Software Engineering Conference Russia

October 2017, St. Petersburg

Enterprise class SQL Server monitoring in distributed production environments with high number of servers

Roman Dimenko

About myself

- DBA and DBD since early 2000`s
- Managed distributed DB teams up to 20 people
- Worked with DELL EMC, Microsoft, Spar, McDonald`s
- High load OLTP databases, DWH
- Industries: telecom, fintech, adtech, retail, cloud

Statement of the problem

- Database monitoring in general

Statement of the problem

- Database monitoring in general
- Proactive and reactive

Statement of the problem

- Database monitoring in general
- Proactive and reactive
- Cost

Statement of the problem

- Database monitoring in general
- Proactive and reactive
- Cost
- Skills

Statement of the problem

- Database monitoring in general
- Proactive and reactive
- Cost
- Skills
- Scalability

Statement of the problem

- Database monitoring in general
- Proactive and reactive
- Cost
- Skills
- Scalability
- Levels of monitoring (from hardware to business logic)

Statement of the problem

- Database monitoring in general
- Proactive and reactive
- Cost
- Skills
- Scalability
- Levels of monitoring (from hardware to business logic)
- Service models (managed service) and service providers and what they use

Comparison of existing solutions

- Monitoring
- Tuning
- SQL Server
- Windows
- Network
- Reporting
- Alerting
- Advice

Features

- Historical monitoring data

Features

- Historical monitoring data
- Close to real-time

Features

- Historical monitoring data
- Close to real-time
- Alerts via email

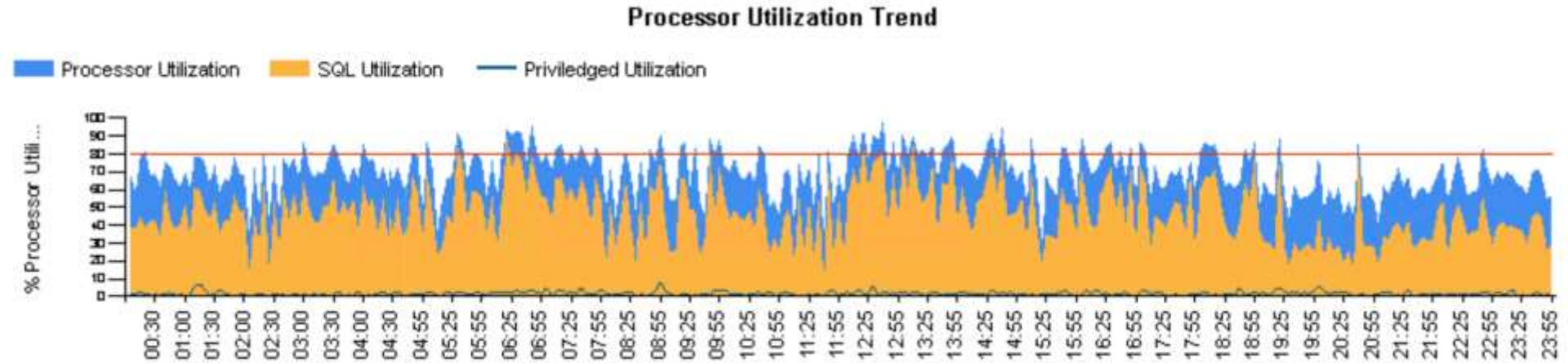
Features

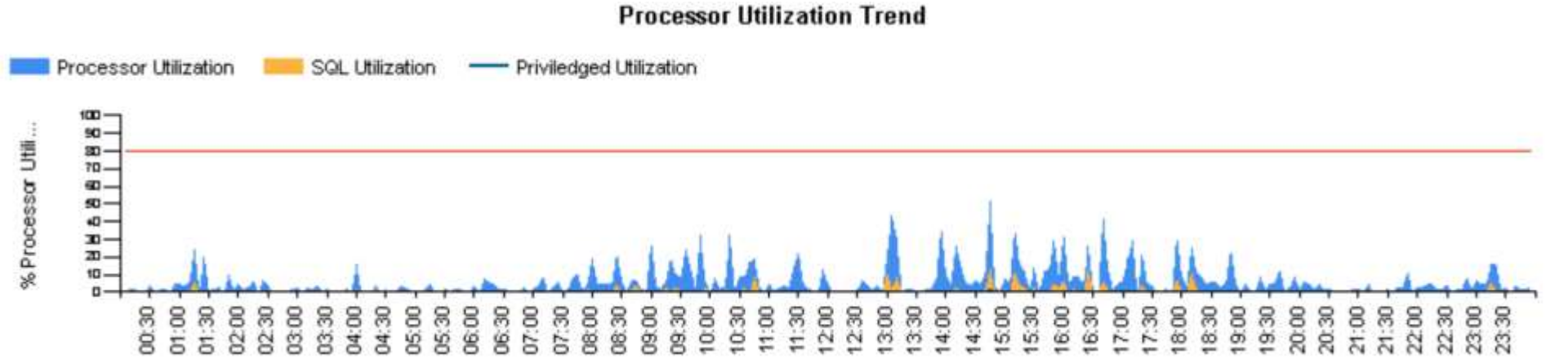
- Historical monitoring data
- Close to real-time
- Alerts via email
- Customizable thresholds

Features

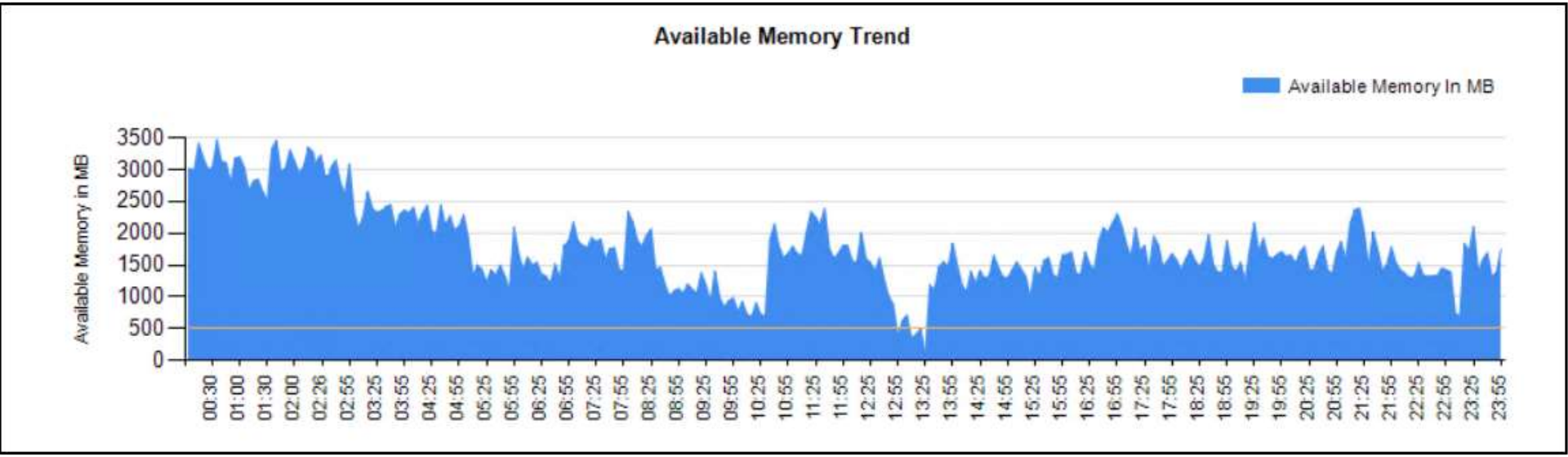
- Historical monitoring data
- Close to real-time
- Alerts via email
- Customizable thresholds
- Advanced SQL Server features monitoring

CPU load


Processor Utilization Trend
Processor Utilization
SQL Utilization
Priviledged Utilization
% Processor Utili...
00:30
01:00
01:30
02:00
02:30
03:00
03:30
04:00
04:30
04:55
05:25
05:55
06:25
06:55
07:25
07:55
08:25
08:55
09:25
09:55
10:25
10:55
11:25
11:55
12:25
12:55
13:25
13:55
14:25
14:55
15:25
15:55
16:25
16:55
17:25
17:55
18:25
18:55
19:25
19:55
20:25
20:55
21:25
21:55
22:25
22:55
23:25
23:55


Processor Utilization Trend
Processor Utilization
SQL Utilization
Priviledged Utilization
% Processor Utili...
00:30
01:00
01:30
02:00
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00
06:30
07:00
07:30
08:00
08:30
09:00
09:30
10:00
10:30
11:00
11:30
12:00
12:30
13:00
13:30
14:00
14:30
15:00
15:30
16:00
16:30
17:00
17:30
18:00
18:30
19:00
19:30
20:00
20:30
21:00
21:30
22:00
22:30
23:00
23:30

Memory consumption



Further plans

- Windows monitoring

Further plans

- Windows monitoring
- Other RDBMS (PostgreSQL, MySQL, Oracle)

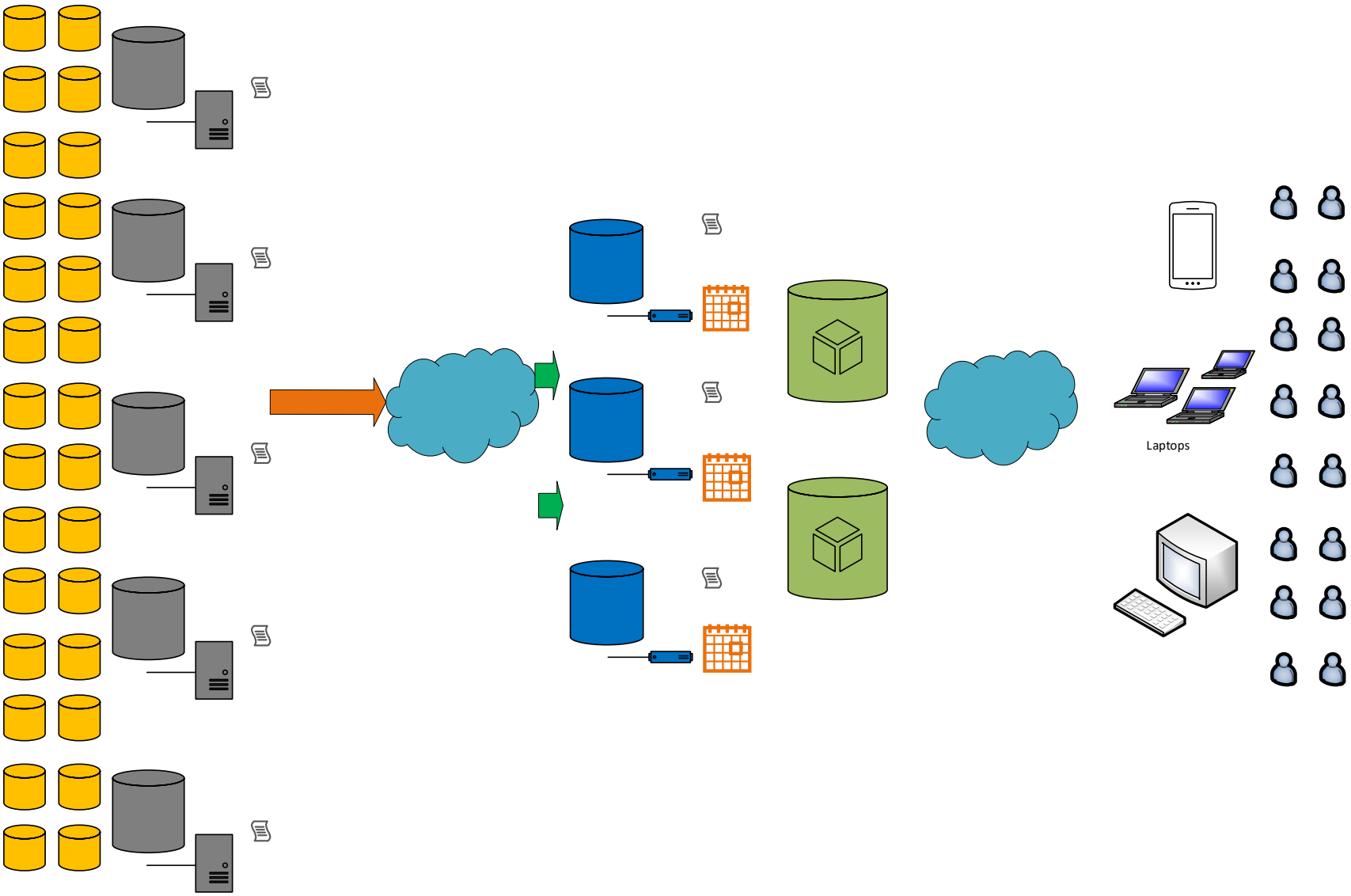
Further plans

- Windows monitoring
- Other RDBMS (PostgreSQL, MySQL, Oracle)
- Linux

Further plans

- Windows monitoring
- Other RDBMS (PostgreSQL, MySQL, Oracle)
- Linux
- AI, Data mining

Architecture



Problems solved

- More than 100 counters
- Machine Metrics
- SQL Server metrics
- Database metrics
- High Availability metrics (AO, Logshipping, Mirroring)
- BI metrics (SSRS, SSAS)

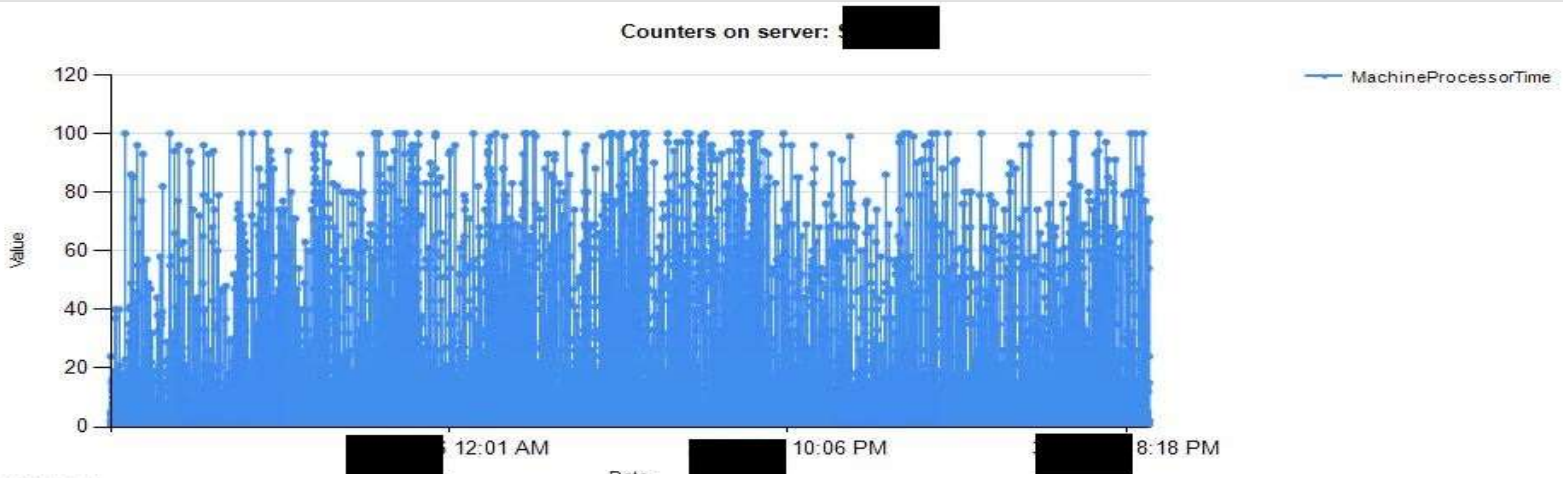
Problems solved

- More than 100 counters
- Trend analysis and prediction

Problems solved

- More than 100 counters
- Trend analysis and prediction
- Ability to drill down from high level problem to root cause

Автоматический мониторинг загрузки процессора на сервере



Activity Report

Server Name	SQL Query	Database Name	Program Name	dd hh mm ss mss	Session ID	Login Name	Wait Info
SQLCLUSTER2	<?query -- sp_server_diagnostics --?>	master	Microsoft® Windows® Operating System	00 00:12:27.426	52	NT AUTHORITY\SYSTEM	(263ms) SP_SERVER_DIAGNOSTICS_SLEEP
SQLCLUSTER2	<?query -- select p.[redacted] , p.[redacted] , p.employeeid , p.lastupdated , sourcetype = case when p.[source] = 1 then 'O[redacted]nber' else '[redacted]Number' end , createuser = 'DailyJob' , createdate = getdate() into #[redacted]umbers_[redacted]sprogram from Invent[redacted]bers p (nolock) where exists (select 1 from [redacted]Services_Subscriptio[redacted]mployeeID m (nolock) where AddonID = 14 and SubscriptionStatus IN ('O[redacted]', 'Trial') and AddOnStatus != 'Expired' and m.employee_id = p.employeeid and not exists(select * from [redacted]teparts where p[redacted]eparts.employeeid = p.employeeid)) and p.[source] = 1 --?>	[redacted]	SQLAgent - TSQL JobStep (Job 0x24DEEF4FFC29D441BE932B23300438C3 : Step 1)	00 00:02:05.300	87	[redacted]erver	(28ms)PAGEIOLATCH_SH:Inventory:1 (*)
SQLCLUSTER2		master	Microsoft SQL Server	00 00:02:01.550	77	LinkedServer	
SQLCLUSTER2		master	Microsoft SQL Server	00 00:01:56.993	76	LinkedServer	
SQLCLUSTER2	<?query -- begin tran --?>	[redacted]Services	SQLCLUST[redacted]rs [redacted]MS_Tables_2	00 00:00:04.826	70	[redacted]erver	
SQLCLUSTER2	<?query -- begin tran --?>	[redacted]Services	SQLCLUSTE[redacted] [redacted]e	00 00:00:04.313	71	[redacted]erver	

Problems solved

- More than 100 counters
- Trend analysis and prediction
- Ability to drill down from high level problem to root cause
- Meet growing business needs

Business value

- Operations throughput

Business value

- Operations throughput
- Decision making

Business value

- Operations throughput
- Decision making
- Cost of failure

Business value

- Operations throughput
- Decision making
- Cost of failure
- Capacity planning (volume, type of hardware)

Business value

- Operations throughput
- Decision making
- Cost of failure
- Capacity planning (volume, type of hardware)
- DB monitoring is usually undervalued

Business value

- Operations throughput
- Decision making
- Cost of failure
- Capacity planning (volume, type of hardware)
- DB monitoring is usually undervalued
- Examples: DELL EMC, VIVA, Spar, Stock market Brokers

Real life cases

- Examples: DELL EMC, VIVA, Spar, Stock market Brokers
- DELL EMC From Nagios and basic alerts to comprehensive DB monitoring
- VIVA From basic alerts
- Spar From no monitoring
- Stock market brokers From SCOM
- WallMart from no monitoring

Nagios case detailed (DELL EMC)

- Before
- Only basic monitoring
- After
- Comprehensive monitoring

Benefits of the solution

- Open Source agents

Benefits of the solution

- Open Source agents
- Low to no impact on performance

Benefits of the solution

- Open Source agents
- Low to no impact on performance
- Secure

Benefits of the solution

- Open Source agents
- Low to no impact on performance
- Secure
- Scalable

Benefits of the solution

- Open Source agents
- Low to no impact on performance
- Secure
- Scalable
- Free

Benefits of the solution

- Open Source agents
- Low to no impact on performance
- Secure
- Scalable
- Free
- Customizable

Benefits of the solution

- Open Source agents
- Low to no impact on performance
- Secure
- Scalable
- Free
- Customizable
- Authors

Questions

Contacts

- www.dimenko.com
- roman.dimenko@gmail.com
- LinkedIn: https://ru.linkedin.com/in/romandimenko

- Early adopters are welcome

Thank you!