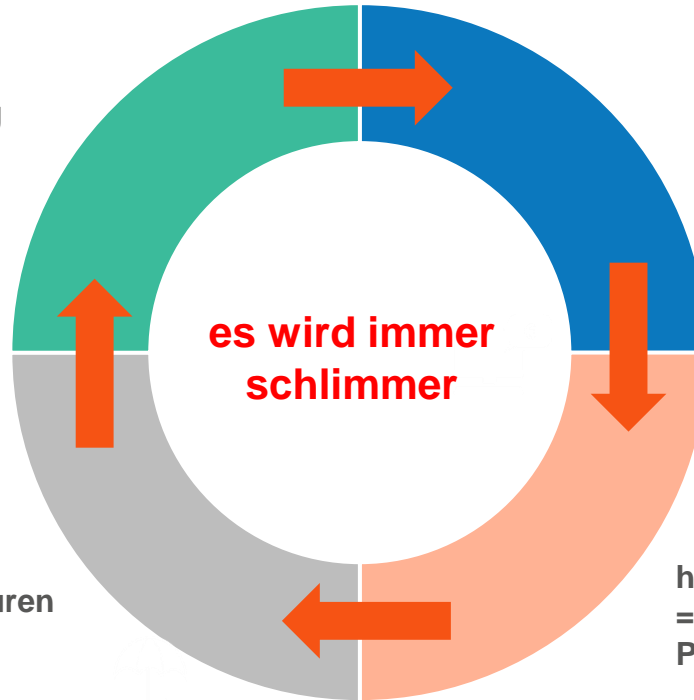
01 Projektvorgehen

häufiger Projektablauf



keine Zeit für
Performanceoptimierung

schlechte Performance
=> lange Testlaufzeiten
=> weniger Testzyklen

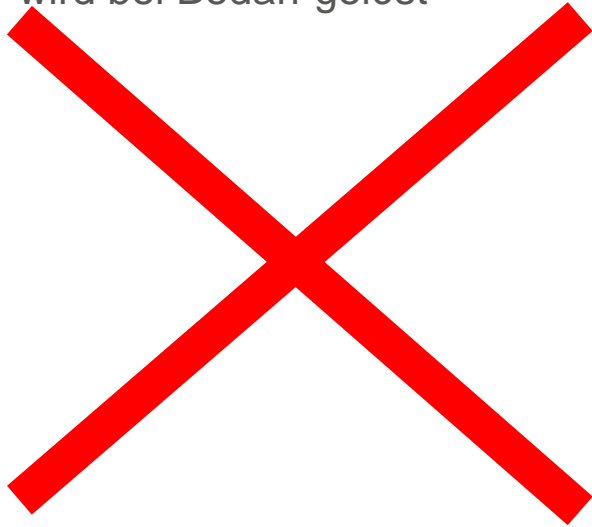


Aufwand steigt
• mehr Tagesgeschäft
• AdHoc Fehlerkorrekturen

höhere Fehlerrate
=> mehr Fehler im
Produktionsbetrieb

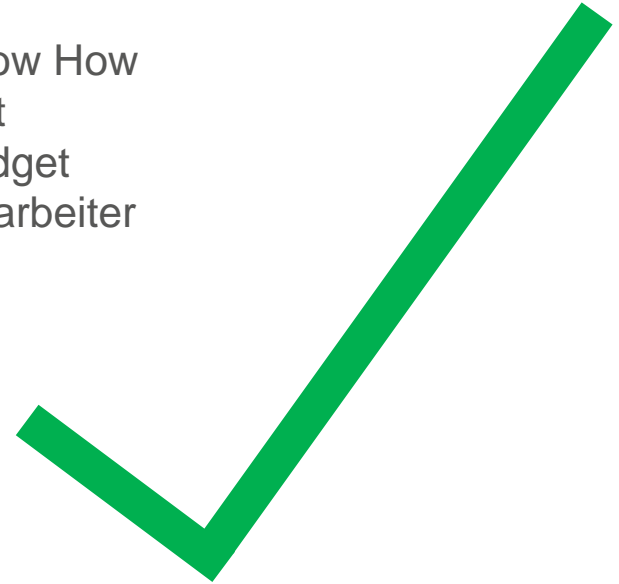
Performance ist Projektrisiko

- muss berücksichtigt werden
- wird bei Bedarf gelöst



Performance muss man planen

- Know How
- Zeit
- Budget
- Mitarbeiter



Varianten Performancediskussion

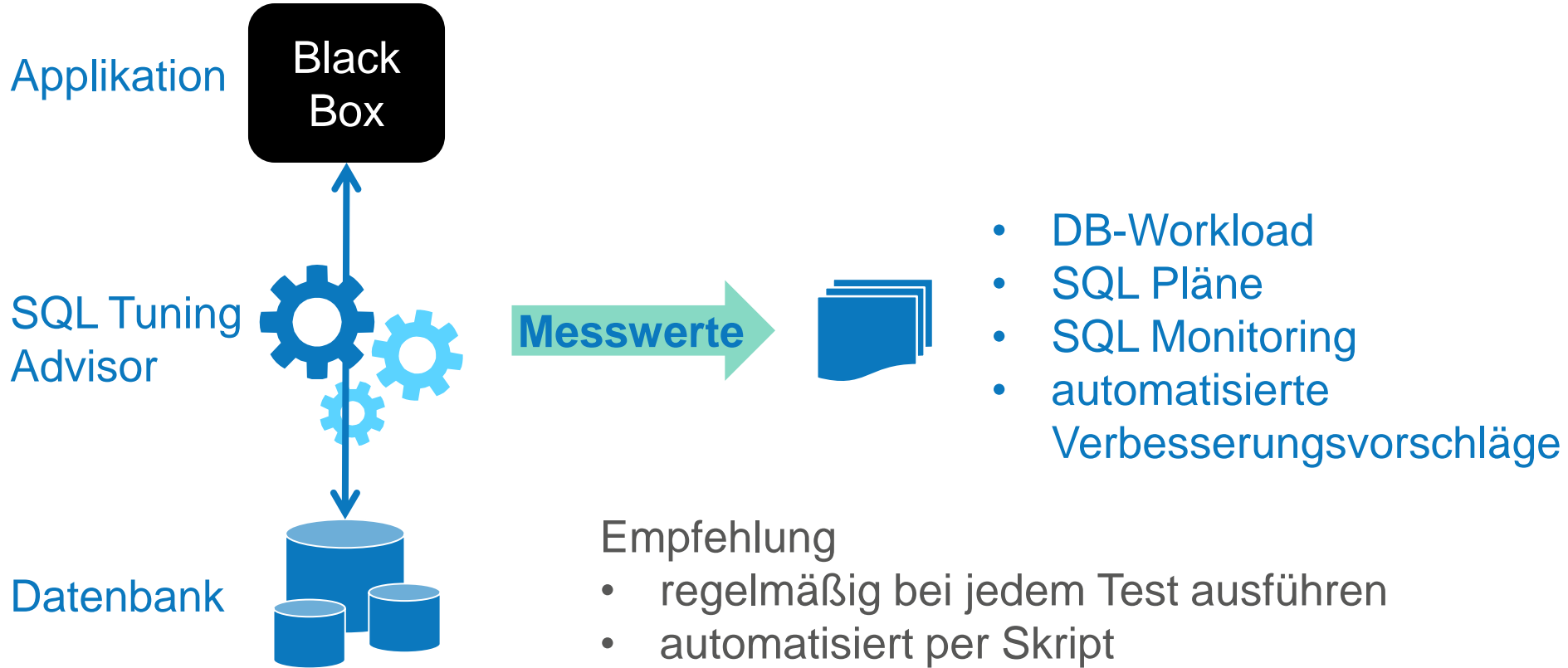


- das Problem liegt immer an einer unerwarteten Stelle
- keine Diskussion vorab in einer Taskforce
- konkrete Lösungssuche aufgrund von Messwerten und Fakten

02 konkrete Umsetzung

Black Box Analyse

konkrete Umsetzung => Black Box Analyse



Ergebnisse der Black Box Analyse

Gesamtlaufzeit der Applikation im Test

Non-DB Anteil

DB Anteil

SQL 1

SQL 2

SQL 3

...

wie oft ausgeführt?

wie viele Zeilen

Warteevents

Monitoring

Ausführungsplan

⇒ klares Bild wo die Probleme liegen

⇒ Ursachen bekannt

⇒ konkrete detaillierte
Lösungsdiskussion möglich

⇒ keine Vermutungen

SQL_ID	SQL_TEXT	MODULE	ACTION	EXECUTIONS	ROWS_PROCESSED	ELAPSED_TIME	SQL_TIME	BUFFER_GETS	DISK_READS	DIRECT_WRITES	FETCHES
1	7t1xx4g64xyu2	select ... osh@zle10030 (TNS V1-V3)	(null)	1	1499754	3997571033	260655384	112092546	2263286	3092889	751
2	67sldq2cdqwt3	INSERT ... SAT	JB_Load...	21726	43451265	1139608075	266254514	132877768	2134	0	0
3	0d2rs013761s0	with wa... osh@zle10030 (TNS V1-V3)	(null)	1	48001	994716024	522389152	12552535	657474	656849	25
4	4xnwm0fm7unt4	WITH FU... osh@zle10030 (TNS V1-V3)	(null)	1	0	798911583	235836350	254008	374924	382277	0
5	gcczhy49gbd5c	select ... osh@zle10030 (TNS V1-V3)	(null)	1	0	797147000	559502147	323420	128997	129838	0
6	gwtcpbd21wxdm	with st... osh@zle10030 (TNS V1-V3)	(null)	1	37564001	735239144	359565924	44067	135663	118420	18783
7	34zmhzy0p3jwh	INSERT ... SAT	JB_Load...	4406	8811918	563746269	341429762	26956727	964	0	0
8	373b7v3tvvgbdz	INSERT ... SAT	JB_Load...	18785	37567310	551264979	389708892	116906004	1272	0	0
9	9xdsn9cff4w7d	select ... osh@zle10030 (TNS V1-V3)	(null)	1	65478001	544358915	493060053	211410	325957	326256	32740
10	34zmhzy0p3jwh	INSERT ... SAT	JB_Load...	4407	8812036	533984798	340510780	26824258	749	0	0
11	27gtkz06wqsdu	INSERT ... SAT	JB_Load...	3149	6297964	532985856	224194361	19192555	1769	0	0
12	27gtkz06wqsdu	INSERT ... SAT	JB_Load...	3150	6299783	524029020	221692426	19173907	1206	0	0
13	f50rbhyk9k2tt	with st... osh@zle10030 (TNS V1-V3)	(null)	1	43340001	519640956	624269963	170992	59351	59620	21671
14	15a0zxh5qz26x	with st... osh@zle10030 (TNS V1-V3)	(null)	1	37528001	472945369	334272225	21995	68564	65938	18765
15	fpb14c863jdaf	select ... osh@zle10030 (TNS V1-V3)	(null)	1	37567310	472443219	209166241	44250	123353	110398	18785
16	90phmltrhhpdz	INSERT ... SAT	JB_Load...	1185	2369177	458571554	134610620	6896251	743	2	0
17	90phmltrhhpdz	INSERT ... SAT	JB_Load...	1183	2365874	448895006	139522838	6892112	601	0	0
18	993fcgyyxr0py	with st... osh@zle10030 (TNS V1-V3)	(null)	1	17540001	445985732	213084629	10934	64497	62572	8771
19	f6wfzcnx9894m	with st... osh@zle10030 (TNS V1-V3)	(null)	1	20376001	423434226	235705817	18367	83153	72484	10189
20	2h0wskc8bu84p	DECLARE... SQL*Plus	(null)	1	1	419379562	333795192	15218235	2756	306	0
21	13r75nbcs0r5w	with st... osh@zle10030 (TNS V1-V3)	(null)	1	17586001	413202046	215174640	11414	64957	62572	8794
22	6pur6ya7rsc48	with re... SAT	JB_BDV_...	1	18278001	382131183	248414024	135509	188546	179630	9140
23	57249dsv3rly8	with st... osh@zle10030 (TNS V1-V3)	(null)	1	37216001	376129963	319239759	22160	48958	41807	18609
24	29ur4kn8gmvzf	with st... osh@zle10030 (TNS V1-V3)	(null)	1	16258001	374413654	201838347	57169	57749	58161	8130
25	3qgbadnrxnggwu	WITH FU... osh@zle10030 (TNS V1-V3)	(null)	1	0	360347377	225941134	691017	423205	334235	1
26	6z4cqy0r0fb04	with st... osh@zle10030 (TNS V1-V3)	(null)	1	17506001	356057744	193089285	9005	65232	62432	8754
27	6jmp604qsphzk	select ... osh@zle10030 (TNS V1-V3)	(null)	1	17623954	351094521	121217963	66711	148558	140958	8813
28	asvy8f80bg5kt	select ... osh@zle10030 (TNS V1-V3)	(null)	1	12597747	308205727	110681792	80076	169346	131799	6300
29	aph0zuwtgvhlt	with FU... osh@zle10030 (TNS V1-V3)	(null)	1	0	298466585	295086215	162425	6405	6405	1
30	9vvqgz4mz2afy	with FU... osh@zle10030 (TNS V1-V3)	(null)	1	0	286567471	163820376	17396	123396	120011	0

SQL Plan Monitoring Details (Plan Hash Value=932927367)

Id	Operation	Name	Rows (Estim)	Cost	Rows (Actual)	Activity (%)	Activity Detail (# samples)
0	SELECT STATEMENT				18M	4.62	Cpu (6)
1	SORT ORDER BY		18M	59979	18M	93.85	Cpu (64)
							acknowledge over PGA limit (3)
							direct path read temp (55)
2	PARTITION LIST SINGLE		18M	339	18M		
3	TABLE ACCESS STORAGE FULL	TABELLEX	18M	339	18M	1.54	Cpu (1)
							cell multiblock physical read(1)

neuer Projektablauf in jedem Release

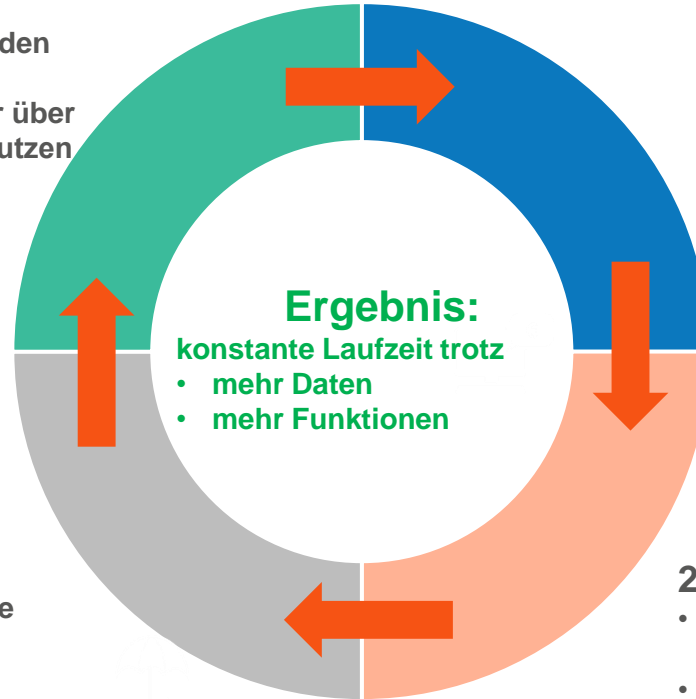
4. Diskussion mit Entwicklern

- Wie muss Applikation geändert werden damit sie schnelle SQL verwendet?
- evtl. auch Diskussion mit Anwender über „teure“ Anforderungen mit wenig Nutzen
(z.B. Protokollanzeige nur 2 Jahre rückwirkend statt vollständig)
- Anpassung Applikation

3. SQL modifizieren

- umschreiben
- Parameter, Komprimierung, Partitionierung u.a. prüfen
- evtl. auch minimale funktionale Änderungen

Ziel : SQL ist schnell



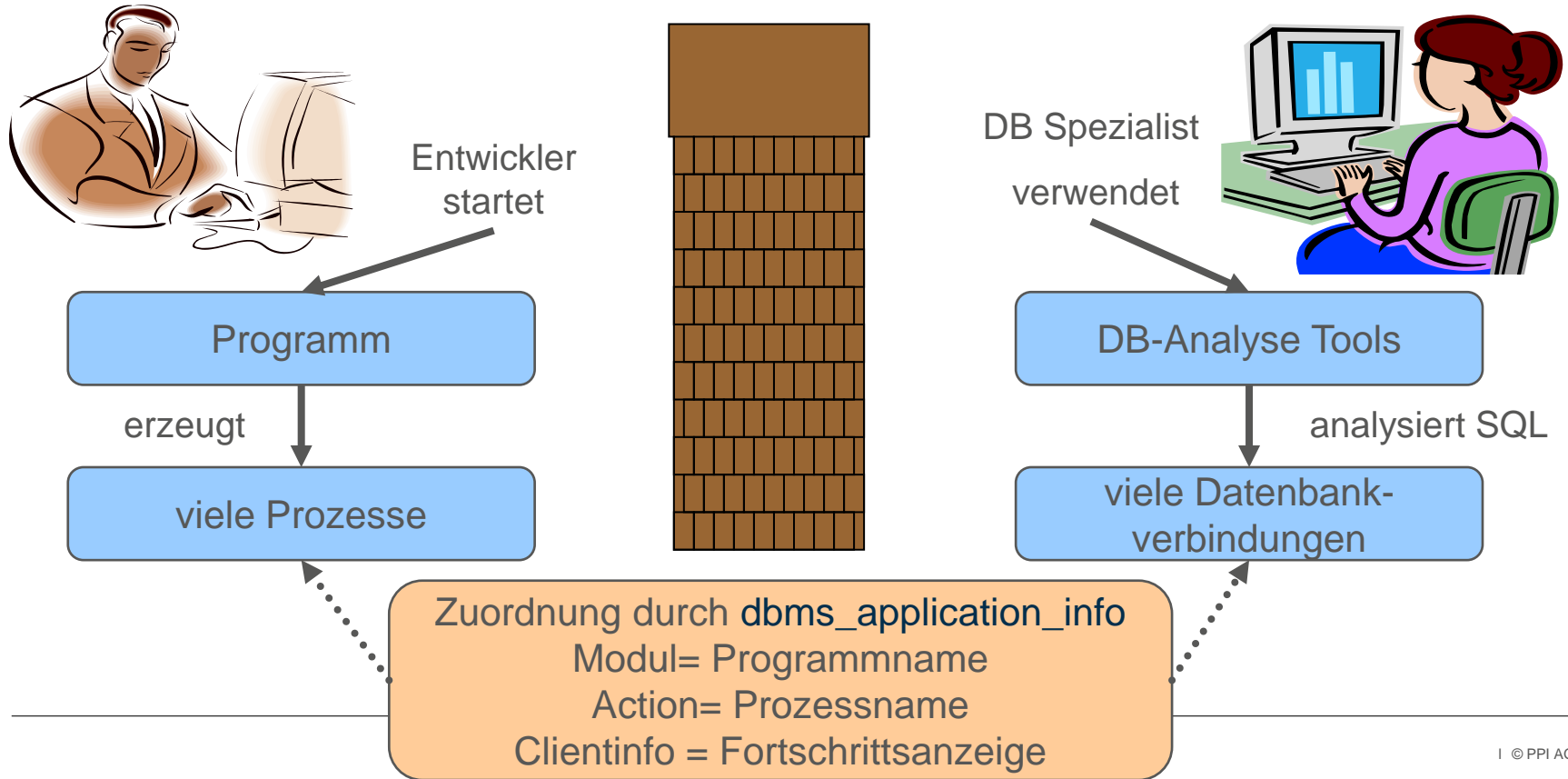
1. Black Box Analyse

- Messwerte
- Fakten

2. Top-3 SQL analysieren

- repräsentieren häufig über 90% der Gesamtlaufzeit
- Ursachen langer Laufzeit ermitteln

Kommunikation DB-Spezialist / Entwickler



Ausgaben

- Zeit, Budget, Mitarbeiter für Analysen
- Erarbeitung von Empfehlungen
- Umsetzung der Empfehlungen

Einnahmen

- ungeplante Ausgaben für „adHoc“ Optimierung am Projektende entfallen
- effiziente Entwicklung
- schnellere Testzyklen
- Frage „Hat mein Code das gewünschte Ergebnis?“ schneller beantwortet

Vorteile

- agieren statt reagieren
- hohe Effizienz
- zufriedene Anwender
- entspanntes Arbeiten

=> netto entstehen keine Kosten

03 Empfehlung

1. Performance planen
2. messen
3. keine Vermutungen
4. Know How aufbauen
5. Tuning Advisor aktiv nutzen
6. Verknüpfung DB / Applikation über `dbms_application_info`
7. regelmäßig in jedem Release

**=> Performanceprobleme nicht lösen
sondern von Anfang an vermeiden**

04 Beispiele

Messwert

- Top SQL Laufzeit 0,01ms
- 1 Mio. Ausführungen pro Stunde

Ursache

- versteckte Abfrage auf TabelleX in PL/SQL Funktion
- Aufrufe

```
select  FeldA,  
        PL/SQL Funktion(FeldB, FeldC)  
from  TabelleA  
fragen TabelleX pro Datensatz ab
```

Lösung

- SQL führt Join auf TabelleX direkt aus
- PL/SQL Funktion enthält nur noch Berechnungslogik und kein Subselect

Messwert

- 95% der Laufzeit entfällt auf ein Delete Statement

Ursache

- Applikation ist restartfähig
- löscht evtl. vorhandene Ergebnisse und trägt neu berechnete Ergebnisse ein

Lösung

- gezielte Partitionierung
- statt delete wird truncate Partition aufgerufen

Messwert

- keine Index-/Partitionsverwendung im Top SQL

Ursache

- where Bedingung `Quartal=032018`
- Feld `Quartal` aber alphanumerisch definiert
- daher **Datentyp** bei Abfrage nicht beachtet

Lösung

- Korrektur auf `Quartal='032018'`

Messwert

- Top SQL fragt extrem viele Daten ab und aggregiert diese

Ursache

- keine Ausführung mit Parallel Query
- fehlerhafte Konfiguration **Parallel Query** Parameter

Lösung

- Parallel Query korrekt konfigurieren

Messwert

- Optimizer schätzt Ergebnismenge des SQL auf 5 Zeilen
- tatsächlich werden 250 Mio. Ergebnisse zurückgegeben

Ursache

- in den Join Bedingungen wird mehrfach `nvl (FeldA, 'unknown')` = `nvl (FeldB, 'unknown')` verwendet

Lösung

- fachliche Diskussion an welchen Stellen `nvl` wirklich notwendig ist
- speichern von Defaultwert `'unknown'` wenn keine Information geliefert wird

Messwert

- Optimizer schätzt Ergebnismenge des SQL auf 1 Zeile
- tatsächlich werden 120 Mio. Ergebnisse zurückgegeben

Ursache

- in den Join Bedingungen sind fachlich **redundante** Daten enthalten wie `Abteilung='XYZ' and Product = 'ABC'`
- Abteilung XYZ ist aber nur für das eine Produkt ABC zuständig so dass durch die zweite Bedingung praktisch keine Sätze mehr ausgeschlossen werden

Lösung

- anlegen von korrelierten Statistiken für die Feldkombination Abteilung/Produkt

Messwert

- massive „cluster **wait**“ Events für Massen-Insert Statement

Ursache

- mehrere Prozesse schreiben gleichzeitig in die gleiche Tabelle
- diese Prozesse sind auf unterschiedlichen Knoten des RAC angemeldet
- beim Schreiben ist erheblicher Kommunikationsaufwand zwischen den Knoten notwendig

Lösung

- Kunde 1: Anpassung der DB-Anmeldung so dass Prozesse, die in die gleiche Tabelle schreiben, auf dem gleichen Knoten arbeiten
- Kunde 2: Partitionierung der Tabellen, so dass Partitions- und Prozessstruktur sich 1:1 entsprechen

Messwerte

- können sehr unterschiedlich sein

Ursache

- es gibt Vielzahl von Ursachen
- oft sind es „Kleinigkeiten“

Lösung

- **systematisches** Vorgehen
- Analyse auf Basis Messwerte und Fakten

Probleme vermeiden und nicht lösen

1. Performance planen
2. systematisch vorgehen
3. verursacht keine Zusatzkosten
4. Nutzen entsteht im Produktivbetrieb und während Entwicklung / Test



Jörg Stahnke

Principal Software Engineer

T +49 402227433-1329

M +49 160 3841771

joerg.stahnke@ppi.de

www.ppi.de

PPI AG

Moorfurthweg 13

22301 Hamburg

Wollen / müssen Sie ihre Performance verbessern?

Wir helfen Ihnen gern dabei.

Weitere Fragen?

Kommen sie zum Experten Roundtable um 13:45 Uhr