Automated Multi-Tier System Design for Service Availability

John Janakiraman
Jose Renato Santos
Yoshio Turner

Hewlett Packard Laboratories

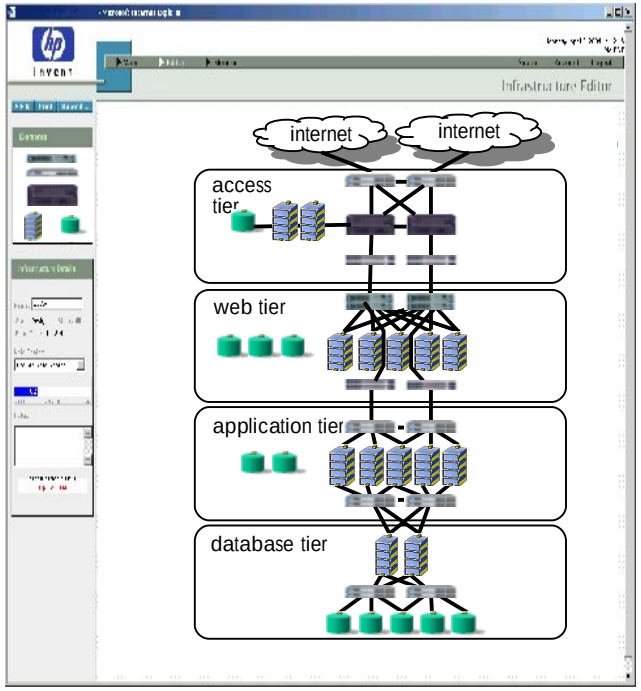
Motivation:
New Enterprise/Internet Computing Model

- **Utility Computing**
  HP (Adaptive Enterprise), IBM (Autonomic Computing), SUN (N1)
  - Shared resources allocated to services on demand
    - Virtual resources hide details of physical environment
  - Self-managing system
    - Service life-cycle (creation, change, deletion)
    - Adaptation (load changes, component faults, etc.)
  - High level service requirements specification
    - E.g. desired performance and availability instead of detailed design
- **Automated System Design/Configuration**
  - Required for utility computing
  - Our focus: design for service availability

HP Utility Data Center (UDC)

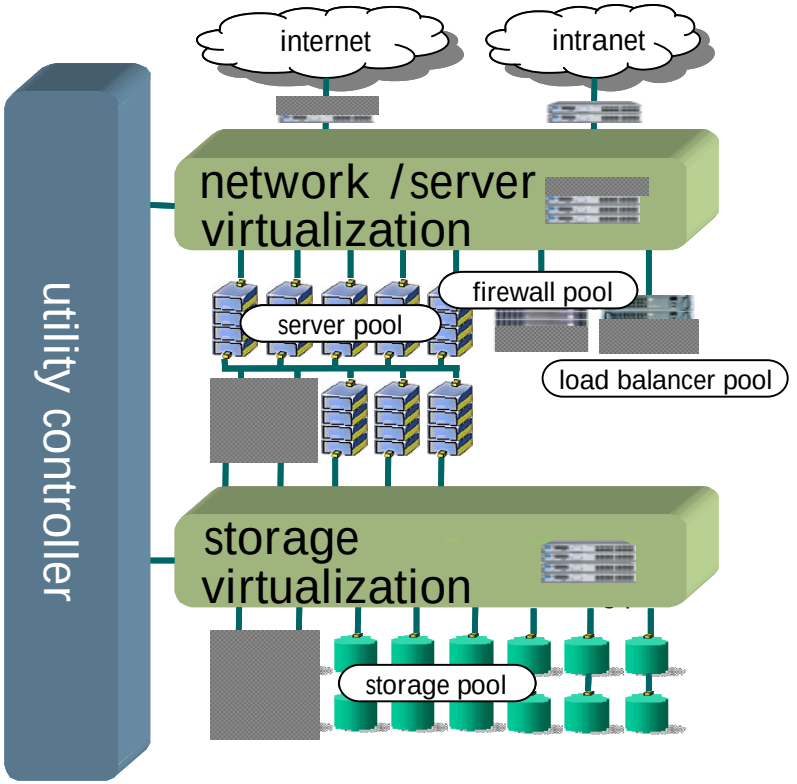
- HP hardware and software solution that enables provision of computing resources to applications on demand.

logical resources specification



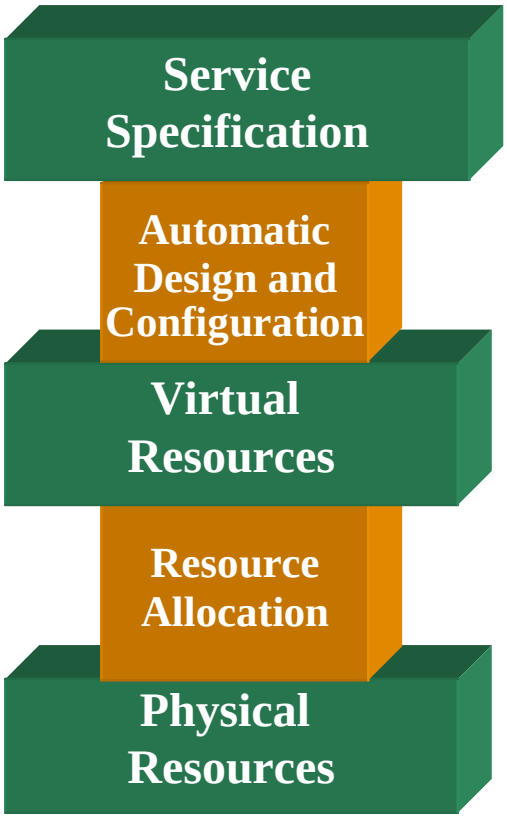
deployment

allocated physical resources



Utility Computing Environment

- **Higher level service specification**
  - Functional specification
  - Performance requirement
  - Availability requirement
- **Design/Configuration automation**
  - System determines the required computing resources for service
    - type of resources
    - amount of resources
    - topology
    - application and OS configurations
  - Dynamically redesign and reconfigure due to changes (load, faults, etc…)



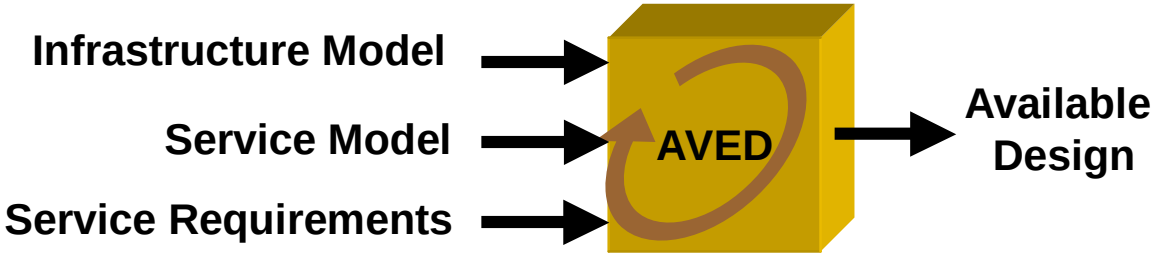
Automated System Design for Availability

Goal:

Automatically explore the space of infrastructure designs and availability mechanisms and select a design that meets availability requirements with minimum cost.

- **Numerous availability mechanisms to select/configure (cost/benefit tradeoffs)**
  - host failover, NIC failover, standby/active spare, database checkpointing, application state checkpointing (on peer, on file, on database), data replication, software rejuvenation, etc…
  - Different mechanisms and operating points have different cost, performance overhead and availability characteristics

AVED: Proof of Concept Tool



- **Initial prototype for stand-alone environment**
- **Current Scope:**

  Application type: Multi-tier services (e.g. 3-tier e-commerce)
  Availability requirement: Maximum Service Downtime per year
  Design space (limited set):
  - Choice of server hw and sw components (no network, no storage)
  - Number of servers
  - State of servers (active or spare)
  - Repair strategies for component failures
- **Key challenges:**
  - How to model relevant properties of service and compute infrastructure
  - How to reason about alternative designs

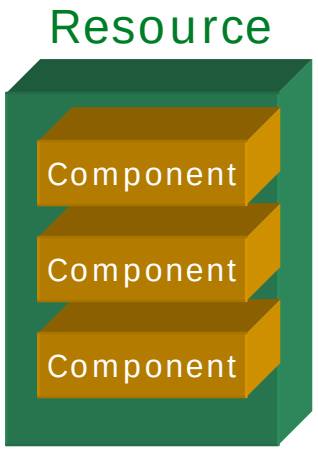
Modeling Approach

Infrastructure Model

- **Component types:** Basic elements that can fail
  - Cost model
  - A set of failure modes
    - A set of repair options for each failure mode (cost/benefit)

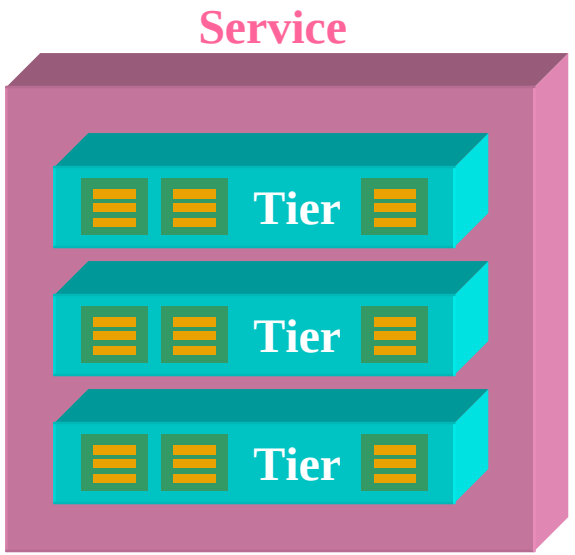
  Example: lp2000r(2-way x86 server) , rp8400 (8-way PA-RISC server), Linux, WebLogic, etc.
- **Resource types:** Unit of provisioning
  - Set of components

  Example: App server = lp2000r+Linux+WebLogic



Service Model

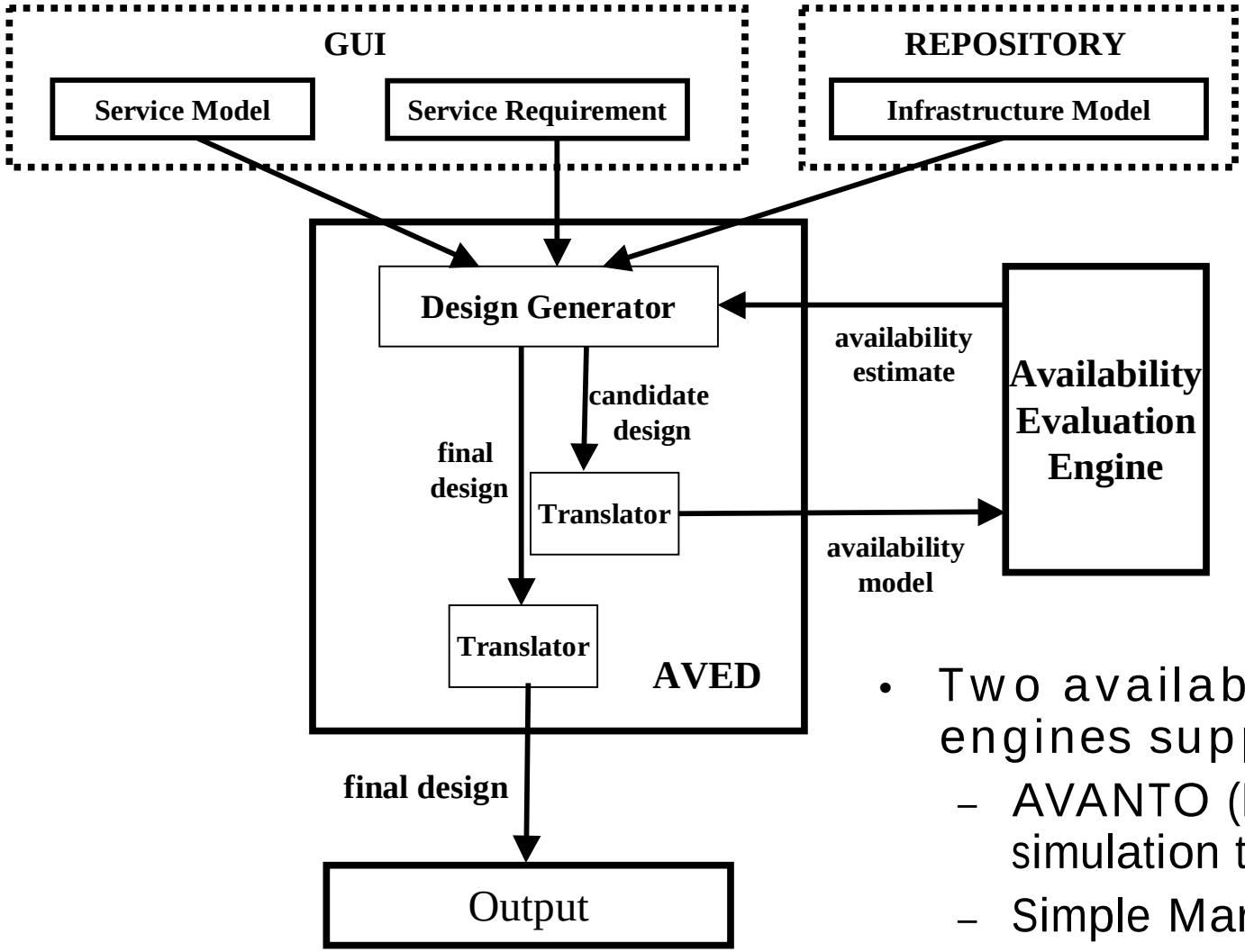
- Set of tiers
  - Set of valid resource alternatives for each tier
  - Performance model for each tier



Service Requirements

- Expected load (peak)
- Maximum Annual Downtime
  - System is assumed down when resources cannot sustain expected peak load

AVED architecture

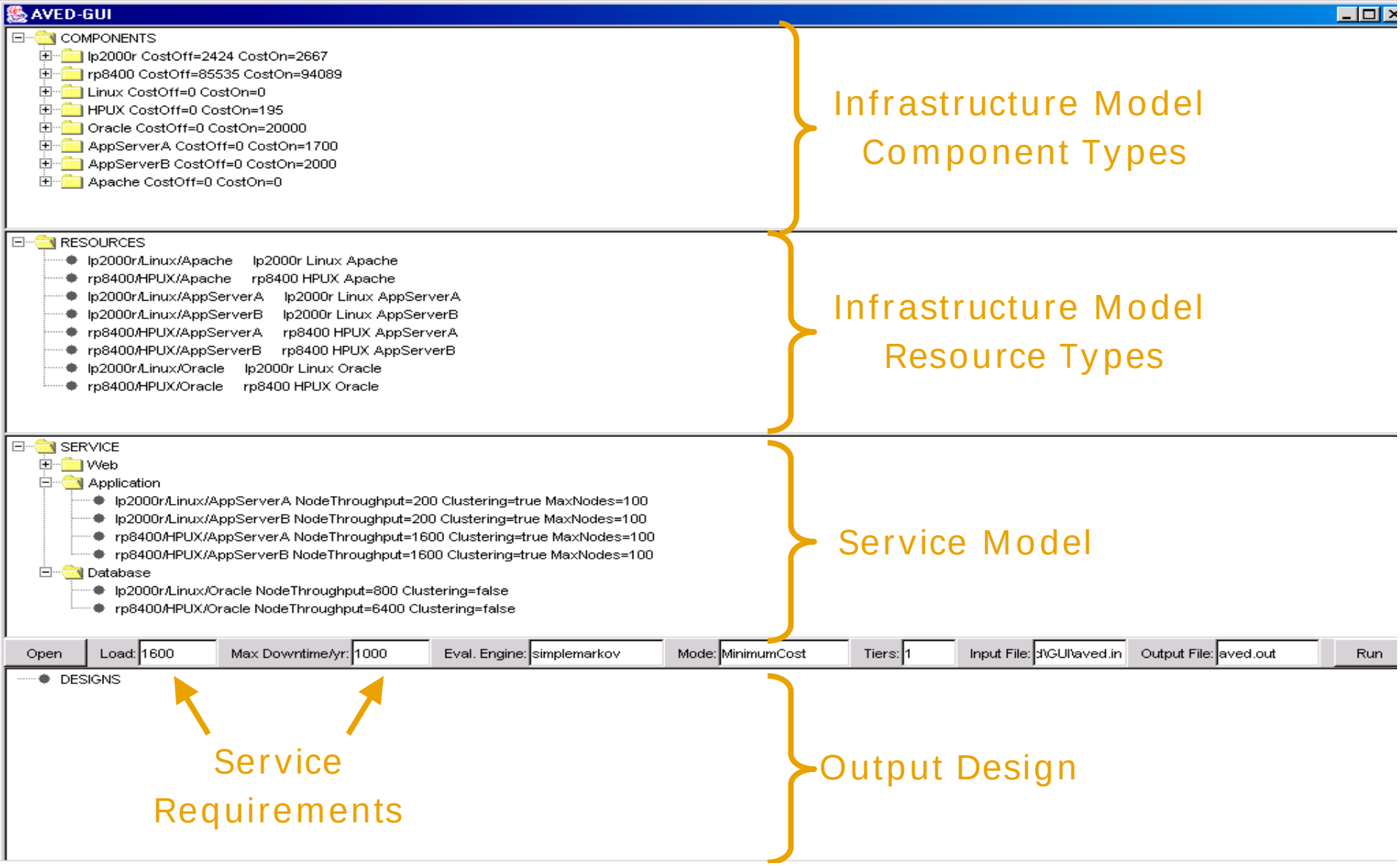


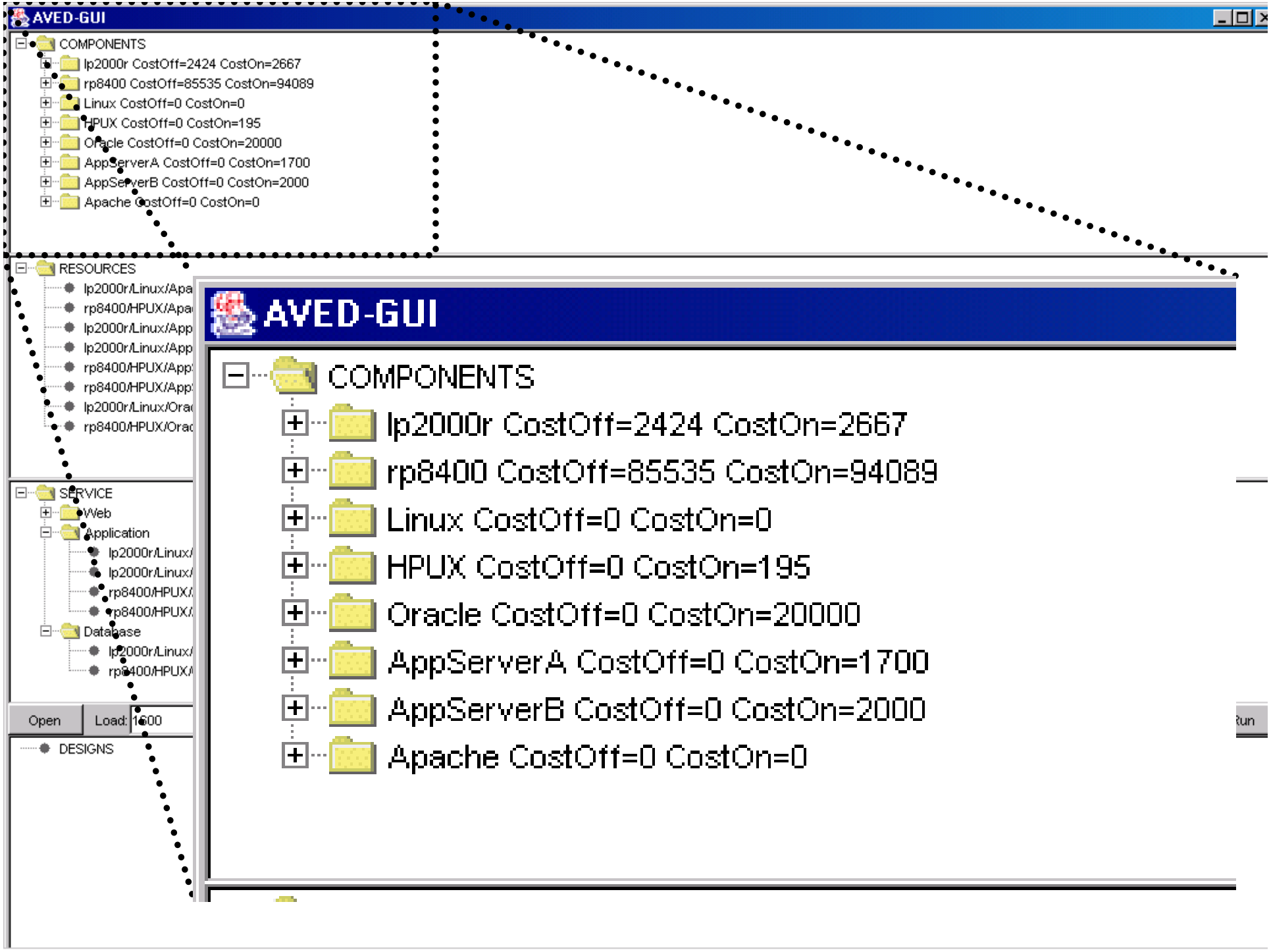
- Two availability evaluation engines supported:
  - AVANTO (HP availability simulation tool)
  - Simple Markov model

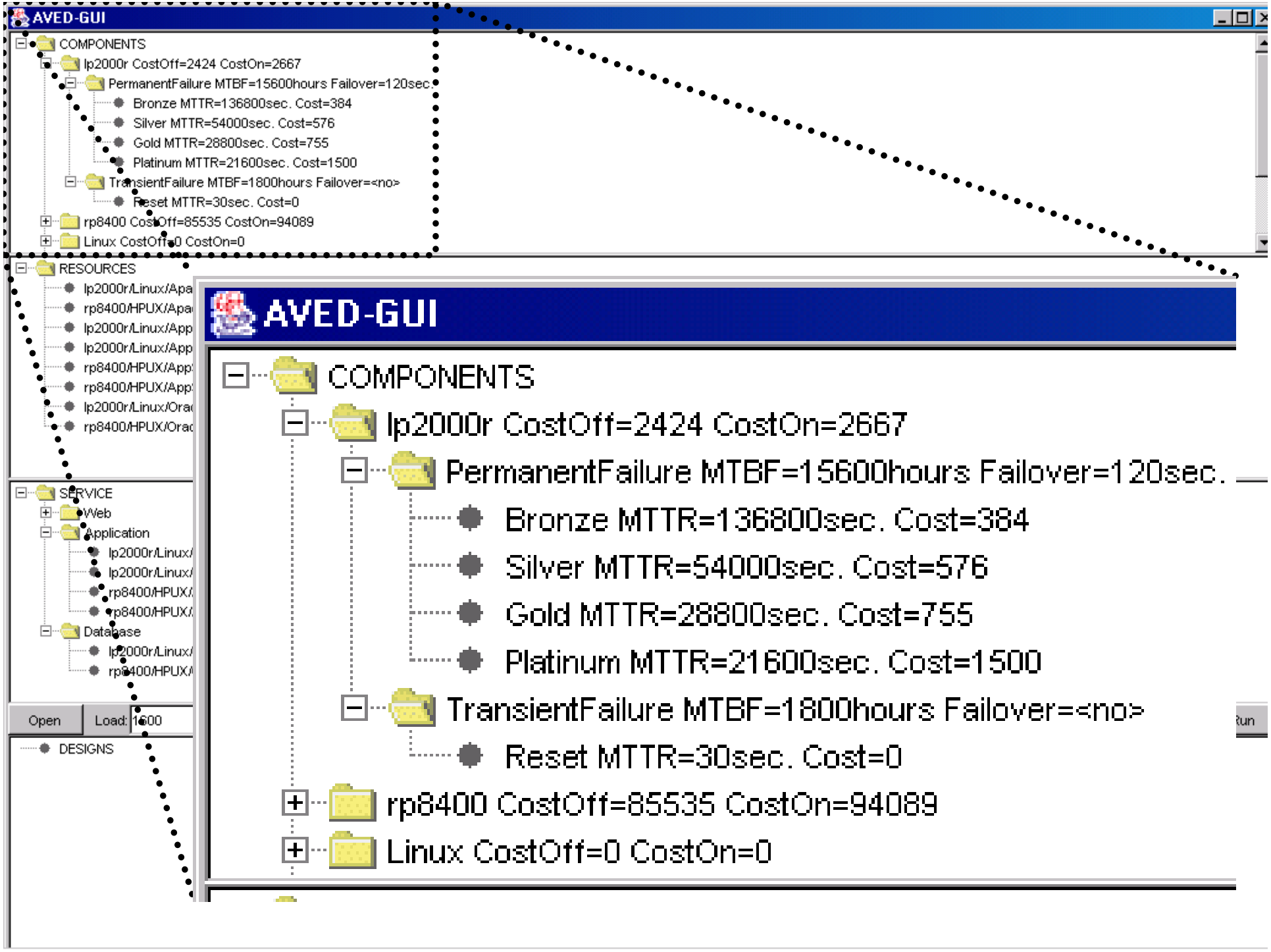
Illustrating the use of AVED

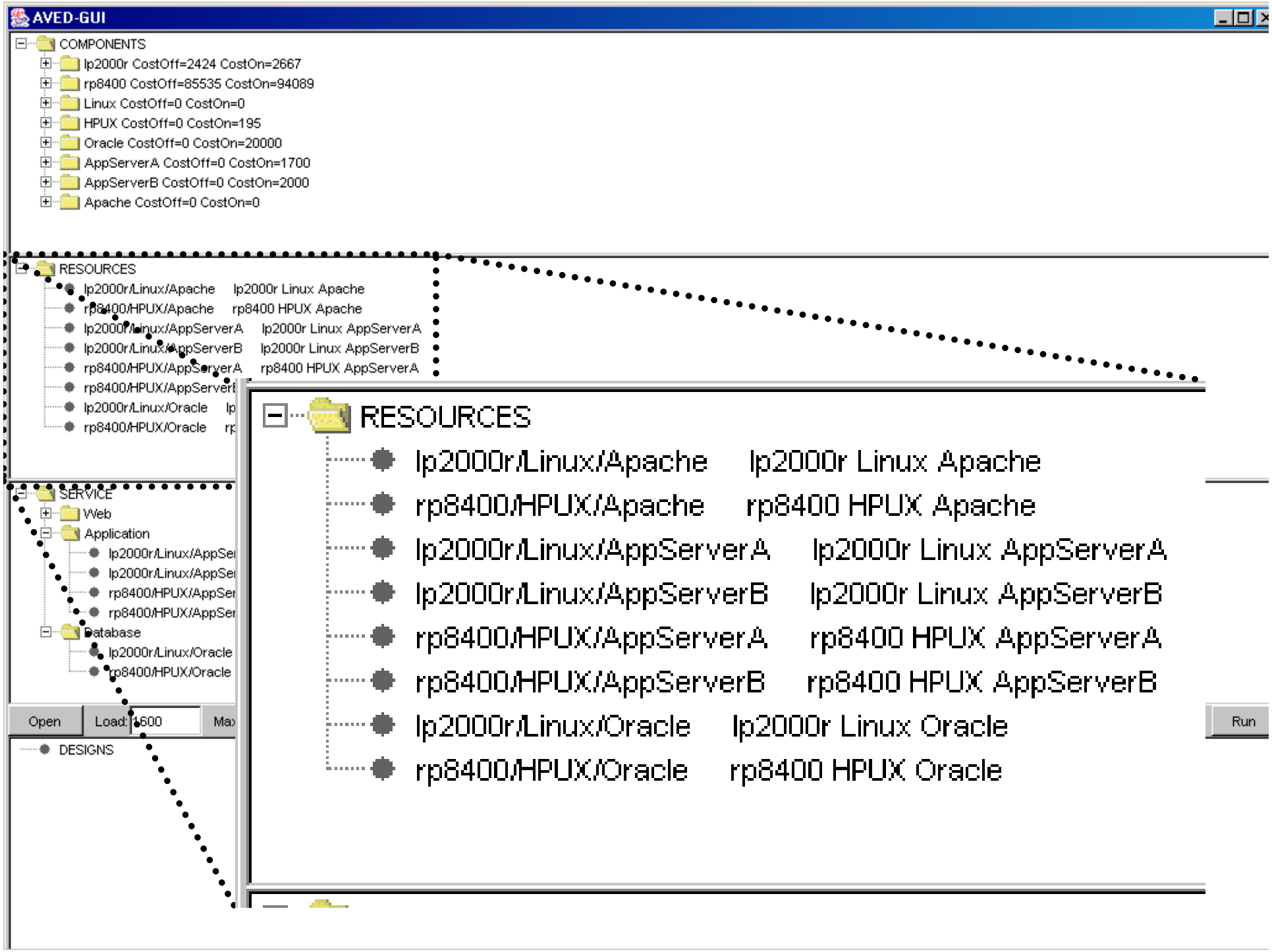
- Application tier scenario
- Published component failure rates (hardware)
- Costs based on vendor published prices
- Reasonable assumptions for unavailable information
  - E.g. SW failure rates

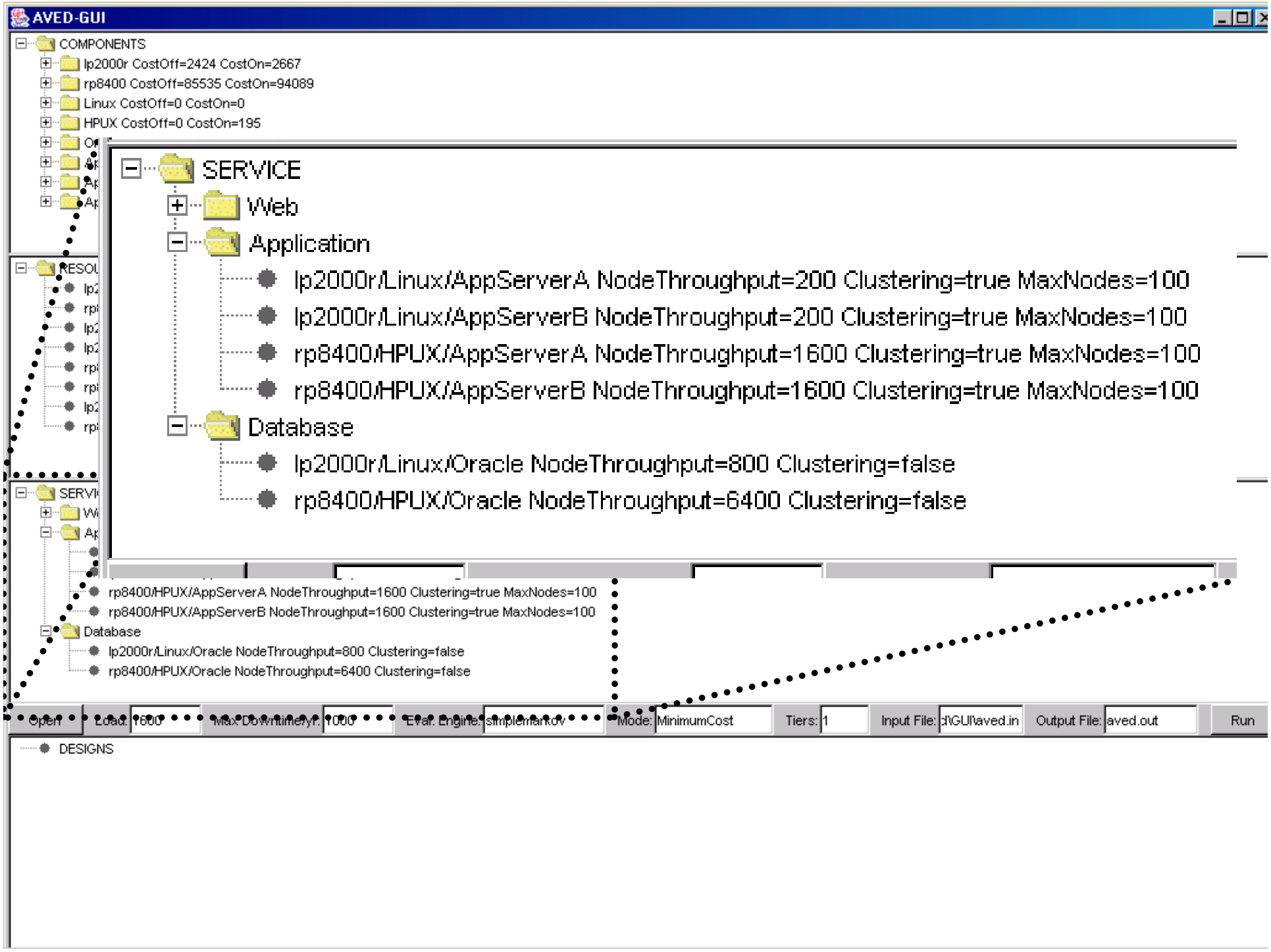
AVED GUI

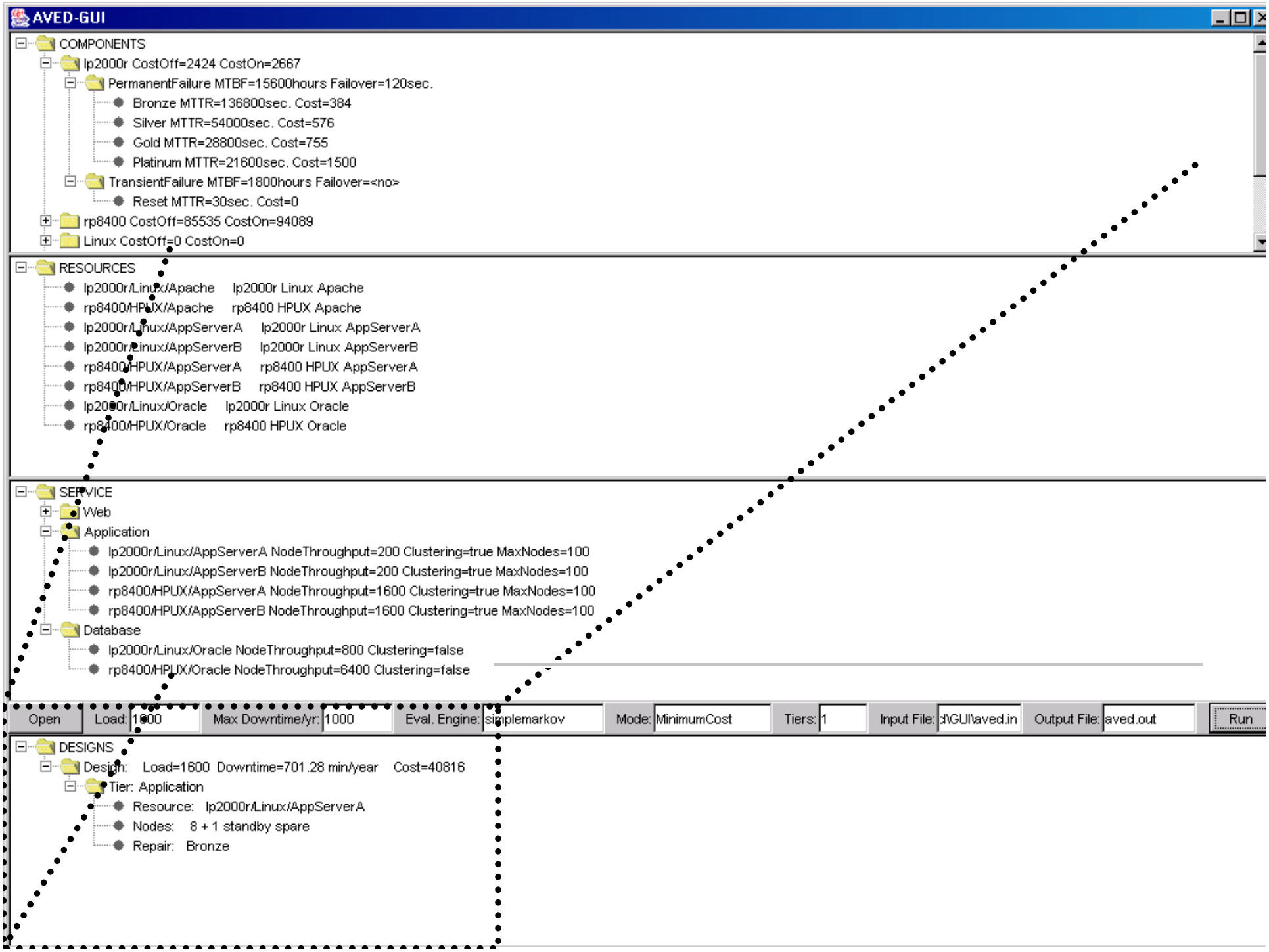

AVED-GUI
COMPONENTS
lp2000r CostOff=2424 CostOn=2667
rp8400 CostOff=85535 CostOn=94089
Linux CostOff=0 CostOn=0
HPUX CostOff=0 CostOn=195
Oracle CostOff=0 CostOn=20000
AppServerA CostOff=0 CostOn=1700
AppServerB CostOff=0 CostOn=2000
Apache CostOff=0 CostOn=0
Infrastructure Model Component Types
RESOURCES
lp2000r/Linux/Apache lp2000r Linux Apache
rp8400/HPUX/Apache rp8400 HPUX Apache
lp2000r/Linux/AppServerA lp2000r Linux AppServerA
lp2000r/Linux/AppServerB lp2000r Linux AppServerB
rp8400/HPUX/AppServerA rp8400 HPUX AppServerA
rp8400/HPUX/AppServerB rp8400 HPUX AppServerB
lp2000r/Linux/Oracle lp2000r Linux Oracle
rp8400/HPUX/Oracle rp8400 HPUX Oracle
Infrastructure Model Resource Types
SERVICE
Web
Application
lp2000r/Linux/AppServerA NodeThroughput=200 Clustering=true MaxNodes=100
lp2000r/Linux/AppServerB NodeThroughput=200 Clustering=true MaxNodes=100
rp8400/HPUX/AppServerA NodeThroughput=1600 Clustering=true MaxNodes=100
rp8400/HPUX/AppServerB NodeThroughput=1600 Clustering=true MaxNodes=100
Database
lp2000r/Linux/Oracle NodeThroughput=800 Clustering=false
rp8400/HPUX/Oracle NodeThroughput=6400 Clustering=false
Service Model
Open
Load: 1600
Max Downtime/yr: 1000
Eval. Engine: simplemarkov
Mode: MinimumCost
Tiers: 1
Input File: d\GUI\aved.in
Output File: aved.out
Run
DESIGNS
Service Requirements
Output Design


AVED-GUI
COMPONENTS
Ip2000r CostOff=2424 CostOn=2667
rp8400 CostOff=85535 CostOn=94089
Linux CostOff=0 CostOn=0
HPUX CostOff=0 CostOn=195
Oracle CostOff=0 CostOn=20000
AppServerA CostOff=0 CostOn=1700
AppServerB CostOff=0 CostOn=2000
Apache CostOff=0 CostOn=0
RESOURCES
SERVICE
Web
Application
Database
Open
Load: 1600
DESIGNS


AVED-GUI
COMPONENTS
lp2000r CostOff=2424 CostOn=2667
PermanentFailure MTBF=15600hours Failover=120sec.
Bronze MTTR=136800sec. Cost=384
Silver MTTR=54000sec. Cost=576
Gold MTTR=28800sec. Cost=755
Platinum MTTR=21600sec. Cost=1500
TransientFailure MTBF=1800hours Failover=<no>
Reset MTTR=30sec. Cost=0
rp8400 CostOff=85535 CostOn=94089
Linux CostOff=0 CostOn=0
RESOURCES
SERVICE
Web
Application
Database
Open
Load:
1600
DESIGNS

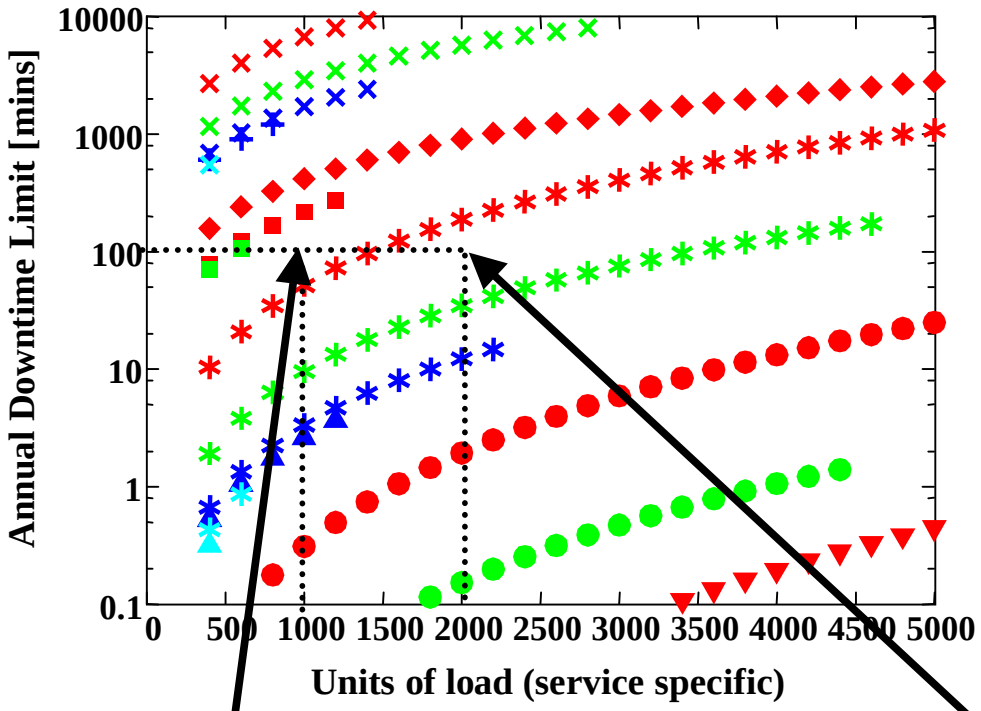

AVED-GUI
COMPONENTS
lp2000r CostOff=2424 CostOn=2667
rp8400 CostOff=85535 CostOn=94089
Linux CostOff=0 CostOn=0
HPUX CostOff=0 CostOn=195
Oracle CostOff=0 CostOn=20000
AppServerA CostOff=0 CostOn=1700
AppServerB CostOff=0 CostOn=2000
Apache CostOff=0 CostOn=0
RESOURCES
lp2000r/Linux/Apache lp2000r Linux Apache
rp8400/HPUX/Apache rp8400 HPUX Apache
lp2000r/Linux/AppServerA lp2000r Linux AppServerA
lp2000r/Linux/AppServerB lp2000r Linux AppServerB
rp8400/HPUX/AppServerA rp8400 HPUX AppServerA
rp8400/HPUX/AppServerB
lp2000r/Linux/Oracle
rp8400/HPUX/Oracle
SERVICE
Web
Application
Database
Open
Load: 1600
DESIGNS
Run


AVED-GUI
COMPONENTS
lp2000r CostOff=2424 CostOn=2667
rp8400 CostOff=85535 CostOn=94089
Linux CostOff=0 CostOn=0
HPUX CostOff=0 CostOn=195
SERVICE
Web
Application
lp2000r/Linux/AppServerA NodeThroughput=200 Clustering=true MaxNodes=100
lp2000r/Linux/AppServerB NodeThroughput=200 Clustering=true MaxNodes=100
rp8400/HPUX/AppServerA NodeThroughput=1600 Clustering=true MaxNodes=100
rp8400/HPUX/AppServerB NodeThroughput=1600 Clustering=true MaxNodes=100
Database
lp2000r/Linux/Oracle NodeThroughput=800 Clustering=false
rp8400/HPUX/Oracle NodeThroughput=6400 Clustering=false
Open
Load:
1600
Max Downtime/yr:
1000
Eval. Engine:
simplemarkov
Mode:
MinimumCost
Tiers:
1
Input File:
d:\GUI\aved.in
Output File:
aved.out
Run
DESIGNS


AVED-GUI
COMPONENTS
lp2000r CostOff=2424 CostOn=2667
PermanentFailure MTBF=15600hours Failover=120sec.
Bronze MTTR=136800sec. Cost=384
Silver MTTR=54000sec. Cost=576
Gold MTTR=28800sec. Cost=755
Platinum MTTR=21600sec. Cost=1500
TransientFailure MTBF=1800hours Failover=<no>
Reset MTTR=30sec. Cost=0
rp8400 CostOff=85535 CostOn=94089
Linux CostOff=0 CostOn=0
RESOURCES
lp2000r/Linux/Apache lp2000r Linux Apache
rp8400/HPUX/Apache rp8400 HPUX Apache
lp2000r/Linux/AppServerA lp2000r Linux AppServerA
lp2000r/Linux/AppServerB lp2000r Linux AppServerB
rp8400/HPUX/AppServerA rp8400 HPUX AppServerA
rp8400/HPUX/AppServerB rp8400 HPUX AppServerB
lp2000r/Linux/Oracle lp2000r Linux Oracle
rp8400/HPUX/Oracle rp8400 HPUX Oracle
SERVICE
Web
Application
lp2000r/Linux/AppServerA NodeThroughput=200 Clustering=true MaxNodes=100
lp2000r/Linux/AppServerB NodeThroughput=200 Clustering=true MaxNodes=100
rp8400/HPUX/AppServerA NodeThroughput=1600 Clustering=true MaxNodes=100
rp8400/HPUX/AppServerB NodeThroughput=1600 Clustering=true MaxNodes=100
Database
lp2000r/Linux/Oracle NodeThroughput=800 Clustering=false
rp8400/HPUX/Oracle NodeThroughput=6400 Clustering=false
Open
Load: 1600
Max Downtime/yr: 1000
Eval. Engine: simplemarkov
Mode: MinimumCost
Tiers: 1
Input File: c:\GUI\aved.in
Output File: aved.out
Run
DESIGNS
Design: Load=1600 Downtime=701.28 min/year Cost=40816
Tier: Application
Resource: lp2000r/Linux/AppServerA
Nodes: 8 + 1 standby spare
Repair: Bronze

Example of AVED use – Results

- Identifies optimal solution for a range of service requirements: load and maximum annual downtime

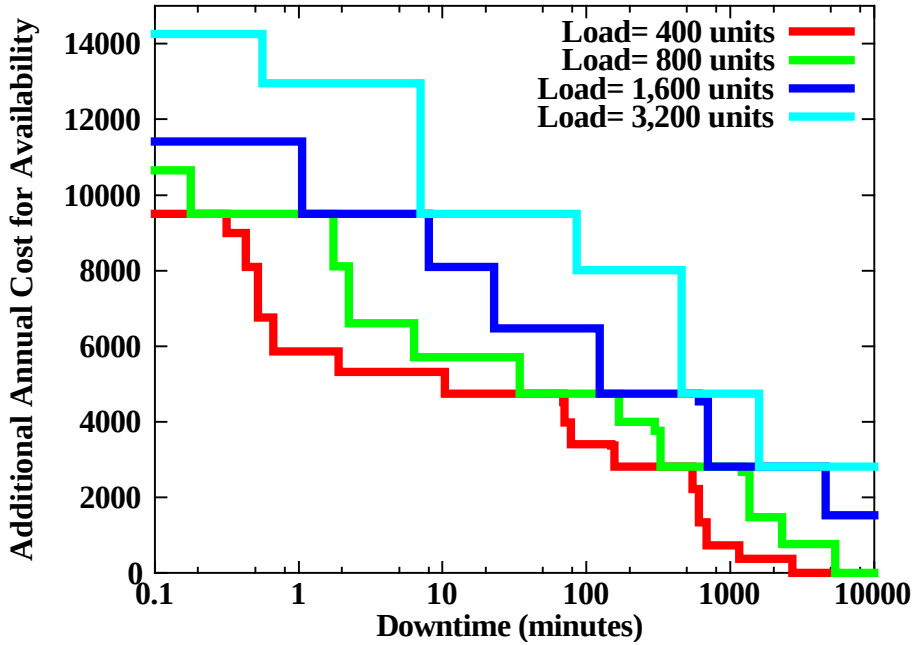


1 - lp2000r/linux/AS-A, bronze, no spare
2 - lp2000r/linux/AS-A, silver, no spare
3 - lp2000r/linux/AS-A, gold, no spare
4 - lp2000r/linux/AS-B, gold, no spare
5 - lp2000r/linux/AS-A, platinum, no spare
6 - lp2000r/linux/AS-A, bronze, 1 cold spare
7 - lp2000r/linux/AS-B, bronze, 1 cold spare
8 - lp2000r/linux/AS-B, silver, 1 cold spare
9 - lp2000r/linux/AS-A, bronze, 1 active spare
10 - lp2000r/linux/AS-A, silver, 1 active spare
11 - lp2000r/linux/AS-A, gold, 1 active spare
12 - lp2000r/linux/AS-B, gold, 1 active spare
13 - lp2000r/linux/AS-A, platinum, 1 active spare
14 - lp2000r/linux/AS-B, platinum, 1 active spare
15 - lp2000r/linux/AS-A, bronze, 2 active spare
16 - lp2000r/linux/AS-A, silver, 2 active spare
17 - lp2000r/linux/AS-A, bronze, 3 active spare

Design 9 optimal for requirement (1000 load units, 100minutes max downtime)

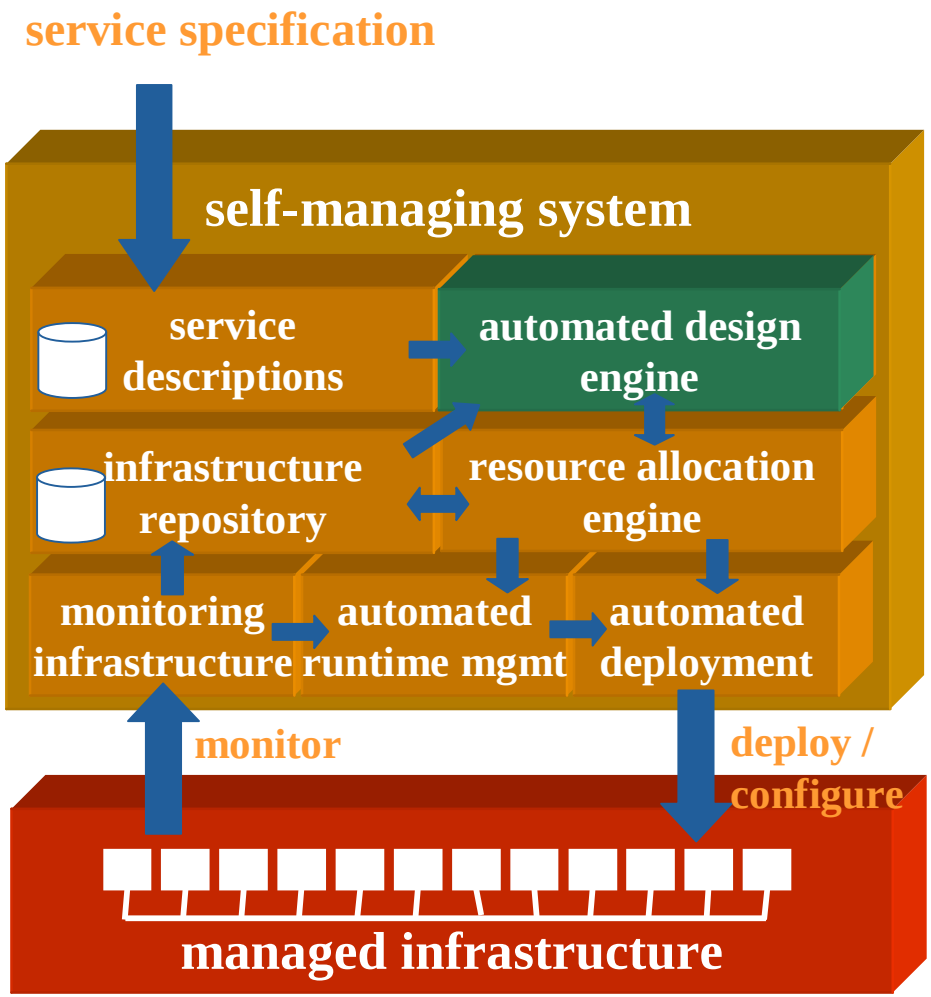
Design 10 optimal for requirement (2000 load units, 100minutes max downtime)

Example of AVED use – Results



- **Availability/Cost tradeoff**
  - e.g., relaxing downtime requirement from 1.5 min/yr to 2.5 min/yr in a design for 800 load units reduces cost from $9500 to $6600

Automated Design in Self-Managing System



Integrating AVED w/ automatic deployment

- Automatic OS deployment using network boot (PXE)
- Automatic application deployment/configuration
- Automatic configuration of failure detection, repair and failover mechanisms
- Extensions for closed-loop adaptive operation
  - Integrate monitoring mechanisms (load, performance, recovery time, failure rates)
  - Adapt AVED for incremental design/configuration changes

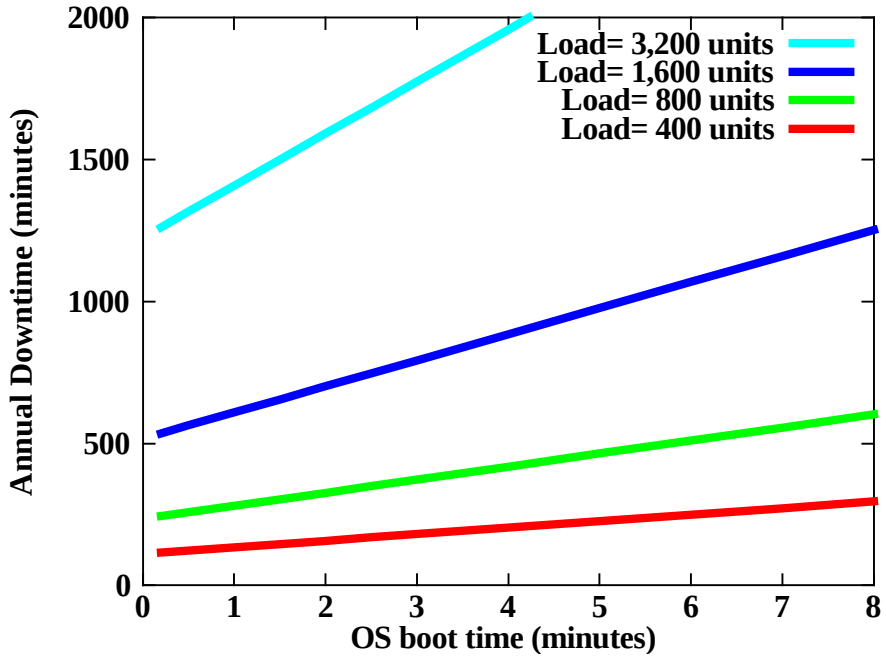
Conclusion

- **AVED:** proof of concept tool that automates the design and configuration of availability mechanisms
  - Important building block in utility computing environment
  - Also useful in stand-alone mode

- **Future work**
  - Extended cost models
  - More complex performance models
    - Experimental models based on monitoring
  - Factor business cost of downtime
  - Alternative availability metrics
    - Outage duration, degraded performance levels, etc.
  - Support for storage and network resources
  - Failure dependencies/correlations


hp
invent

Example of AVED use – Results



- downtime is sensitive to repair parameters, e.g., the OS boot time.
  - sensitivity analysis necessary for parameters with unreliable values
  - can select designs with lower sensitivity to such parameters

parameters used in example

Inputs: Component Behaviors & Costs

Component	Cost Cold	Cost Active	Failure Type	MTBF	Repair Option	MTTR	Repair Cost	Failover Time
Machine A lp2000r	$2400	$2640	Transient	75 days	Reset	30 sec	$0	N/A
			Permanent	650 days	Bronze	38 hrs	$380/node	2 min
					Silver	15 hrs	$580/node	
					Gold	8 hrs	$750/node	
					Platinum	6 hrs	$1500/node	
Machine B M-B	$85000	$93500	Transient	150 days	Reset	30 sec	$0	N/A
			Permanent	1300 days	Bronze	38 hrs	$380/node	2 min
					Silver	15 hrs	$580/node	
					Gold	8 hrs	$750/node	
					Platinum	6 hrs	$1500/node	
Linux	$0	$0	Crash	60 days	Reboot	2 min	$0	N/A
Unix	$200	$0	Crash	365 days	Reboot	4 min	$0	N/A
App Server AS-A	$1700	$0	Crash	30 days	Reboot	2 min	$0	N/A
App Server AS-B	$2000	$0	Crash	90 days	Reboot	30 sec	$0	N/A

Inputs: Service Characteristics

Resource	Performance Model		Cluster?
	Node capacity	Max Nodes	
lp2000r/Linux/AS-A	200 units	25	true
M-B/Unix/AS-A	1600 units	25	true
lp2000r/Linux/AS-B	200 units	25	true
M-B/Unix/AS-B	1600 units	25	true

- design choices: type of machine/OS/application server resource, number of extra machines, state of extra machines (cold or active), repair option

Relative cost