KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ARAŞTIRMA PROBLEMLERİ

BUILDING AN INFRASTRUCTURE WITH AUTOMATION TOOLS AND LINUX SYSTEMS

MUHAMMET CÜNEYD KURTBAŞ

Prof.Dr. Fazıl SARAY
Danışman, Selçuk Üniv. .....

Doç.Dr. Cengiz ÖZCAN
Jüri Üyesi, Selçuk Üniv. .....

Tezin Savunulduğu Tarih: 20.05.2008

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışması linux sistemlerinin otomasyon araçları ile dizayn ve yönetimini sağlamak amacıyla gerçekleştirilmiştir.

Tez çalışmamda desteğini esirgemeyen, çalışmalarıma yön veren, bana güvenen ve yüreklendiren danışmanım Prof.Dr. Ahmet SAYAR'a sonsuz teşekkürlerimi sunarım.

Tez çalışmamın tüm aşamalarında bilgi ve destekleriyle katkıda bulunan hocam Dr. Talha OKTAY'a teşekkür ediyorum.

Tez çalışmamda gösterdiği anlayış ve destek için sayın Çağlayan SANCAKTAR'a teşekkürlerimi sunarım.

Hayatım boyunca bana güç veren en büyük destekçilerim, her aşamada sıkıntılarımı ve mutluluklarımı paylaşan sevgili aileme teşekkürlerimi sunarım.

Mayıs – 2008 M.Cüneyd Kurtbaş

Bu dokümandaki tüm bilgiler, etik ve akademik kurallar çerçevesinde elde edilip sunulmuştur. Ayrıca yine bu kurallar çerçevesinde kendime ait olmayan ve kendimin üretmediği ve başka kaynaklardan elde edilen bilgiler ve materyaller (text, resim, şekil, tablo vb.) gerekli şekilde referans edilmiş ve dokümanda belirtilmiştir.

Öğrenci No: 200201132

Adı Soyadı: Muhammet Cüneyd KURTBAŞ

İmza: ……..

İÇİNDEKİLER

BUILDING AN INFRASTRUCTURE WITH AUTOMATION TOOLS AND LINUX SYSTEMS

ÖZET

Bu tezde projenin mimarisi, gerekliliklerin kurulumları, kurulumları yaparken karşılaşılan problemler, bu problemlerin çözüm yöntemleri ve elde edilen kazanımlardan bahsedilmiştir.

SSH yapısı ile uzak makinelere .pem key-pair dosyaları aracılığıyla bağlantı yapılması

(Console(Komut İstemi), PuTTy, Termius), Açık kaynaklı, tutarlı bir iş akışı sağlayan, altyapının kod ile oluşturulmasını sağlayan, Terraform ile bulut sunucularının yönetilmesi, yapılandırılması ve yeni kaynakların yaratılması, Minimum insan etkileşimi ile projemiz için gerekli araçlar ve docker-docker swarm kurulumlarının; yönetimimizde olan sunucuları tek bir çatı altında (host inventory) toplayarak karmaşık kurulum setleri ile yapılması yerine, Ansible playbook'ları (yaml) ile tam bir otomasyon sağlanması, Open source bir container teknolojisi olan ve birbirlerinden bağımsız containerlar sayesinde sanallaştırma sağlayan Docker ile ilgili araştırma yapılmış ve projemiz için gerekli container yapılandırmaları ve kurulumların yapılması, Docker swarm ile uzak makinelerin aynı ağa dahil edilmesi ve birbirleriyle konuşabilmesinin sağlanması süreçleri gerçekleştirilmiştir.

Anahtar kelimeler: terraform, ansible, docker, swarm, open source playbooks

GİRİŞ

Diğer veri yapılarında olduğu gibi kuyrukta bulunan elemanlar, string veya integer gibi basit veri türünde olabileceği gibi bir nesne de olabilir. Öncelik kriteri kuyruk türüne göre farklılık göstermektedir.

Bu uygulamanın amacı; Bir iniş bir kalkış olmak üzere 2 pisti bulunan İstanbul Havalimanı'nda gün içerisinde (1-24 saat dilimi boyunca) yapılan uçuşların yönetimi için bir sistem geliştirilecektir. Havalimanında aynı anda sadece bir uçak kalkış yapabiliyorken sadece bir uçak iniş yapabilmektedir. Uçakların her biri iniş ve kalkışta farklı önceliklere sahiptir ve bir günde en fazla 24 uçak iniş için izin isteyebilmektedir. Havalimanındaki uçakların öncelik sırası, iniş saati, gecikme süresi ve kalkış saati bilgileri kullanılarak; iniş pistini ve kalkış pistini kullanım sırasının belirlenmesi hedeflenmektedir.

YÖNTEM

Havalimanına iniş yapacak uçaklar öncelikle kuleden iniş yapabilmek için izin talep edecektir. İniş izni talep eden her bir uçak için havalimanında yeterli kapasite olup olmadığı kontrol edilecek, pist boş ise iniş yapılmak istenen saate izin verilecek ve uygun yere eklenecektir. Aksi halde uçakların iniş sıralaması önceliğe göre belirlenmelidir. Uçakların iniş ve kalkışları, 1.Ambulans uçağı, 2.Savaş uçağı, 3.Yolcu uçağı, 4.Kargo uçağı önceliğine göre belirlenecektir. Havalimanına iniş talep eden uçakların önceliği, uçak numarası ve talep ettiği iniş saati input.txt dosyasından okunacaktır. Havalimanına iniş yapan her uçağın, kalkış için bekleme süresi 1 saat ve uçakların kalkış saatine, ötelenmeden dolayı oluşan gecikme süreleri dâhil edilecektir. Aynı önceliğe sahip iki uçak, aynı saatte kalkış yapacaksa öncelik ilk iniş yapan uçağa verilecektir.

Uygulamada önceliği yüksek olan uçaklar nedeniyle önceliği düşük olan herhangi bir uçağın uçuş saati, en fazla 3 kez ertelenebilecek, 3'ten fazla ertelenme durumu söz konusuysa, öncelik gözetilmeksizin beklemede olan uçağın kalkışı gerçekleştirilecektir. Her yeni input satırı okunduğunda, kalkış yapacak olan uçakların bulunduğu output.txt dosyası güncellenecektir. output.txt içeri her satırda Öncelik ID, Uçak ID, Talep Saati, İzin Verilen İniş Saati, Erteleme Süresi ve Kalkış Saati bilgilerini içerecektir.

UYGULAMA

Uygulamamızın kod tarafına geçmeden önce aşağıda gösterilen örnek bir modele göre iniş izni talepleri ve uçaklara ait bilgileri alarak adım adım öncelikli kuyruğumuzu gerçekleştirelim;yeri uzaklığından tek bara ölçümlerini kullanan algoritmalar kadar etkilenmezler, ancak her baradan senkronize ölçüm alması gerektiğinden maliyeti yüksektir.

Fig. 1.

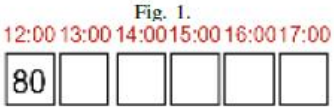
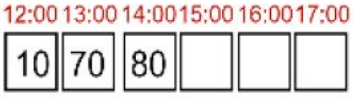


Fig. 2.

Fig. 3.

Fig. 4.



80 ID'li uçak öncelik durumuna uygun olarak 2 saat ertelendi.


Uçak ID: 50
Öncelik ID: 3
İniş Talep Saati: 12.00
12:00 13:00 14:00 15:00 16:00 17:00
10 70 80 50
50 ID'li uçağın talebi öncelik durumuna
uygun olarak 3 saat ertelendi.
Uçak ID: 40
Öncelik ID: 4
İniş Talep Saati: 13.00
12:00 13:00 14:00 15:00 16:00 17:00
10 70 80 50 40
40 ID'li uçağın talebi öncelik durumuna
uygun olarak 3 saat ertelendi.

Fig. 5.


Uçak ID: 20
Öncelik ID: 1
İniş Talep Saati: 16.00
12:00 13:00 14:00 15:00 16:00 17:00
10 70 80 50 20
20 ID'li uçağın talebi onaylandı.
40 ID'li uçağın talebi öncelik durumuna
uygun olarak 3+1 saat ertelenmesi nedeniyle
başka bir havalimanına yönlendirildi.

DENEYSEL SONUÇLAR

Öncelikli kuyruk prensipte yine bir kuyruk çeşidi olmakla birlikte sadece bir davranışı normal kuyruktan farklılık gösterir. O da ilk giren ilk çıkar (first-in, first-out) şeklinde değil, nasıl sıraya girilirse girilsin daha önemli olan önce çıkar, şeklinde çalışmasıdır.

Temel olarak fonksiyon/metodlarında bir farklılık bulunmaz. Sadece önceliklendirme işleminin nasıl yapılacağının tanımlanması gerekir.

Bağlı liste oluşturma işleminde öncelikli olarak bir tane root düğümü oluşturup ardından gelen düğümleri sıra ile birbirlerini «next» pointer'ı ile işaret etmektedir.

Kuyruk işlemi de bu işleme çok benzemektedir. Sadece kuyruğa eklenen en son elemanı ayrı bir biçimde tutmamız gerekmektedir.

Kısacası root düğümünün yanında bir tane de «next» düğümü oluşturulur. Dolayısıyla Kuyruk birbirine bağlı liste haline dönüşmüş olacaktır.

Kuyruğa Eleman ekleme ve kuyruktan eleman çıkarma işlemlerinin mantığı ise Aynı şekilde eleman ekleme işlemi rear üzerinden, çıkarma işlemi ise front üzerinden yapılır.

Başlangıçta ilk eklenen eleman hem öndeki eleman hem de arkadaki eleman olur. Bu mantık üzerinden bağlı liste ile kuyruk yapısı oluşturulur.

\centerline{\includegraphics{blank.JPG}}

Öncelikli bir kuyruğa alma mekanizmasının kullanılması aşağıdaki avantajları sağlayabilir:

Uygulamaların kullanılabilirlik veya performans için öncelik belirlenmesini gerektiren iş gereksinimlerini karşılamasına olanak sağlar. Örneğin, belirli müşteri gruplarına farklı düzeylerde hizmet sunulabilir.

İşletim maliyetlerinin en aza indirilmesine yardımcı olabilir. Tek kuyruklu yaklaşımda gerekirse tüketici sayısının ölçeğini küçültebilirsiniz. Yüksek öncelikli iletiler yine ilk olarak işlenir (ancak büyük olasılıkla daha yavaş bir şekilde) ve düşük öncelikli iletiler daha uzun bir süre için ertelenebilir. Her bir kuyruk için ayrı bir tüketici havuzu içeren birden fazla ileti kuyruğu yaklaşımını uyguladıysanız düşük öncelikli kuyruklar için tüketici havuzunu küçültebilir hatta çok düşük öncelik kuyruklarda ileti olup olmadığını dinleyen tüm tüketicileri durdurarak bu kuyrukların işlemlerini askıya alabilirsiniz.

Birden fazla ileti kuyruğu yaklaşımı, iletileri işleme gereksinimlerine bağlı olarak bölümleyerek uygulama performansı ve ölçeklenebilirliğinin en üst düzeye çıkarılmasına yardımcı olabilir. Örneğin, çok önemli görevler hemen çalıştırılan alıcılar tarafından işlenecek şekilde önceliklendirilebilir. Önemi daha düşük arka plan görevleri de daha az meşgul zamanlarda çalıştırılması zamanlanan alıcılar tarafından işlenebilir.

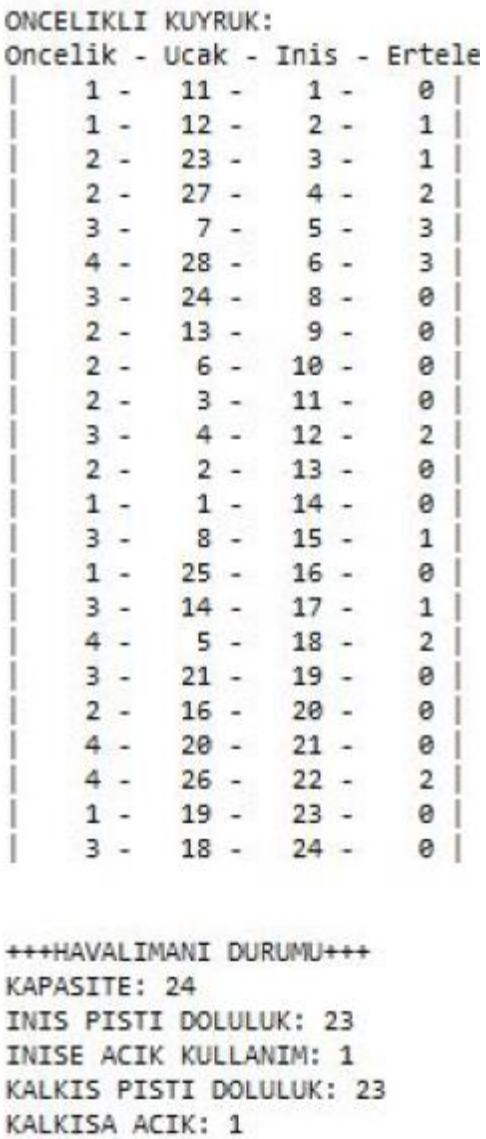
Bu düzen, aşağıdakilerin geçerli olduğu senaryolarda kullanışlıdır:

Sistem farklı önceliklere sahip birden çok görevi işlemelidir.

Farklı kullanıcı veya kiracılara farklı önceliklerle hizmet sunulması gereklidir.

SONUÇ

Uygulama, Windows işletim sistemi üzerinde Dev C++ IDE'si kullanılarak C dilinde gerçeklenmiştir. Sonuçlarına ait ekran görüntüleri;



```
ONCELIKLI KUYRUK:
Oncelik - Ucak - Inis - Ertele
|    1 -   11 -    1 -    0 |
|    1 -   12 -    2 -    1 |
|    2 -   23 -    3 -    1 |
|    2 -   27 -    4 -    2 |
|    3 -    7 -    5 -    3 |
|    4 -   28 -    6 -    3 |
|    3 -   24 -    8 -    0 |
|    2 -   13 -    9 -    0 |
|    2 -    6 -   10 -    0 |
|    2 -    3 -   11 -    0 |
|    3 -    4 -   12 -    2 |
|    2 -    2 -   13 -    0 |
|    1 -    1 -   14 -    0 |
|    3 -    8 -   15 -    1 |
|    1 -   25 -   16 -    0 |
|    3 -   14 -   17 -    1 |
|    4 -    5 -   18 -    2 |
|    3 -   21 -   19 -    0 |
|    2 -   16 -   20 -    0 |
|    4 -   20 -   21 -    0 |
|    4 -   26 -   22 -    2 |
|    1 -   19 -   23 -    0 |
|    3 -   18 -   24 -    0 |

+++HAVALIMANI DURUMU+++
KAPASITE: 24
INIS PISTI DOLULUK: 23
INISE ACIK KULLANIM: 1
KALKIS PISTI DOLULUK: 23
KALKISA ACIK: 1
```

KAYNAKLAR

https://docs.microsoft.com/tr-tr/azure/architecture/patterns/priority-queue,Erişim 10.12.2021

https://hasscript.com/908/oncelikli-kuyruk-prioty-queue-nedir,Erişim 08.12.2021

Class Priority Queue, baskent.edu.tr/~tkaracay

Öncelikli Kuyruk Dizi Gerçekleştirimi, Bozok Üniversitesi

Doğrusal Veri Yapıları, medium.com/@tolgahan.cepel

Algoritma ve Veri Yapıları İleri Seviye, BTK Akademi

draw.io

Overleaf: New to LaTeX?