Yüksek Servis Sürekliliği (High Availability)

Yasin KAPLAN - kaplan@intrakets.com.tr

Servis sağlayıcılar açısından servis sürekliliğinin belirlenmesi, servislerinin tanımlanması açısından hayati önem taşımaktadır. Rekabetin sürdürülebilmesi için birim sürede servis sürekliliğinin belirli bir oranın da üstünde tutulması bir zorunluluktur. Öte yandan bunu sağlamanın bir maliyeti vardır ve servis sürekliliğinin makul maliyetlerde gerçekleştirilmesi gerekmektedir. Bunun sağlanması içinde, hangi bileşenlerin servis sürekliliğinde göz önünde bulundurulması gerektiği, bunların birbirlerine nasıl bağlanacağı, söz konusu bir ağ ise, bu ağın nasıl tasarlanacağı nasıl iyileştirileceği, arızalarda nasıl bir stratejinin uygulanacağı üzerinde çalışmak gereklidir.

Günümüzde servis sağlayıcıların baş etmesi gereken zorunluluklar bulunmaktadır:

- Uygulamaya alınmış bir çözümün ömrünü uzatmak.
- Bir çözümün sürekliliğini artırmak; çözüm çalışmadığı sürece, para da kazandırmayacaktır.
- Yeni servis türlerini en hızlı bir şekilde pazara sunmak ve mevcut sistemlerin yeteneklerini ve kapasitelerini düşük maliyetle artırabilmek.

Sürekliliğin Hesaplanması

Sistemin güvenirliği *(Reliability)* ve süreklilik *(Availability)* sıklıkla aynı şeymiş gibi ifade edilse de, - her ne kadar yüksek servis sürekliliği açısından ikisi de önemli olsa da - aslında farklı şeyleri ifade etmektedirler:

Güvenirlik, birim zamanda herhangi bir bileşenin, cihazın veya servisin kesintiye uğramama olasılığıdır. Süreklilik ise, bir zaman aralığında, servisin kesintiye uğramadığı sürenin, toplam süreye oranıdır.

Başka bir değişle, MTTF veya MTBF, iki arıza arası ortalama süre *(Mean Time to Failure veya Between Failure)*, MTTR, ortalama tamir süresi *(Mean Time to Repair)* olmak üzere, Süreklilik = MTTF / (MTTF+MTTR) olarak ifade edilir. Süreklilik MTTF sonsuza veya MTTR sıfıra yakınsadıkça, %100'e yaklaşacaktır. Elbette her zaman daha yüksek yüzde daha tercih edilir servis sürekliliği

anlamına gelecektir. Öte yandan MTTR ve MTBF değerlerinde bir değişiklik ihtiyacı olmaksızın, yedeklilik ile sürekliliğin artırılması mümkündür. Bur durumda hesap şekli değişecektir.

Servis Süreklilik Seviyesi *(9 Sayısı)*	**Süreklilik** *(Availability)*	**Kesinti** *(Downtime)* **/ Yıl**	**Örnekler**
HAL-1	%90.0	36 gün 12 saat	Kişisel müşteriler
HAL-2	%99.0	87 saat 36 dakika	Giriş seviyesi işler
HAL-3	%99.9	8 saat 46 dakika	ISP'ler
HAL-4	%99.99	52 dakika 33 saniye	Data Center'lar
HAL-5	%99.999	5 dakika 15 saniye	Carrier sınıfı Telekom'lar, tıbbı uygulamaları, bankalar
HAL-6	%99.9999	31.5 saniye	Askeri uygulamalar

Yukarıdaki tablodan da anlaşılacağı üzere süreklilik, % 100'e yakın oranlar civarında öngörülmektedir. % 99 kulağa hoş gelse de, bir senelik bir periyotta bu, dört güne yakın bir kesintiye karşılık gelmektedir. Birçok uygulama da yılda yaklaşık dokuz saatlik bir kesintiye karşılık gelen % 99,9 oranına yaklaşılmadığı sürece yüksek servis sürekliliğinden bahsedilememektedir. Telekom endüstrisinde dört ile beş 9'lu oranlar yüksek servis sürekliliğinde bir ölçü olmaktadırlar.

Elbette bu seviyelerde bir servis sürekliliğini sağlamak için maliyetler beklediğinden çok hızlı bir şekilde artmaktadır. Bu durum servis sağlayıcılar kadar cihaz üreticileri için de baş edilmesi gereken bir problemdir. Bu yüzden cihaz üreticileri ürünlerini alt bileşenlere ayırarak maliyet açısından avantaj sağlayacak şekilde sürekliliği iyileştirme eğilimindedirler. Aşağıdaki tabloda bir sistemin alt bileşenlerine ayrılarak sürekliliğin sistemin alt bileşenlerine nasıl dağıtılabileceği gösterilmektedir:

Sistemin sürekliliği	Donanımın sürekliliği ve Yazılımın sürekliliği
Donanımın sürekliliği	Platformun sürekliliği ve I/O kartlarının sürekliliği
Yazılımın sürekliliği	İşletim sisteminin sürekliliği ve Middleware sürekliliği ve Yazılımın sürekliliği ve Uygulamanın sürekliliği

Bir sistemdeki veya ağdaki tüm bileşenlerin aynı süreklilik oranına sahip olması beklenmez. Ayrıca toplam süreklilik son kullanıcı açısından bakıldığında önemi olan bir ölçüdür. Sistemi oluşturan bileşenlerin bireysel süreklilik oranları değişiklik gösterebilir. Maliyet açısından en uygun kombinasyonu sağlamak cihaz üreticileri ve servis sağlayıcılar açısından üzerinde durulması gereken bir konudur.

Veri Ağlarının Süreklilik Üzerinde Etkileri

Bir cihazın veya sistem bileşeninin ağ üzerindeki konumunun süreklilik üzerinde bir ayrıca etkisi vardır. Bileşen omurganın çekirdeğine yakın bir noktada konuşlandığında, süreklilik gereksinimleri kendini daha çok hissettirmeye başlar. Omurganın uçlarında ise gereksinimler daha gevşektir. Örneğin abone veya son kullanıcı ile omurga arasındaki bağlantıyı sağlayan devrelerde *(Local Loop, Yerel Devre)* genellikle bir koruma mekanizması yoktur. Telcordia maliyetlerin hem son kullanıcı ve hem de ağ işleticileri açısından kabul edilebilir bir düzeyde olması için bu tür devrelerde süreklilik oranını %

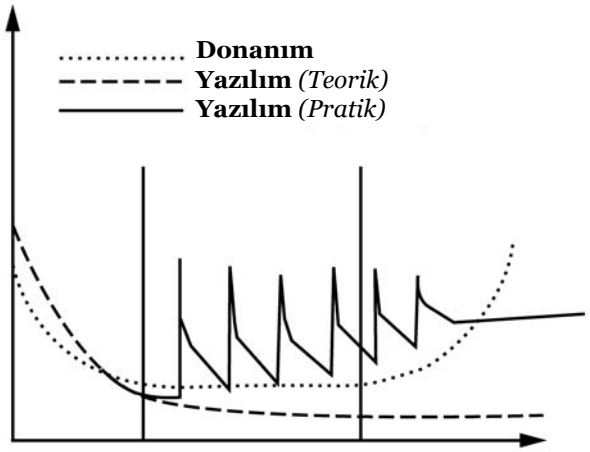
99,93 olarak öngörmüştür. Ancak omurgayı oluşturan ana bileşenler arasındaki bağlantılarda çok daha yüksek süreklilik gereksinimleri bulunmaktadır. Ayrıca hizmet türlerine göre de süreklilik seviyeleri arasında bir farklılık söz konusudur; 155 hizmeti ile 118 hizmeti arasında olduğu gibi...

Belirli bir bileşenin süreklilik gereksinimleri belirlenirken, bileşenin ağ üzerindeki konumu ile birlikte, ne amaçla kullanılacağı ve son kullanıcıya hizmet ulaştırılırken başka hangi bileşenlere bağlanacağı da göz önünde bulundurulmalıdır.

Yaşlanmış Bileşenlerin Arızalardaki Etkileri

Sistemi oluşturan bileşenlerin ömrü ve zamanla normal işletimde davranış değişiklikleri, sürekliliğin hesaplanmasında ve denetim altına alınmasında değerlendirilmesi gereken önemli parametrelerdir. Bir sistemi oluşturan donanım ve yazılım bileşenlerinin zamanla sergiledikleri davranış şekilleri arasında farklılıklar bulunmaktadır.

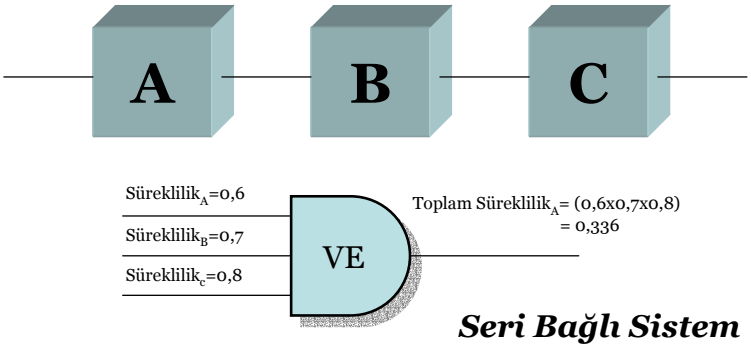
Donanımın sergilediği arıza oranı, arıza oranı - zaman eksenli bir tabloda "Küvet Eğrisi" olarak gözlemlenir. İlk devreye alındığında bir sistemin donanım bileşenlerinde en zayıf parçalar arızalanacak ve yenileri ile değiştirilecektir. Belirli bir zaman sonra arıza oranı donanımdaki zayıf bileşenlerin düzeltilmesi ile düşecek ve istatistiksel olarak sabit bir seyir gösterecektir. Ancak kullanılan malzemenin doğasından kaynaklanan sebeplerden ötürü bir süre sonra ana bileşenler arızalanmaya ve bir süre sonra tümüyle çalışmaz hale gelecektirler.



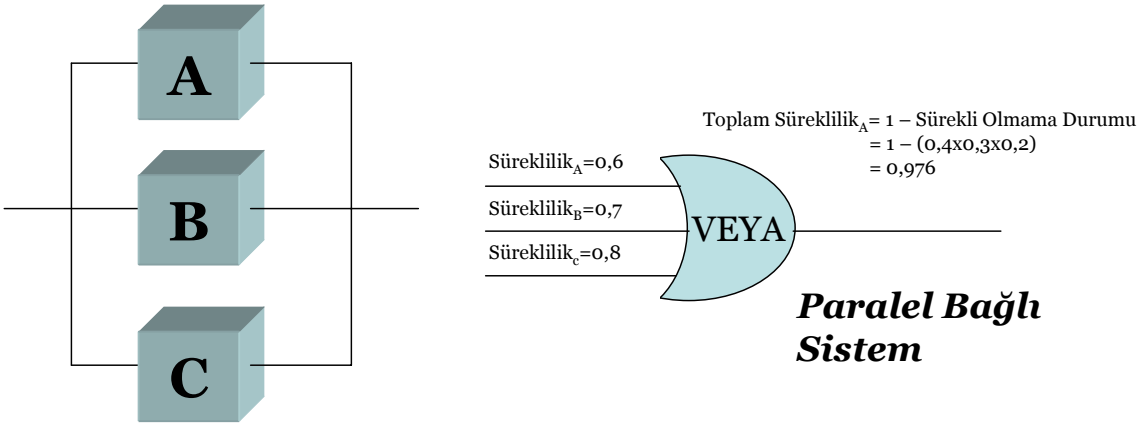
Yazılım ise tıpkı donanımın ilk devreye alınışından sonraki periyotta olduğu gibi olgunlaşacak ve zaman geçtikçe iyice olgunlaşarak arıza oranı sıfıra yaklaşacaktır. Bu teoride sürekli azalan bir fonksiyon gibi görünse de, pratikte yazılımın sergilediği davranış, zamanla genliği giderek azalan sıçramaların bulunduğu bir eğri olacaktır. Bu sıçramalar yazılım güncellemelerini gerektiren arızalar olacaktır.

Bütünleştirilmiş Sistemlerde Sürekliliğin Hesaplanması

Donanım sürekliliği hesaplanırken, sistemi oluşturan tüm ayrık bileşenler, bütünleştirilmiş devreler, tranzistör'ler, diyotlar, dirençler, kondansatörler ve röleler gibi, hesaba katılır. Bu bileşenlerin nasıl bir araya getirildiği sürekliliğin belirlenmesinde önemli bir etkiye sahiptir.



Eğer bileşenler seri olarak bağlanmışsa, süreklilik en zayıf bileşenin bireysel sürekliliğinden çok daha düşük olabilir *(Bellcore yaklaşımı)*. Paralel bağlama yaklaşımında ise, toplam sistem sürekliliği en kuvvetli bileşenin bireysel sürekliliğinden çok daha yüksek olabilir. Bu yüzden tasarımcılar mümkün olduğunca paralel bağlama yaklaşımını tercih ederler. Maliyetlerin başlangıçta kontrol altında tutulabilmesi için başlangıçta asgari gereksinimlerle kurulan bir yapıda daha sonra eklenecek paralel bileşenlerle süreklilik yüzdesinin artırılması yoluna gidilebilir.

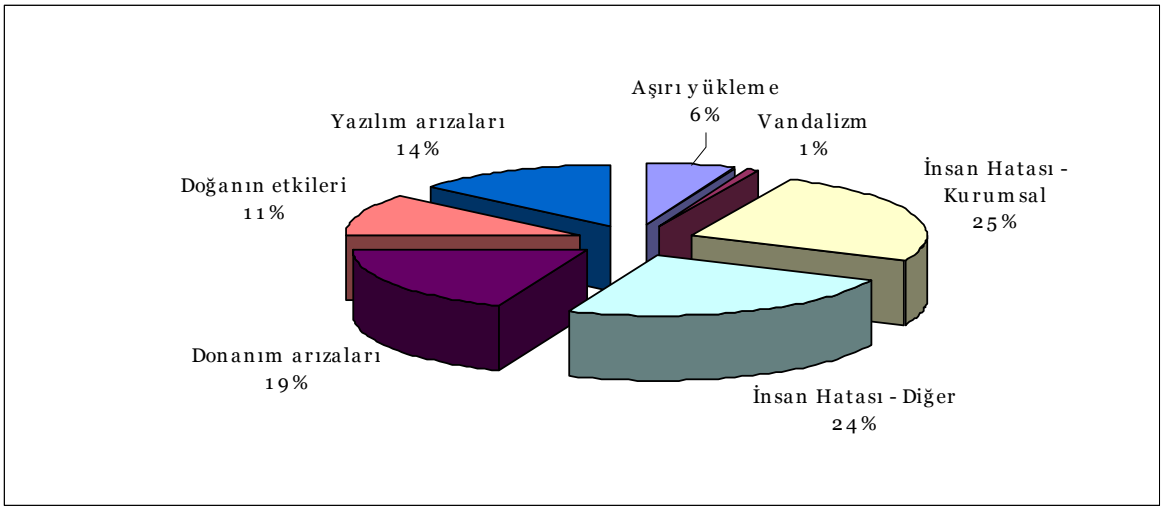


Sisteme yedeklilik özellikleri eklendiğinde sürekliliğin hesaplanması oldukça zahmetli bir işlem haline gelebilir. Yedek bileşenlerin devreye girme süresi *(Failover Time)*, MTTR'ın sürekliliğe olan etkisinin pek çok açıdan incelenmesi gerekir. Daha kestirme ve sağlıklı sonuçlara Güvenilirlik Blok Diagramlarını *(Reliability Block Diagrams, RBD)* kullanarak ulaşmak mümkündür. RBD'ler ile sistemi oluşturan bileşenler ve süreklilik karakteristikleri daha kolay bir şekilde analiz edilebilir. Yedeklilik daha çok göz önünde bulundurulduğundan, Bellcore yaklaşımına göre daha gerçekçi sonuçlar elde edilebilir.

Sistemin karmaşıklığına göre çalışma koşullarının benzetimi gerçekleştirilerek, istatistiksel olarak da sürekliliğin hesaplanması mümkündür. Ancak bu yaklaşım her ölçekteki sistem için sağlıklı bir sonuç vermez.

Kesintinin Sebepleri

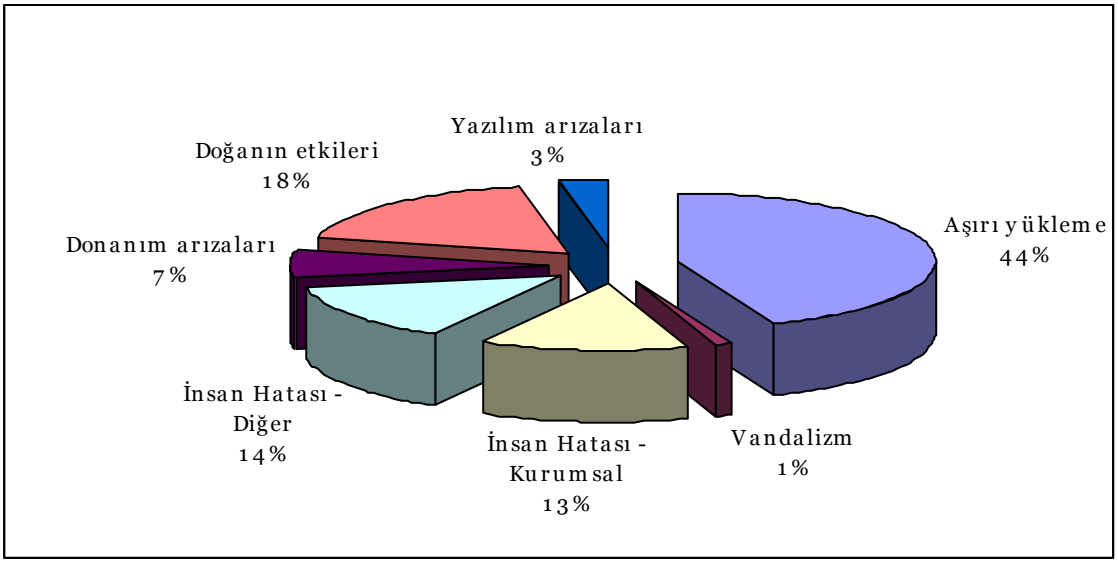
Bir sistemdeki kesintinin birçok sebebi olabilir. Bunların dağılımı sistemin nasıl tasarlandığı kadar nasıl ve hangi şartlarda işletildiği ile de ilgilidir.



Kamusal Anahtarlamalı Telefon Ağlarında *(PSTN)* arızaların sebep olduğu kesinti sebeplerinin sayılarının dağılımı *(Sources of Fauiler in the PSTN - D. Richard Kuhn, National Instute of Standarts and Technology)*

Kesintilerin sebepleri arasında en büyük payı aşırı yükleme durumları almaktadır. Bu durum çoğunlukla yeni bir hizmet duyurulduğu zaman öngörülemeyen bir talep patlaması gerçekleştiğinde yaşanmaktadır. Benzeri durumlar özel zamanlarda da yaşanabilir. Örneğin anneler gününde bir telefon ağında yaşanacak aşırı yükleme gibi.

Bu tür durumlardan kaçınmak için sistem ek kapasite ile tasarlanmalı, ağ bileşenlerinin yönetimi için harici yönetim, out of band management, yöntemleri tercih edilmelidir. Ek kapasite aşırı talebin karşılanmasını olanaklı kılarken, harici yönetim aşırı ağ trafiğinin ağ bileşenlerinin yönetilemez hale gelmesinin önüne geçer. Ek kapasite sisteme aynı anda bağlanabilecek azami kullanıcı sayısı üzerinden hesaplanmalıdır.



Kamusal Anahtarlamalı Telefon Ağlarında *(PSTN)* arızaların sebep olduğu kesinti sürelerinin dağılımı *(Sources of Fauiler in the PSTN - D. Richard Kuhn, National Instute of Standarts and Technology)*

Sürekliliği etkileyen etkenlerden biri de sistemi oluşturan bileşenlerdeki planlı veya plansız kesintilerdir. Sistem güncellemeleri ve bakım çalışmaları planlı kesintilerin sebeplerini oluşturlar. Bu tür çalışmaların süreklilik yüzdesini asgari olarak etkilemesi için yaklaşımlar bu değerlendirmede daha sonra tartışılmaktadır. Plansız kesintilerde ise genellikle donanım ve yazılım hataları ile birlikte yetersiz eğitim almış, tecrübesiz operatörlerin rolü *(Gartner raporlarına göre %40 oranında!)* bulunmaktadır.

Yapılan çalışmalar planlı kesintilerde yaşanan kesinti sürelerinin azalma eğiliminde ancak plansız kesintilerde sürenin artma eğiliminde olduğunu göstermektedir. Bu yüzden tecrübeli, iyi eğitim almış operatörlerin istihdam edilmesi ve sürekliliği artıracak tedbirler üzerinde çalışılması gerekmektedir.

İnsan hatalarının önüne geçmenin en kestirme yollarından biri otomatik düzeltme işlevlerinin ağ veya sistemde uyarlanmasıdır. Bu yaklaşım gerekli iş gücünden de tasarruf sağlayacaktır. Arızalarda otomatik düzeltme planlarının yapılması kadar, abone işlemlerinin, servis aktivasyon süreçlerinde operatörden bağımsız, el değmeden, yapılabilir hale gelmesi de sürekliliği olumlu olarak etkileyecektir. Başka bir yaklaşım da tasarımın sade tutulmaya çalışılması, ara bağlantıların ise basitleştirilmesidir.

Planlı Olmayan Kesintilerin Yönetimi

Ne kadar güvenilir bileşenler kullanılırsa kullanılsın, ne kadar detaylı kalite denetimi kuralları uygulanırsa uygulansın, arızalar kaçınılmazdır. Bu yüzden hizmet sürekliliğinin iyileştirilmesinde hata tespiti süresinin kısaltılması ve MTTR değerinin düşürülmesi çok önemlidir. Planlı olmayan bir kesintinin veya arızanın idare edilebilmesi için dikkatli bir şekilde hazırlanmış bir arıza yönetim planın bulunması bir zorunluluktur. Aşağıda bir arıza durumunda izlenmesi gereken adımlar sıralanmaktadır:

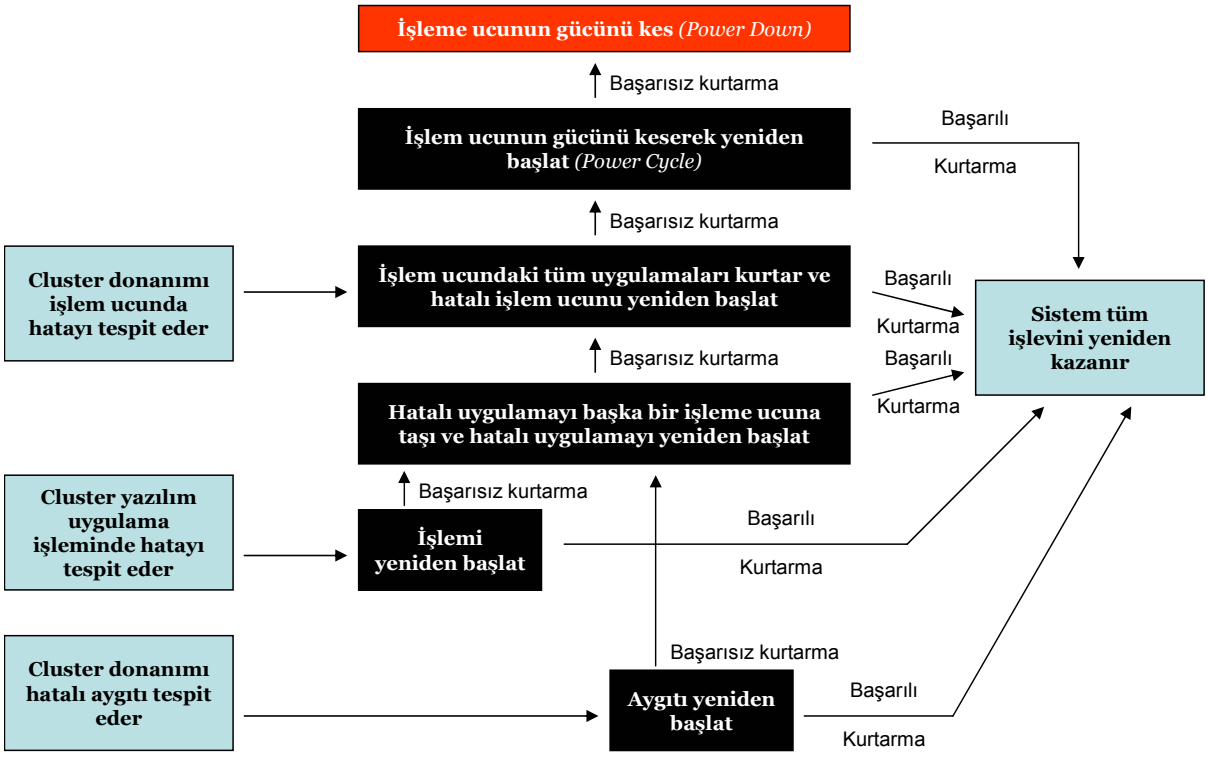
1. **Tespit**. Arızanın belirlenmesi. Ancak bu aşamada henüz arızalı bileşen tanımlanmamıştır. Tespit ağdaki cihazlardan gelen alarmlarla veya ağdaki trafik değişimleri ile sağlanabilir. Trafik değişimlerinde de yönetim sistemlerinin alarmlar üretmesi olasıdır. Birçok arıza gerçekleşmeden önce yönetim sistemlerindeki trafik değişiklikleri bazı belirtiler içerebilirler. Bu yüzden alarmlar kadar sistemin genel işletimi sırasında normal olmayan trafik değişikliklerinin de çok iyi izlenmesi ve analiz edilmesi gereklidir. Bu yaklaşım birçok arızanın ve kesintinin gerçekleşmesinden önce tahmin edilebilmelerini ve düzeltilebilmelerini olanaklı kılar.
2. **Teşhis**. Arızalı bileşenin belirlenmesi. Teşhis işlemi de kendi içinde aşamalı bir işlemdir. Genelden özele doğru bir dizi kontrol işlemeni içerir. Sistemin yapısına göre izlenecek prosedür önceden tanımlanmalıdır. Örneğin bir ISP'nin PoP noktasının erişilemez olduğu durumda, önce bağlantıyı sağlayan cihazların, konfigürasyonlarının, ara yüzlerinin ve nihayet transmisyon sisteminin kontrol edilmesinde izlenecek sırada olduğu gibi...
3. **İzolasyon**. Arızanın genel bir sistem arızasına sebep olmasının önüne geçilmesi *(Bu aşamada sistemin tümüyle eski halinde çalışır duruma getirilmesi söz konusu değildir)*. Kimi arızalar zincirleme bir reaksiyonu başlatarak sistemi tümüyle çalışmaz hale getirmesine sebep olabilir. Olası bir durum arızalı bir hattan geçmesi gereken trafiğin, çalışan diğer hatlardan iletilmeye çalışılmasıyla, bu hatların sature edilmesidir.
4. **Düzeltme**. Sistemin kendinden beklendiği gibi çalışır hale getirilmesi. Düzeltme arıza türüne göre, konfigürasyon hatasının düzeltilmesi, arızalı bileşenin değiştirilmesi veya alternatif bir hattın devreye sokulması ile gerçekleştirilebilir.
5. **Tamir**. Sistemin yedeklilik özelliği dahil tam kapasitede çalışır hale getirilmesi.

Servis Sürekliliğini Artırmak için Olası Mimariler

Sistem seviyesinde arızalardan korunmanın en etkin yolu yedekliliktir. Bileşenlerdeki yedeklilik miktarı ve çeşitliliği bir sistemin devre dışı kalma durumundaki karakteristiğini belirler. Aşağıda en yaygın yedekleme yöntemleri bulunmaktadır.

Clustering

Clustering yönteminde, bir bilgisayarın, sunucunun veya sistemin tümü çoklanarak paralel çalışması sağlanır. Cluster dizisindeki bir sistem devre dışı kalırsa işlevleri, dizideki başka bir sisteme aktarılarak sistemin sürekliliği sağlanır. Dizideki yedek sistem sayısı, N sistem sayısı olmak üzere, yedeklilik 2xN veya N+1 şeklinde uyarlanabilir. Yedek sistemler aktif veya pasif durumda bulunabilirler. Pasif durumda yedek sistem çalışmaya hazır halde beklerken, aktif durumda, asıl sistemler ile senkronize çalışma durumu söz konusudur. Aktif durumda ayrıca yük paylaşımını da sağlamak mümkündür ancak uygulanması biraz zordur. Öte yandan uygulanabildiği durumda finansal açıdan daha avantajlı olmaktadır. Zira tüm sistemler çalışır durumda ve yük paylaşımı gerçekleştirildiği için sistemin toplam performansı da artacaktır.



Clustering ile arıza yönetimi

Clustering herhangi bir PC tabanlı sistem ve standart ağ bağlantıları ile çalıştırılabilir. Birbirinin yedeği olan sistemler bu ağ bağlantıları üzerinden birbirlerini yoklayabilirler. Clustering ayrıca birbirinden coğrafi olarak ayrık sistemlerin yedeklenmesi için de tercih edilebilecek bir yöntemdir. Bu yüzden doğal afetlerin sebep olabileceği servis kesintilerinden kaçınılmak amacı ile devreye alınabilir. Clustering ayrıca yazılım hatalarının sebep olabileceği servis kesintilerine karşı da koruma sağlar.

Clustering'in en zayıf iki yanı uygulama maliyeti ve hata anında sistemin olağan çalışma durumuna geçmesi için gereken sürenin uzunluğudur. Clustering uygulanacağı zaman yedeklenecek sistem(ler)in eşleniği bir sistemi hazırda bulundurmak gerektiğinden, maliyetler katlanacaktır. Uygulama şekline göre asıl sistemde bir arıza meydana geldiğinde, yedeği asıl ile senkronize hale getirmek uzun zaman alabilir *(Diğer alternatiflerde bu milisaniye mertebesinde iken, Clustering'de bu süre saniyelere mertebesinde gerçekleşebilir)*.

Donanım Hata Töleransı

Donanım Hata Töleransı, Merkezi İşlem Biriminin *(CPU)* çoğaltılarak aynı işlemlerin paralel olarak işletilmesine dayanır. CPU'ların çıkışlarındaki değerler karşılaştırılarak, sonuçlar arasında bir fark olup olmadığı incelenir. Ancak iki CPU ile bu karşılaştırmayı yapmak çok anlamlı olmadığında en az üç CPU'nun devrede olduğu yaklaşım kullanılır. Bu yaklaşım daha etkin bir hata izolasyon yeteneği sağlar. Eğer bir CPU'nun sonucu diğer ikisi ile uyuşmaz ise, farklı sonucu üreten CPU arızalı olarak kabul edilir ve tamir için devre dışı bırakılır.

Bu yaklaşımın en önemli avantajı, sağlanan yedekliliğin uygulamaya saydam olmasıdır. Bu tür bir yedekliliği sağlanması için uygulama da herhangi bir değişikliğin yapılmasına gerek yoktur. Ayrıca olası bir hata servis kesintisine veya gecikmeye sebep olmaz. Ancak bu yaklaşım yazılım hatalarına karşı bir koruma sağlamaz.

Çevresel Aygıtların (Peripheral) **Hot Swap Oluşu ve Yedeklilik**

Peripheral Hot Swap *(PHS)* özelliği, bir sistemdeki bir bileşenin, (bir kart, disk veya güç kaynağı), sistemin kapatılmasını gerektirmeden çıkarılıp takılmasını sağlar. Bu yaklaşım, sistemde, planlı olsun ya da olmasın, parça değişimi, güncelleme veya ekleme yapılması sırasında, sistemin servis sürekliliğinde bir aksama yaşanmasının önüne geçer.

PHS, tamir süresi *(Time to Repair)* üzerinde olumlu etkileri olmasına rağmen, yalnız başına, işletim sırasında meydana gelecek bir arızada veya yedek bir kartın temin edilmesi sırasında yaşanacak servis kesintisi için koruma sağlamaz. Bu yüzden çevresel aygıtların yedekliliği geliştirilmiştir. Bu şekilde, herhangi bir çevresel aygıt devre dışı kaldığında, yedeği söz konusu aygıtın işlevini üstlenerek servisin aksamasını engelleyecektir. Bu yaklaşım ayrıca bir bakım veya güncelleme sırasında sistemin çalışmasında olumsuz bir etki olmaksızın işlemin gerçekleştirilebilmesine olanak tanır.

Paket Anahtarlamalı Veriyolu

Paket anahtarlamalı veriyolları, sunucu sistemlerinde çevresel aygıtların ve CPU'nun paket anahtarlamalı bir şekilde haberleşebilmesini sağlarlar. Paket anahtarlamalı veri yollarında, veri yoluna bağlı aygıtlar daha az iletken kullanarak birbirleri ile seri olarak haberleşirler. Kullanılan iletken sayısının azaltılması arıza olasılığını azalttığı gibi, seri haberleşme de bileşenler arasındaki mesafenin artırılabilmesine olanak tanır. PC'ler için geliştirilmekte olan StarFabric ve InfiniBand gibi paket anahtarlamalı veri yolları bulunmaktadır. Bu yaklaşımın başka bir avantajı ise, klasik veri yollarındaki band genişliği darboğazlarının aşılmasına olanak tanımalarıdır. Günümüzde PCI-X ile kabaca 1 Gbps

veri band genişliğine erişilmiştir ancak bu bile şimdiden yetersiz kalmaktadır. Infiniband ile teorik olarak sınırsız band genişliğine erişmek mümkündür.

Ağ Yönlendirmesi

Ağ yönlendirmesi OSI modeline göre üçüncü katmanda, bir ağa bağlı sistemler arası trafiğin en elverişli ve erişilebilir uçlar arasında iletilebilmesini sağlar. Dinamik yönlendirmenin kullanıldığı ağlarda, ağın güncel durumu ağa bağlı uçlara duyurularak, bu uçların en elverişli yollar üzerinden haberleşmesi sağlanır. Söz konusu ağın sürekliliğinde bir bozulma olduğunda, uçlara ağın yeni durumuna ilişkin güncellemeler gönderilerek, uyum sağlamaları olanaklı hale getirilir. Ağ yönlendirmesi fiziksel olarak birbirinden çok ayrık sistemler arasında yedekliliğin sağlanması için elverişli bir çözümdür ancak ağ yapısı, ara bağlantıların türü ve kapasitesine bağlı olarak arızalara yavaş cevap verilmesine sebep olabilir.

Ağ yönlendirmesi OSI modeline göre üçüncü katmanda bir işlemdir. Günümüzde daha üst katmanlarda anahtarlama ve yönlendirme yeteneğine sahip cihazlar kullanımdadır *(Layer 4-7 Switch'ler)*. Bunlarla ağdaki aksaklıklara daha hızlı cevap vermek mümkündür. Ağdaki konumlarına göre bu cihazlarla yük dengelemesi ve yedekleme de gerçekleştirilebilir.

Sonuç

Servis sürekliliği üzerine yapılan çalışmalar ve deneyimler aşağıdaki sonuçlara erişilmesini sağlamıştır:

- Yüksek servis sürekliliğini sağlamada anahtar yedekliliktir. Hiçbir zaman hatasız ve sürekli çalışan bir aygıt veya cihaz yapmak, pek çok açıdan, kullanılan malzemeden tutunda işçiliğe ve tasarıma kadar, mümkün değildir. O halde sistemi oluşturan bileşenlerin yedekliliği sağlandığı ölçüde, süreklilik de artırılmış olur.
- N bileşen sayısı ve M yedek bileşen sayısı olmak üzere, M artıkça sürekliliktekı artış oranı, artış hızı, azalacaktır *(Ters J eğrisi)*.
- Süreklilik doğrudan sistemi oluşturan bileşenlerle ilgilidir. Bu bileşenlerin birbirleri ile nasıl irtibatlandırıldıkları da göz ardı edilmemelidir.
- MTTR azaldıkça sistem sürekliliği de artacaktır. Örneğin boot süresinin kısaltılması, hızlandırılmış hata tespiti ve sistem güncellemelerinin daha kısa sürede yapılabilmesi MTTR süresini kısaltacaktır.
- Sistem üzerindeki çalışan yazılım zaman geçtikçe iyileşirken, donanım daha çok arızaya sebep verme eğiliminde olacaktır.
- Sistem maliyeti, servis sürekliliğinin arttığından daha hızlı bir şekilde artacaktır. Servis sürekliliğinde üst sınırlar zorlandığında, daha küçük artışlar için daha büyük maliyetler söz konusu olacaktır.

Bu değerlendirmede sistem ve ağ seviyesinde servis sürekliliğinin artırılabilmesi için yaklaşımlar tartışılmaktadır. Bir sistemin sürekliliği hesaplanırken bileşen arızaları, bunların tamir süresi dışında doğal afetler, terörizm ve sabotaj gibi rastlanma sıklığı düşük ancak servis sürekliliğine etkisi büyük olabilecek olaylar da analiz edilmeli ve hesaba katılmalıdır.

Ağ tasarımı, sürekliliğin sağlanmasında en önemli aşamalardan biridir. Hataların tespiti, ve doğru alarmların üretilmesi bir kesinti sırasında atılması gereken en önemli adımdır. Daha sonra teşhis, hatanın izole edilmesi ve tamir aşamaları gelmektedir. Tamir stratejisinin bir parçası olarak ağ ek bir kapasite ile tasarlanabilir. Böylelikle bu ek kapasite, tamir safhasında kullanıcıların asgari bir şekilde kesintiden etkilenmesini sağlayabilir.

Her servis sağlayıcı için, aynı yüksek servis sürekliliği modelinin uygulanması elbette mümkün değildir. Uygulanacak model, servis sağlayıcının hedeflerine ve iş modeline göre değişiklik arz edecektir. PSTN sisteminde kullanılan bir santral için süreklilik Bellcore GR-929-CORE'da %99,99943 olarak öngörülmüştür; bu tür bir sistemin yerine tasarlanan bir Soft Switch için beklenen süreklilik oranı da elbette bu mertebede olacaktır. Süreklilikte "dokuz" sayısı arttıkça, maliyetlerin de artacağı unutulmamalıdır.

Kaynaklar:

1. Kamusal Anahtarlamalı Telefon Ağında Arızaların Kaynakları *(Sources of Failure in the Public switched Telephone Network)*, D. Richard Kuhn, National Instute of Standards anf Technology, http://hissa.nist.gov/kuhn/pstn.html, 1997
2. High Avaibility Arhitecture for IP-Centric System Solutions, Force Computers, 2002
3. Economics of High Availability, An Intel Primer, Intel Corporation, 2002
4. Always-On Availability for Mutliservice Carrier Networks, Cisco Systems, 1999