

Visual Studio Team Test

Christian Gerdes

Software QA Specialist

LIGHTS IN LINE AB

Agenda

- Vad krävs av prestandaverktyg?
- Visual Studio 2008 Team Test (VSTT)
- Fiddler
- Team Foundation Server
- Demo: VSTT in action

Vad krävs av prestandaverktyg?

- Testautomatisering (skriptning)
- Distribuerad lastgenerering
- Scenarion för att simulera belastning
- Insamling av statistik
- Analys

Testautomatisering

- Inspelning av flöden / användningsfall
- Kodning av användningsfall
- Parametersättning och korrelering
- Verifiering / test
- Versionshantering

Distribuerad lastgenerering

- Kunna skala upp till tusentals VU
- Distribuera lasten över flera agenter / kanoner
- Kontroll över miljön

Scenarion för att simulera belastning

- Rampup / konstant belastning
- Modellera olika profiler
- Simulera betänketider och webbläsare (cookies, cache, egenskaper)
- Simulera olika nätverksegenskaper
- Schedulera och automatisera körningar

Insamling av statistik

- Mäta individuella testresultat
- Lagra / logga fel, spara snapshots
- Mäta resursförbrukning på hårdvara
- Mäta prestandavärden från applikationen under test (AUT)

Analys

- Sammanställa alla mätdata för en körning
- Kunna skapa egna grafer (korrelera)
- Exportera grafer och mätdata
- Kunna jämföra resultat mellan körningar
- Kontroll över körningar

VSTT Testautomatisering

- Inspelning av flöden / användningsfall
 - Webtest: Internet Explorer och Fiddler (proxy)
- Kodning av användningsfall
 - Webtest eller Unittest: C#.NET eller VB.NET
- Parametersättning och korrelering
 - Automatisk korrelering av alla Hidden fields
 - Manuell korrelering (Text, DOM, RegEx, Egna)
 - Parametersättning (Text, Excel, DB)
 - Egna plugins

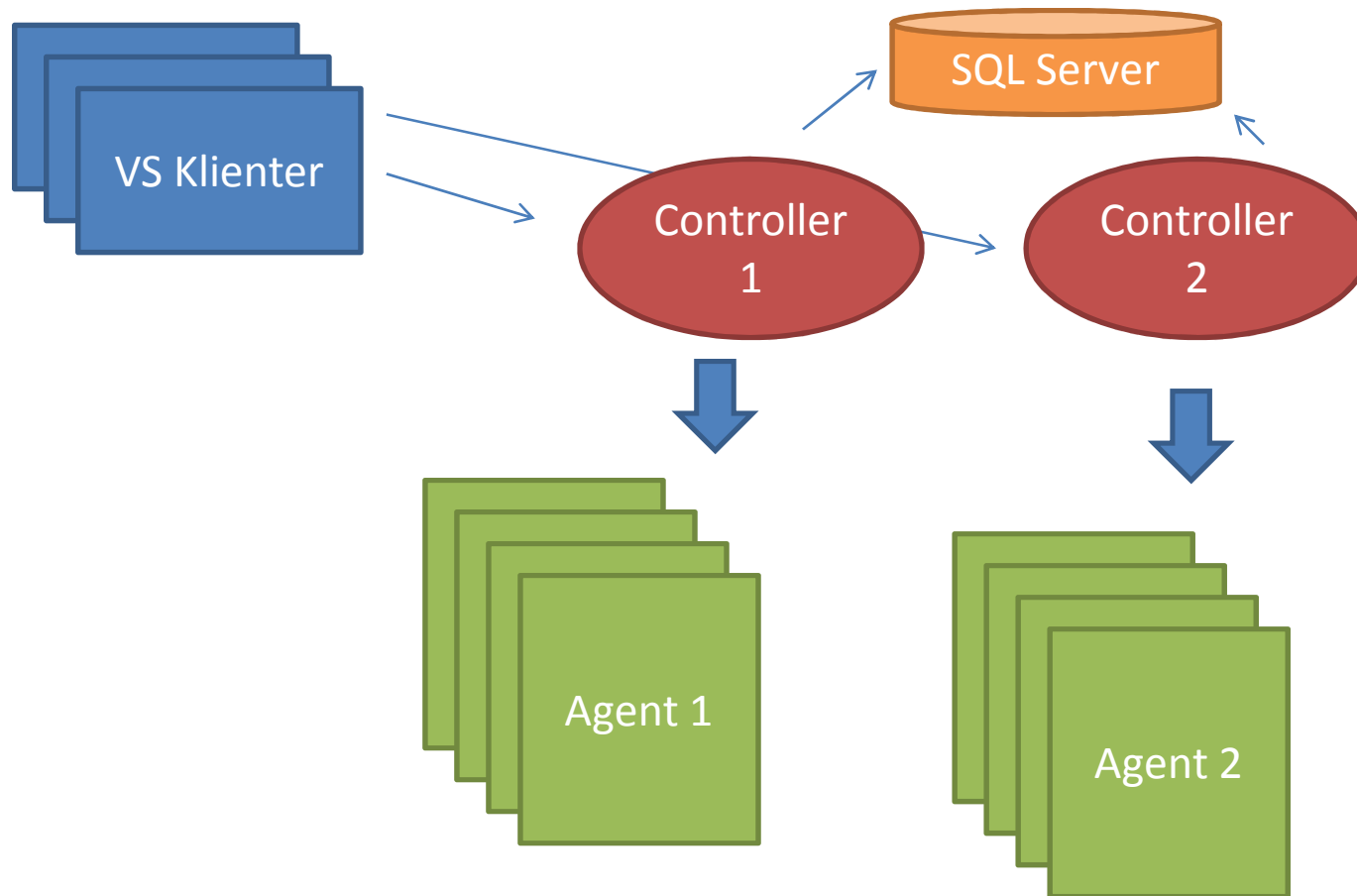
VSTT Testautomatisering forts.

- Verifiering / test
 - Mycket bra debug möjligheter
 - Enkel och avancerad variant
 - Inbyggd mätning av resurser, delsvarstider och datamängder, underlättar verifiering
 - Dubbelkolla med Fiddler / Network Monitor
- Versionshantering
 - Integration med TFS, tredje parts integrationer med CVS / SVN

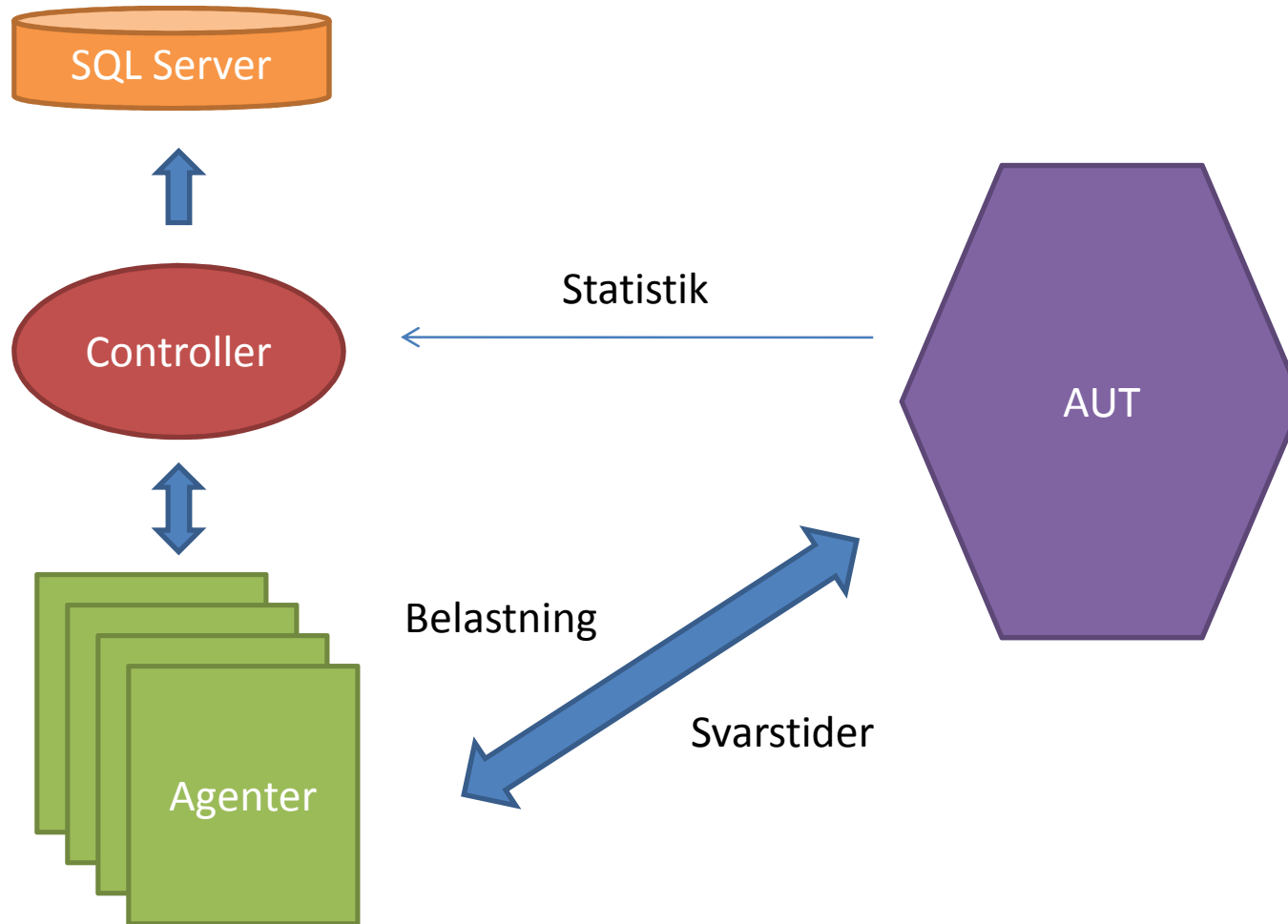
VSTT Distribuerad lastgenerering

- Kunna skala upp till tusentals VU
 - Beror mycket på hur tunga testerna är
- Distribuera lasten över flera agenter / kanoner
 - Controller/Agent arkitektur
 - En agent tillhör endast 1 controller
- Kontroll över miljön
 - Från början byggt för Team
 - Flera VS klienter kan ansluta till samma controller

VSTT Arkitektur



VSTT Arkitektur (scenario)



VSTT Insamling av statistik

- Mäta individuella testresultat
 - Svarstider för individuella sidor och requests (summary, percentile eller detailed), nedbrytning (connect, first byte, datamängd)
 - Egna namngivna transaktioner (response, elapsed)
 - Genomströmning mäts i tester/s och requests/s
 - Förhållandet mellan cachade/ocachade mäts
 - Mätning av tid spenderad i kod/plugins
 - Thresholds / SLA mäts i procent och antal

VSTT Insamling av statistik forts.

- Lagra / logga fel, spara snapshots
 - Snapshots av felaktiga sidor sparas automatiskt
 - Komplet stacktrace i unittests
 - Passed/failed tester/requests (totalt samt per sekund)
 - Fel kategoriseras och mäts individuellt
 - Korreleringsfel, valideringsfel, http statuskod fel, timeouter, SLA/Thresholds på resurser, Exceptions (kod)

VSTT Insamling av statistik forts.

- Mäta resursförbrukning på hårdvara
 - Idag endast Windows Performance Counters
 - Möjlighet att bygga egna datainsamlare
- Mäta prestandavärden från applikationen under test (AUT)
 - Samma som ovan
 - Integration med Team Developer för profilering
 - SQL profiler integration
 - Tredje parts integrationer t.ex. DynaTrace

Analys

- Sammanställa alla mätdata för en körning
 - Alla mätresultat lagras i controllerns SQL DB
 - Mätresultaten kan ses och analyseras i realtid i VS
- Kunna skapa egna grafer (korrelera)
 - Nya grafer kan skapas under som efter körningen
- Exportera grafer och mätdata
 - Grafer kan exporteras till Word/Excel direkt från VS
- Kunna jämföra resultat mellan körningar
 - Flera körningar kan öppnas parallellt i VS för jämförelse
- Kontroll över körningar
 - Mycket bra översikt, vem som körde, resultatet samt anteckningar kan göras synliga i översikten på respektive controller
 - Körresultat kan exporteras till disk
 - Körresultat och analyser kan publiceras till TFS

VSTT Styrkor

- Baserat på utvecklare IDE, kraftfullt för egen kodning och debuggning, versionshantering
- Från början byggt för Team, integration med TFS
- Kraftfulla plugin möjligheter
- Öppna API och Open Source Community (www.codeplex.com)

Frågor?

...sen

DEMO