Automated Multi-Tier System Design for Service Availability

G. (John) Janakiraman, Jose Renato Santos, Yoshio Turner
Internet Services and Storage Laboratory
HP Laboratories Palo Alto

May 22, 2003

Author Information

– G. (John) Janakiraman and Jose Renato Santos

Working in:

HP Labs Internet Systems & Storage Group

in Palo Alto, California

This group is currently designing a new computing model (Planetary computing) to handle the large infrastructure needs required to support the growing Internet.

Author Information – Yoshio Turner

Education: Attended CS program at UCLA

Working in:

HP Labs Datacenter Architecture Team and the Internet Systems & Storage Lab

Current Research Interests

- Data center server architecture
- High service availability for services that are hosted on datacenter clusters
- Interconnection networks – routing, congestion, and Quality of Service

Grand Vision

- HP, IBM, and Sun Microsystems are attempting to deliver computing as a utility.
- "The idea is that a user who wishes to deploy an Internet or Enterprise service would issue a request to a computing utility, which in response would automatically allocate and configure appropriate resources from pools of compute, storage, and networking resources to create a secure, virtualized computing environment that realizes the service." (Janakiraman 1)

Acronym Review

- **UDC**
  - Hewlett Packard's Utility Data Center
  - Commercial solution
  - Recognizes current state of achieving utility vision
- **AVED**
  - Proof-of-concept prototype
  - Provides design automation for highly available system infrastructures
  - Improves self-management functionality of UDC

Current Practice and Impact of AVED

- Currently, human system developers manually generate design alternatives for the system.
- Then, they use availability modeling tools to evaluate the availability of their designs.

Modeling Tool Usage
-Predicts Service Downtime
-Predicts the cost of the downtime based on the business mission
-Provides an upper limit on the system availability that can be achieved.
-Provides cost-benefit tradeoffs of the different design options

solutions to evaluate.

Components of a Self Managing System

High Level properties
Service uptime – Throughp
Tolerable Degradation – Du
Bounds and cost of degrada

Automated Design Engine

- Selects and configures infrastructure components and availability mechanisms
- Ensures that high level requirements are met.
- Automatically explores design alternatives and embeds environment models
- Determines the design alternative that satisfies the requirements at the minimal cost.

Infrastructure Repository

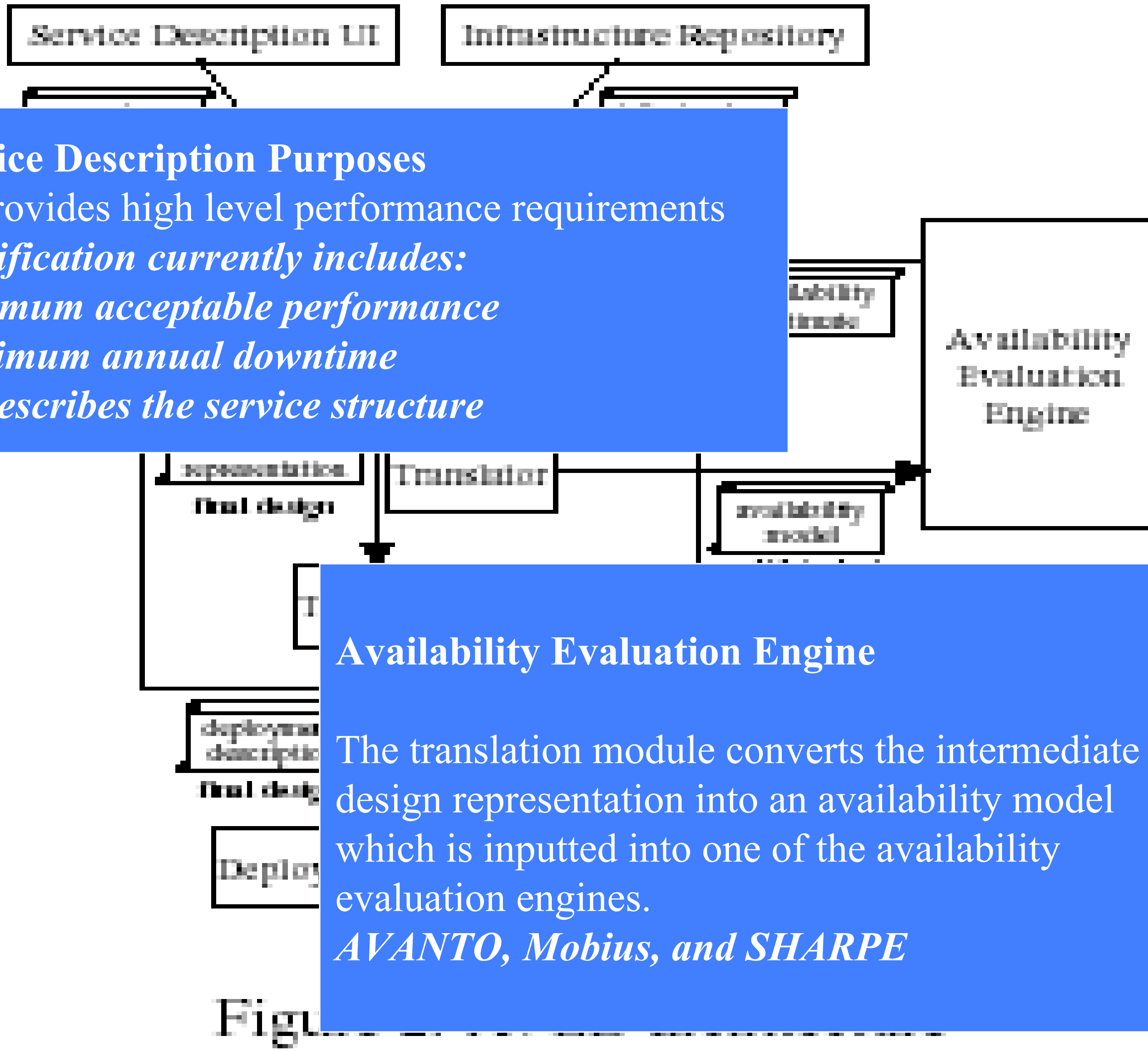
Provides empirical information a
infrastructure element failure p
transient failure rates, restart time
permanent failure rates, and scop
Availability attributes

Automated Deployment and Automated Runtime Management

- Deploy and configure the availability mechanisms.
- Ensure that backup resources are available for upgrades and ensure that they are scheduled when business impact is minimal.

Monitoring Infrastructure

- Monitor failure characteristics
- Monitor recovery characteristic dependencies, failure rates, repai and recovery times.



Service Description Purposes

1. Provides high level performance requirements

Specification currently includes:

Minimum acceptable performance

Maximum annual downtime

2. ***Describes the service structure***

Availability Evaluation Engine

The translation module converts the intermediate design representation into an availability model which is inputted into one of the availability evaluation engines.

AVANTO, Mobius, and SHARPE

Three Tier Structure Example

Tier name – tiers that are to comprise the service implementation

```
tier_name=Web-Tier
  tier_resource=ResourceA  cluster=True singleload=100  nmax=25
  tier_resource=ResourceB  cluster=True singleload=300  nmax=25
```

nmax – Maximum number of active resources

ice-specific load units.

```
  tier_resource=ResourceC  cluster=True singleload=200  nmax=25
  tier_resource=ResourceD  cluster=True singleload=600  nmax=25

tier_name = Database-Tier
  tier_resource=ResourceE  cluster=False singleload=500  nmax=1
  tier_resource=ResourceF  cluster=False singleload=1500 nmax=1
```

Resource Type Definition

```
resource=ResourceA   MachineB Linux WebServerX
resource=ResourceB   MachineA Unix  WebServerX
```

Hardware

Operating System

Application

Hardware Component Specification

```
component=MachineA cost_cold=1000 cost_active=1100
  perm_mtbf=650d failover_used=true failover_duration=2m
     repair_mttr=15h repair_cost=580
     repair_mttr=6h  repair_cost=1500
  tran_mtbf=75d failover_used=false
     repair_mttr=30s  repair_cost=0
```

Preemptive Maintenance

Preemptive maintenance can have an impact on availability.

Software rejuvenation can improve MTBF for a failure mode.

Future plans to define PM option parameters.

Design Space Search

Algorithm 1 Single tier design space search for a single resource

```
dt = Evaluate(MinCostDesign_0)
if (dt ≤ downtime) then
    return (MinCostDesign_0)
end if
dt = Evaluate(MinDowntimeDesign_MaxSpares)
if (dt > downtime) then
    return NO SOLUTION
end if
MinSpares = MaxSpares
Current = MinDowntimeDesign_MaxSpares
for i=0 to MaxSpares-1 do
    dt = Evaluate(MinDowntimeDesign_i)
    if (dt ≤ downtime) then
        MinSpares = i
        exit for loop
    end if
end for
Current = InvalidDesign
for i=MinSpares to MaxSpares do
    if (Cost(MinCostDesign_i) > Cost(Current)) then
        return Current
    end if
    for all possible Designs with i spares do
        if (Cost(Design) < Cost(Current)) then
            dt = Evaluate(Design)
            if (dt ≤ Downtime) then
                Current = Design
            end if
        end if
    end for
end for
return NO SOLUTION
```

Designing the Application Tier of an Internet Service for High Availability:

Components Failure Behavior and Costs

Component	Cost Cold	Cost Active	Failures	MTBF	Repair Option	MTTR	Repair Cost	Failover Time
Machine A (M-A)	$2,400.00	$2,640.00	Transient	75 days	Reset	30 sec.	$0.00	No
			Permanent	650 days	Serv. Contract			2 min.
					1.Bronze	38 hours	$380.00/machine	
					2.Silver	15 hours	$580.00/machine	
					3. Gold	8 hours	$750.00/machine	
					4. Platinum	6 hours	$1500.00/machine	
Machine B (M-B)	$85,000.00	$93,500.00	Transient	150 days	Reset	60 sec.	$0.00	No
			Permanent	1300 days	Serv. Contract			2 min.
					1.Bronze	38 hours	$10,000.00/machine	
					2.Silver	15 hours	$12,500.00/machine	
					3. Gold	8 hours	$16,000.00/machine	
					4. Platinum	6 hours	$25,000/machine	
Linux	$0.00	$0.00	Crash	60 days	Reboot	2 min.	$0.00	No
UNIX	$0.00	$200.00	Crash	365 days	Reboot	4 min.	$0.00	No
Applic. Server A (AS-A)	$0.00	$1,700.00	Crash	30 days	Restart	2 min.	$0.00	No
Applic. Server B (AS-B)	$0.00	$2,000.00	Crash	90 days	Restart	30 sec.	$0.00	No

Table 1: Example input parameters: Components failure behavior and costs

Service Characteristics

Resource	Performance Model		cluster flag
	singleload	nmax	
M-A/linux/AS-A	200 *load units*	25 nodes	true
M-B/unix/AS-A	1600 *load units*	25 nodes	true
M-A/linux/AS-B	200 *load units*	25 nodes	true
M-B/unix/AS-B	1600 *load units*	25 nodes	true

Table 2: Example input parameters: Service characteristics

Optimal Solution

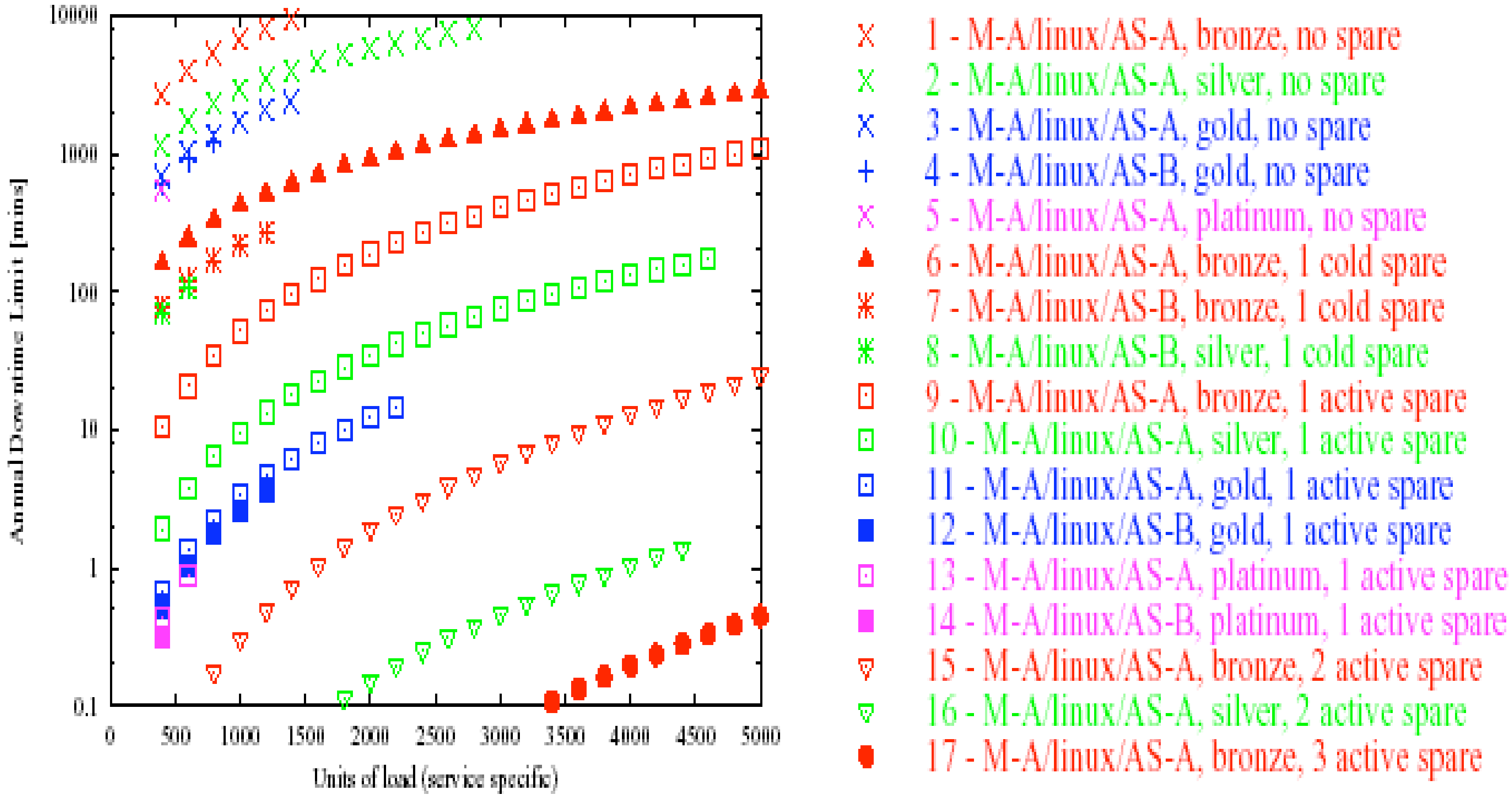


Figure 3: Optimal solution for a range of service requirements: load and annual downtime limit.

Additional Annual Cost Required for Availability

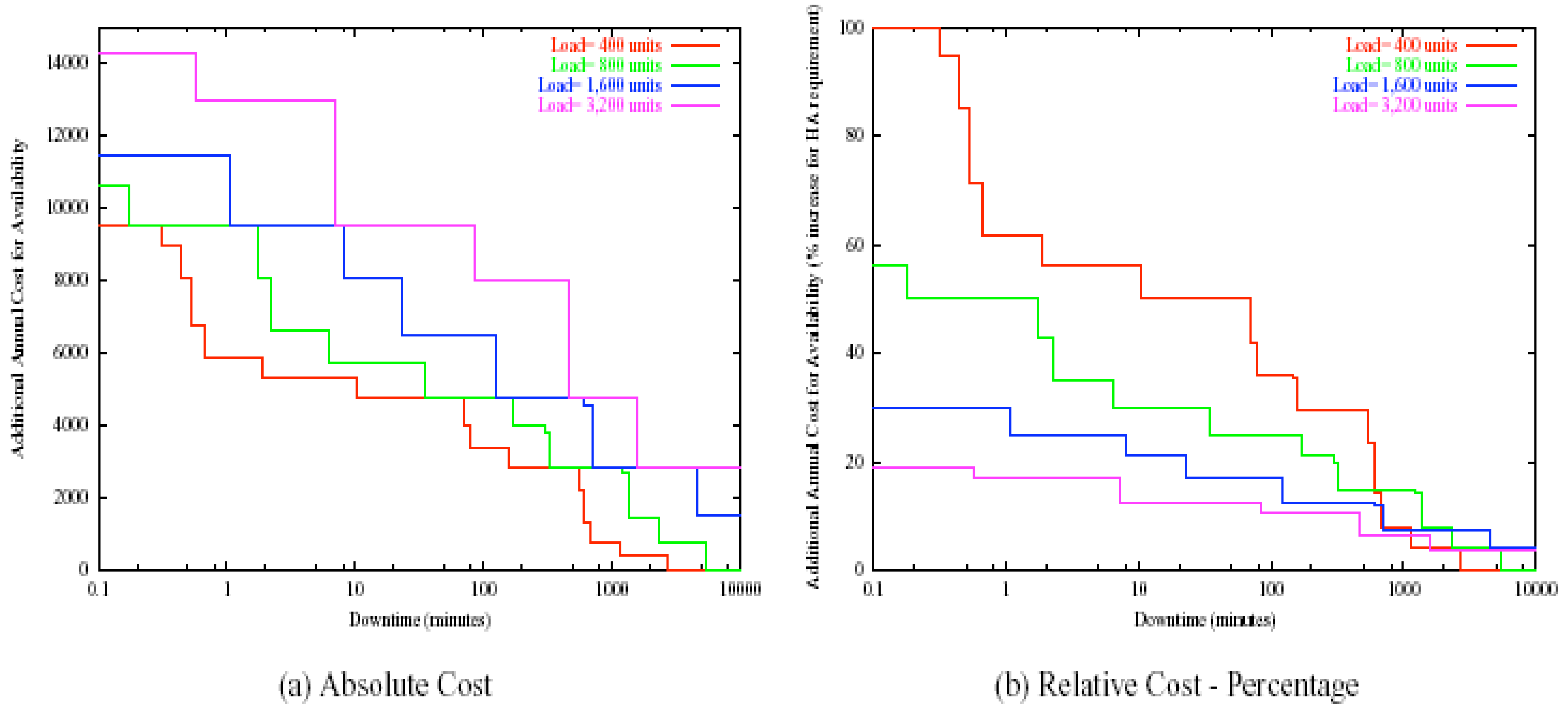
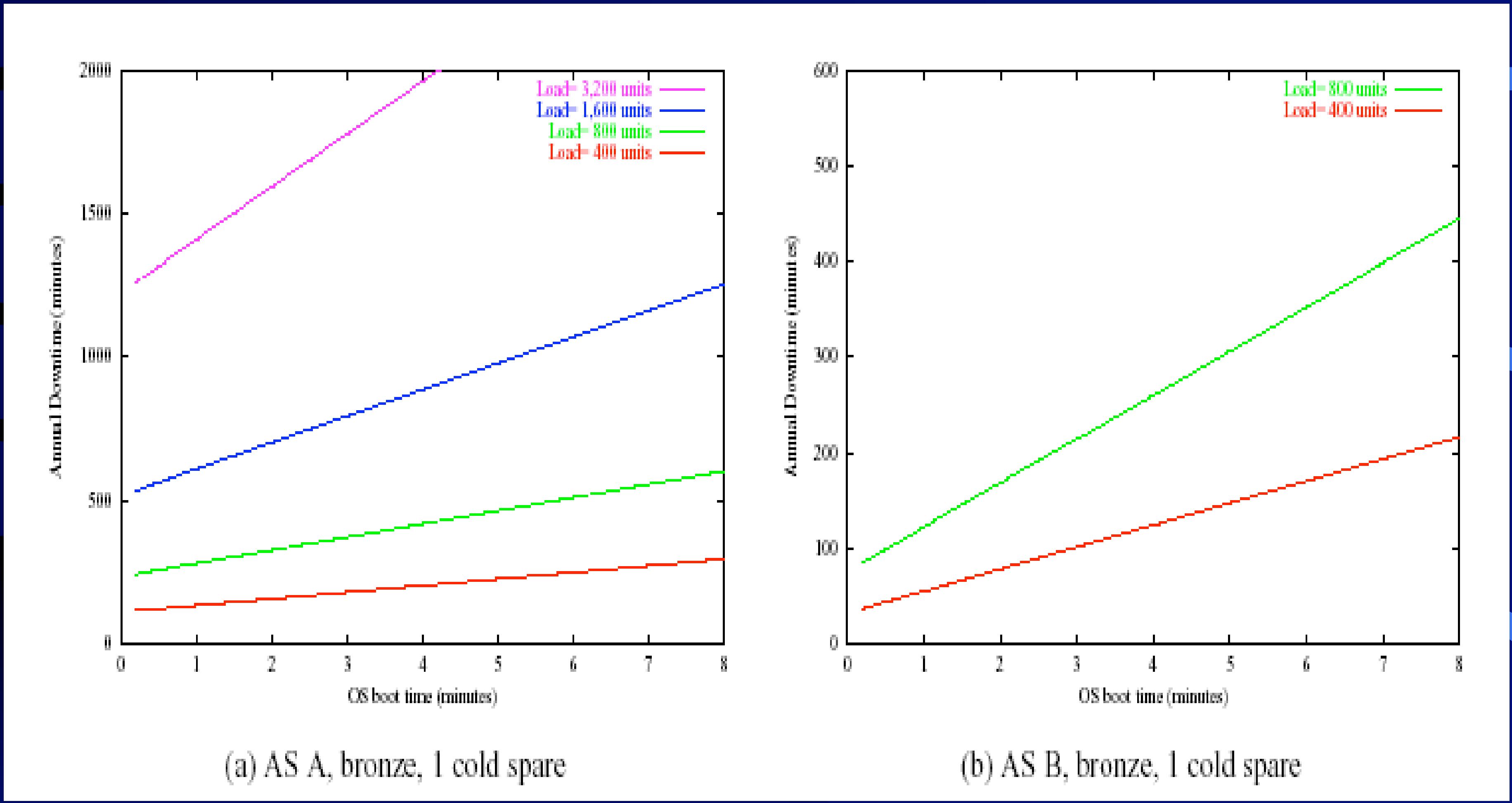
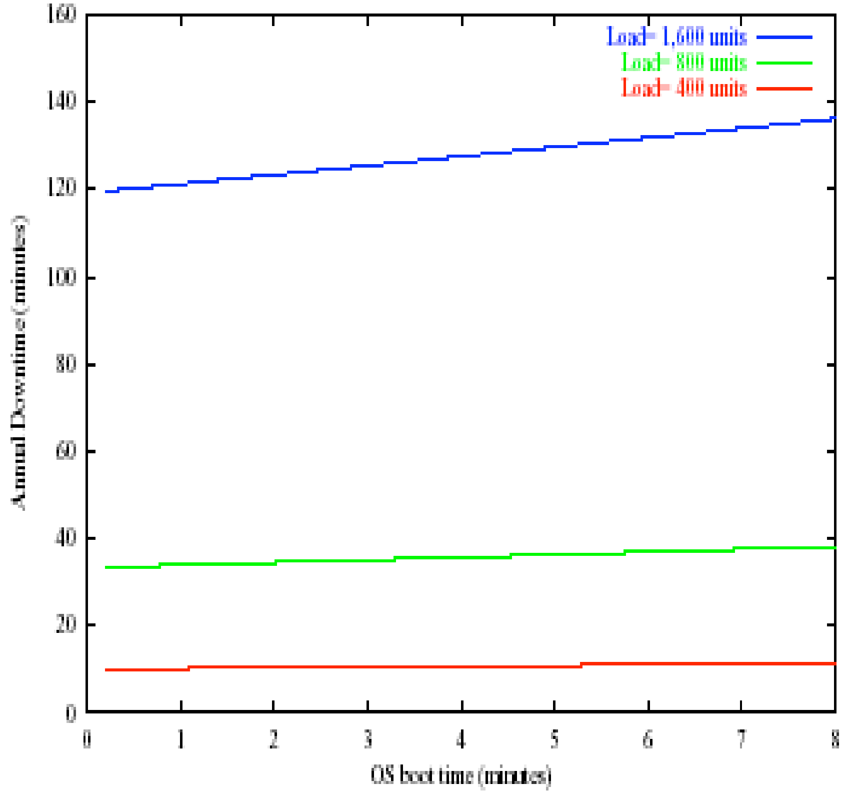


Figure 4: Additional annual cost required for availability.

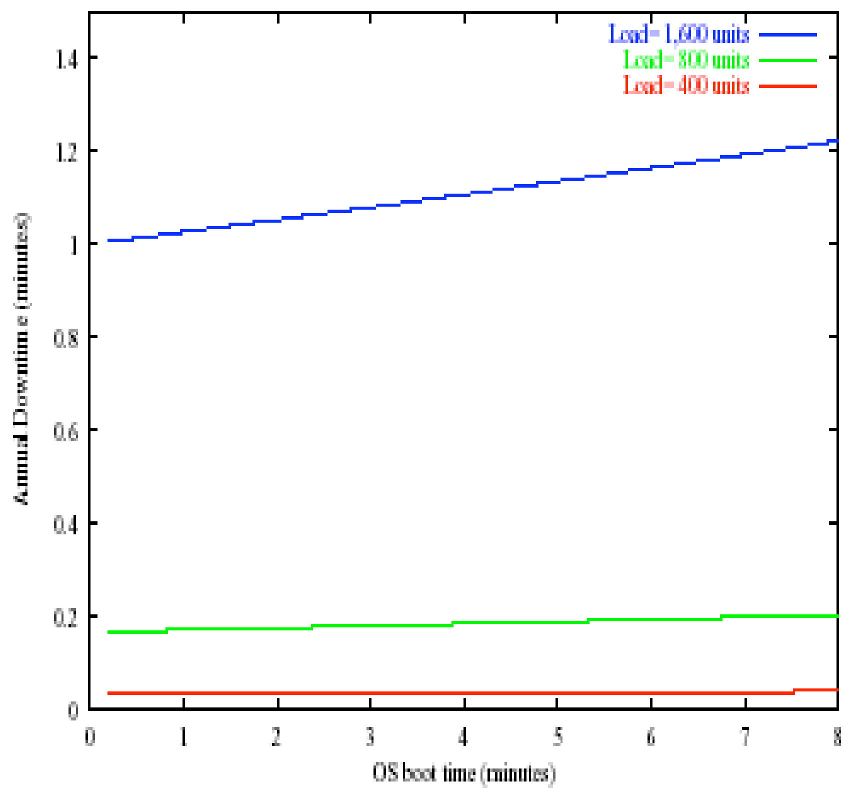
Downtime Sensitivity to Repair Time with Cold Spares



Downtime Sensitivity to Repair Time with Active Spares



(c) AS A, bronze, 1 active spare



(d) AS A, bronze, 2 active spares

Conclusions

- Since a significant loss of business can result from degraded service availability, service availability management is very important.
- Research focused on automating service availability management.
  - Requirements must be at a high level
  - Service Description must specify failure, recovery, and repair parameters.
  - Design function must automatically generate design alternatives, build availability models, and evaluate them to select the best design.
  - System must also perform monitoring, deployment, and configuration.
- These concepts were demonstrated in AVED.

Future Plans

- Addressing overall service availability through examining network and storage system impact
- Factoring in network topologies (LAN), network application placement, and network failures and recovery.
- Improving data dependability
- Making design space richer
  - Database engine configuration parameters
  - Application server configuration parameters
  - Virtual machine usage
  - Software rejuvenation
- Relaxing restrictions
  - Each tier will no longer need to be homogeneous
  - Tiers will be able to have heterogeneous components
- Coupling AVED with a UDC environment