UNIVERSITY of GLASGOW

Lecture 2: A Scalable Control Plane Architecture

Prof. Joe Sventek
joe@dcs.gla.ac.uk

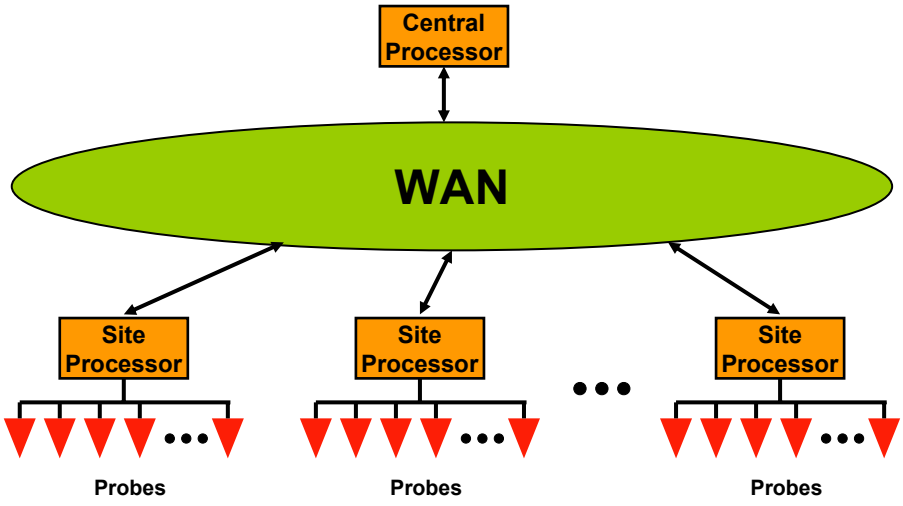
UNIVERSITY of GLASGOW

Complexity, anyone?

- As the industry continues its inexorable rush to multi-service networks, the following complexities result
  - The number of active elements in the network are exploding to handle the exponentially-increasing traffic
  - The transmission rates of individual interfaces is quadrupling every 3-5 years
  - We are coercing many high-speed streams of data onto a single fibre
  - The mix of traffic types requires the introduction of QoS mechanisms and policies into the network
  - Manual management of these complex beasts is no longer sufficient to meet customer guarantees
- One important question to answer: how can we guarantee first time, every time success in the configuration of such complex management systems?

UNIVERSITY of GLASGOW

acceSS7 Architecture



UNIVERSITY of GLASGOW

SS7 Example - Relative Sizing

- Links per site – up to 2000
- Links per probe – maximum of 8, average of 4

  $\Rightarrow$ 250-500 probes per site
- Sites – up to 2000
- Applications @ probe processor - up to 200
- Applications @ site processor - up to 20
- Worst case situation
  - $>10^6$ processor components
  - $> 2 \times 10^8$ software components

Traditional Configuration Approaches

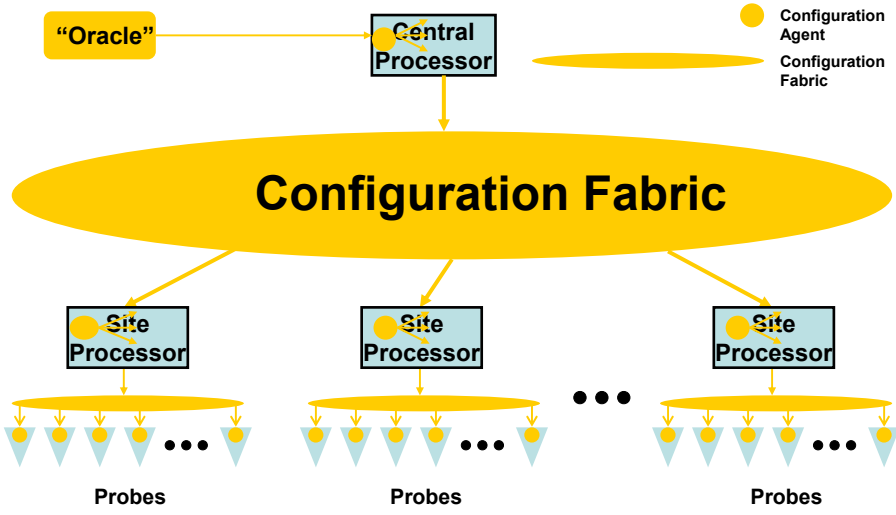
- Traditional configuration solutions to these kinds of management applications have the following design:
  - Application components are configuration naïve – all configuration is performed outside of the components
  - The dependencies between components are captured in a database
  - The configuration/reconfiguration act itself is specified in scripts, driven by the dependency information, from a central location; the scripts are traditionally written to activate the application components in a lock step fashion, consistent with the dependency graph
  - Where parallelism is legal, it is achieved through complexity in the scripts
- In short, such configuration solutions melt down when confronted with an application the size of acceSS7.

Asymptotic Configuration

- Applications like Call Detail Record generation in acceSS7 are of a class that we refer to as background monitoring and management. These applications are required to be 24x7, while supporting *in situ* reconfiguration.
- These types of applications can deliver value to the customer long before all components have achieved the correct configuration.
- Our approach to achieving better scalability for this class of applications is based upon three premises:
  - Make the components responsible for configuring themselves correctly
  - Use multi-cast communication wherever possible to enable communications costs that scale substantially better than linearly in the number of components
  - Use replicated databases and randomization of access to avoid congestion when accessing the configuration database

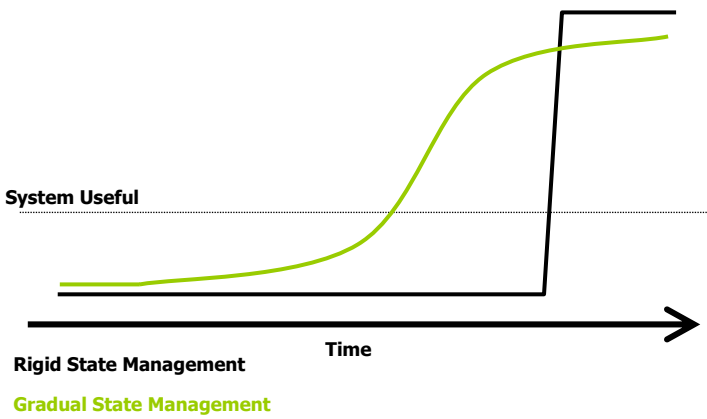
UNIVERSITY of GLASGOW

Configuration Architecture

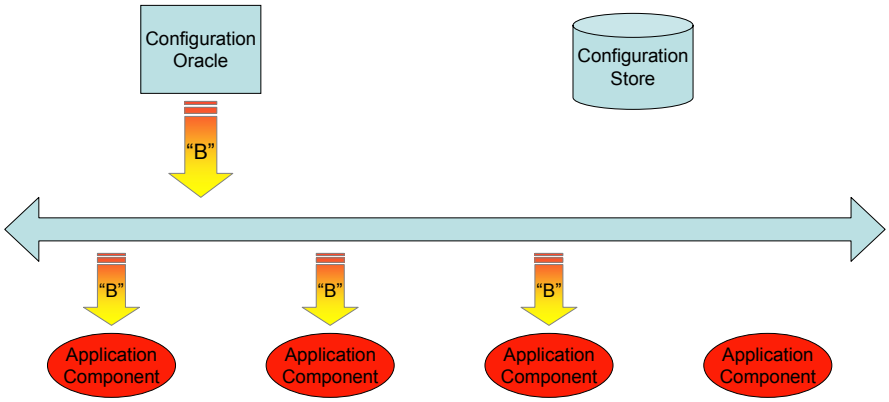


UNIVERSITY of GLASGOW

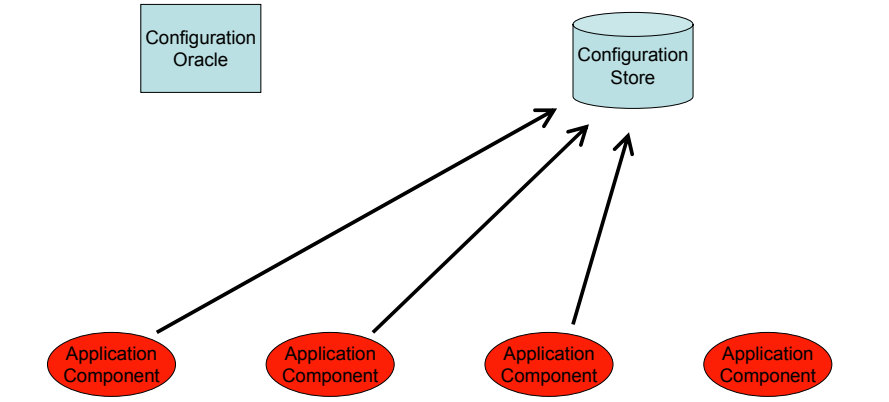
Rigid vs. Gradual State Management



Gradual Configuration

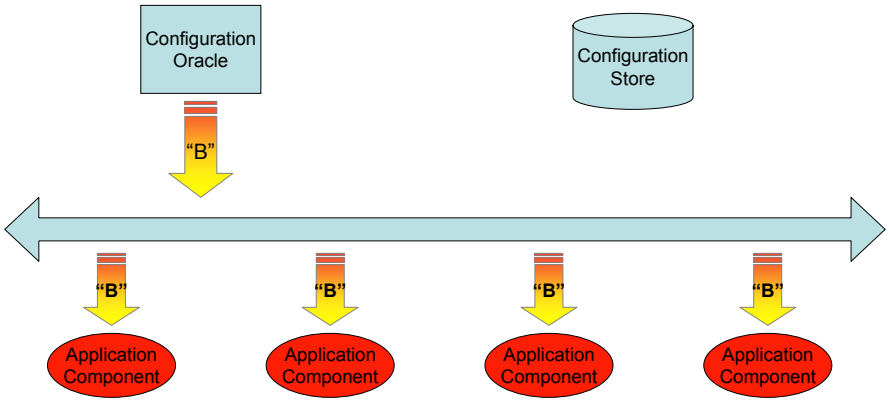


Gradual Configuration



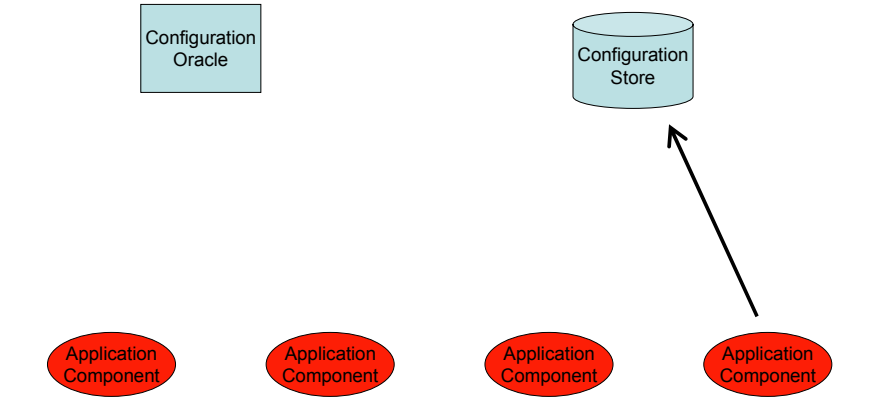
UNIVERSITY of GLASGOW

Gradual Configuration



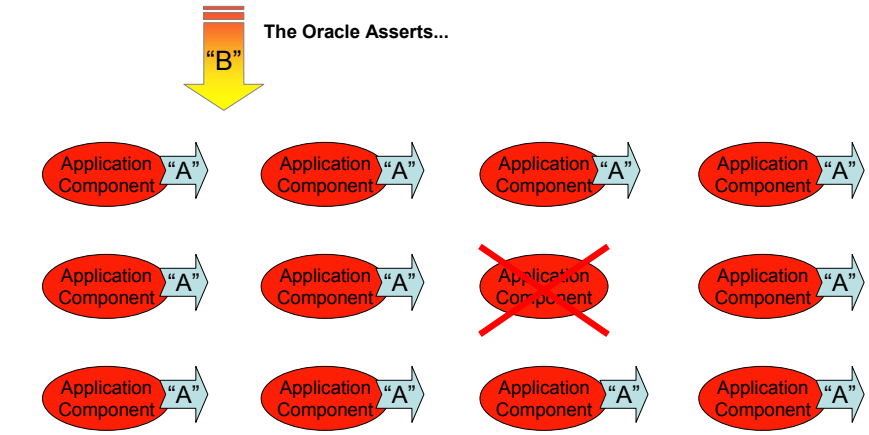
UNIVERSITY of GLASGOW

Gradual Configuration



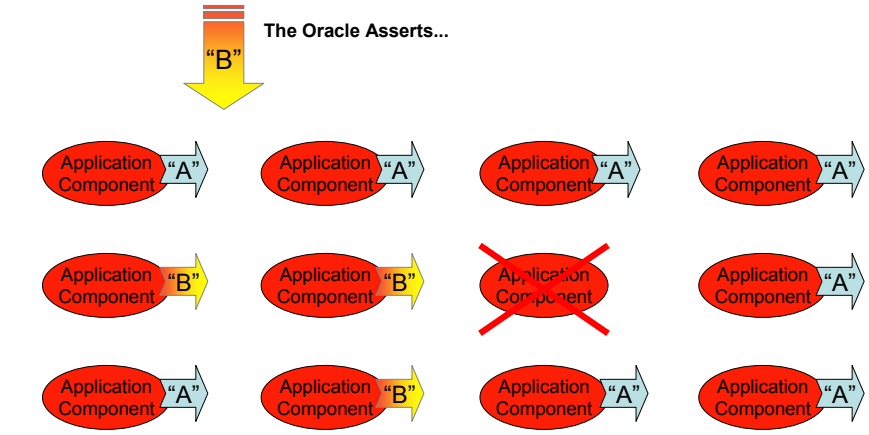
UNIVERSITY of GLASGOW

Approaching Equilibrium



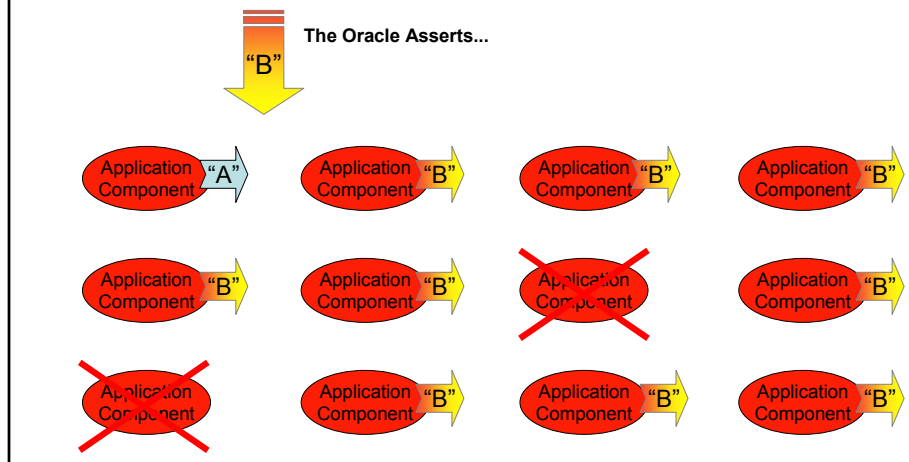
UNIVERSITY of GLASGOW

Approaching Equilibrium



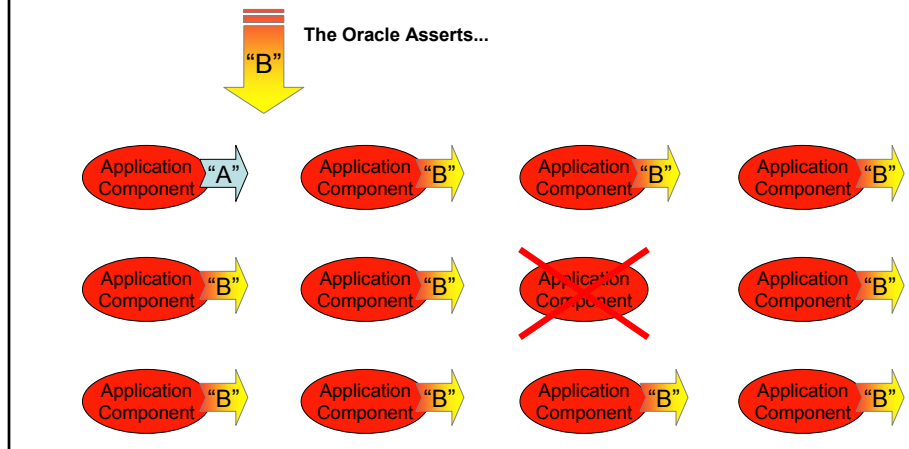
UNIVERSITY of GLASGOW

Approaching Equilibrium



UNIVERSITY of GLASGOW

Approaching Equilibrium



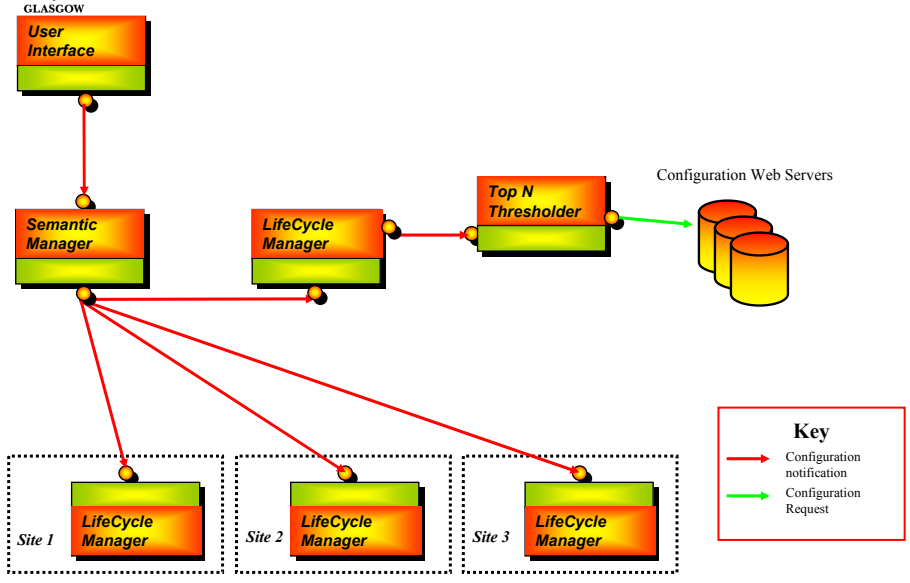
UNIVERSITY of GLASGOW

Critical Concepts...

- Applicable if the application can be useful when a subset of components are executing correctly
- Enables use of (but does not require) unreliable multicast communications
- Enables the system to become usable faster and be more robust
- Permits systems to be self-healing
- Requires some design time consideration and configuration-awareness on the part of application components
- Implies fine-grained & continuous monitoring
- Scales to extremely large systems

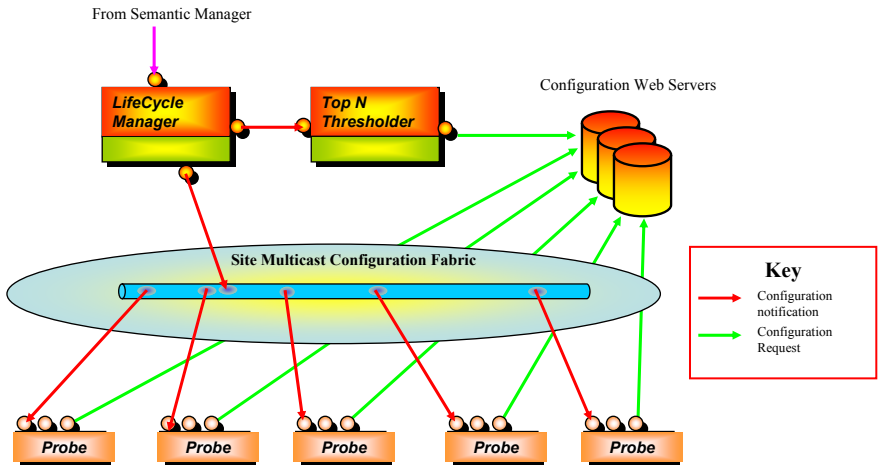
UNIVERSITY of GLASGOW

Central Site Configuration



UNIVERSITY of GLASGOW

Probe/Site Level Configuration



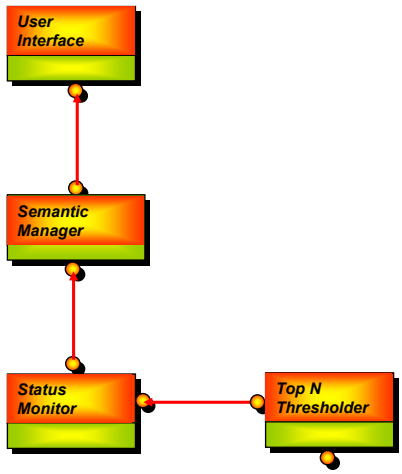
UNIVERSITY of GLASGOW

Status Monitoring

- The status of a Component can be derived in 2 ways :
  - Actively : you can ask/have the component tell you its state.
  - Passively : you can infer the state of the component by observing its behaviour.
- The Prototype uses both approaches.
  - Thresholders report their state directly to 'Status Monitor' Components.
  - The Status Monitor components 'snoop' on Probe traffic.

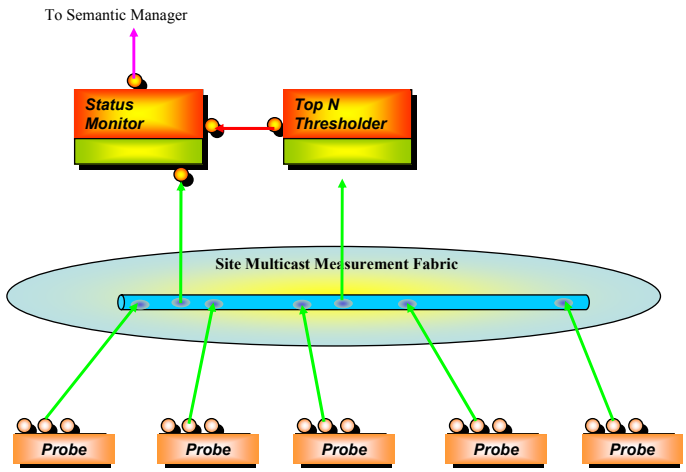
UNIVERSITY of GLASGOW

Central Site Status Monitoring



UNIVERSITY of GLASGOW

Probe/Site Level Status Monitoring



Implementation Specifics

- All code implemented in Java (configuration traffic is not the bottleneck for these systems)
- "Oracle" asserts correct state every 15 seconds
- Unreliable multi-cast used for communicating configuration updates to probes; reliable point-to-point used between central site and site processors
- Reliable multi-cast used to communicate tagged data from probes to site processor; reliable point-to-point used between sites and central
- Full replication of configuration databases; each component picks a random copy from which to retrieve its data

Lessons learned

- >90% of the probe components moved to the correct configuration after the first multi-cast of a new configuration
- Communicating two states, current and future+time, permits almost instantaneous change to a new state, subject to the synchronization of the processor clocks
- The prototype system was stable when confronted with constant configuration changes
- The prototype took no appreciable difference in the time necessary to converge as the number of probes grew; since the prototype used pt-to-pt communications between central and sites, its scalability is much more dependent upon the number of sites. For a truly large number of sites, a form of WAN multi-cast must be used