

課程大綱(Agenda)

雲原生架構以及趨勢

軟體產品設計以及
軟體開發流程 DevOps
CI/CD

IaC 與 K8S觀念

DevOps 最佳實踐,
Software Development
Lifecycle

MLOps DevOps 流程

課後實作

使用語言



2~4 hours/week

GCE cloud shell
+ ansible

4~6 hours/week

Use your preferred
language

2~4 hours/week

Python

週別	日期	課程進度、內容、主題	負責處別	講師名單
1	2/16	雲原生架構與現代軟體發展趨勢(微服務以及容器化) 介紹	TSID + BSID	陳蕙猛/胡君怡/吳聲葆
2	2/23	軟體產品設計思考以及使用者需求最佳範例	TSID + BSID	Cindy/張弘明/范姜峻韋/洪挺鈞
3	3/2	雲原生開發DevOps,CI/CD 持續整合與交付部署自動化	TSID	高傳詔/林秉毅(林俊成)
4	3/9	軟體定義現代化資料中心以及高效率的資訊服務基礎建設 Infrastructure as code	ICSD	李佳壕/李青峰
5	3/16	設計以及架構建置K8s 自有叢集	ICSD	李佳壕/李青峰
6	3/23	雲原生應用架構於智慧製造軟體產品設計案例分享 (Smart Manufacture, Digital Business)	ICSD	Cindy/陳建良 蔡易陵/粘怡祥
7	3/30	半導體產業現代化資料中心實地參訪	ICSD / TSID	李佳壕/陳建良
8	4/6	Mid-term Evaluation (書面繳交; 交大校慶停課一次)	ICSD	李佳壕/李青峰

週別	日期	課程進度、內容、主題	負責處別	講師名單+(助教名單)
9	4/13	軟體持續交付佈署自動化 企業資訊系統DevOps最佳實踐- Design Phase	TSID	Joshua 趙瑞祥(鄭清斌)
10	4/20	企業資訊系統DevOps最佳實踐- Design Phas	TSID	羅子威(范競勻, 吳振愷)
11	4/27	企業資訊系統DevOps最佳實踐- Development & Testing Phase	TSID + BSID	沈冠廷/楊捷凱
12	5/4	企業資訊系統DevOps最佳實踐 – Deployment	TSID	沈冠廷/楊捷凱
13	5/11	企業資訊系統DevOps最佳實踐– Monitor and Troubleshooting	TSID	詹文志(林家民)
14	5/18	機器學習MLOps的 DevOps 流程	AAID	蘇冠華
15	5/25	機器學習MLOps的 DevOps 流程	AAID	蘇冠華
16	6/1	(Option) Final Project 準備 + 助教諮詢	TSID/BSID	
17	6/8	Final Project Presentation -1	TSID/BSID	
18	6/15	Final Project Presentation -2	TSID/BSID	

成績考核(Evaluation)

成績考核(Evaluation)

1. 出席 10% (課程回饋)
2. 隨堂測驗 20% (5題選擇 or 做答題; 當天23:59分前繳交)
3. 軟體實作 (GKE setup) + 期中發表 30% (4~6人一組)
4. 軟體實作 + 期末發表 40% (4~6人一組)

Topic 1. 講師 胡君怡 / 課程說明

1.1 講師能清楚傳遞課程內容 *

	1	2	3	4	5	
非常不同意	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	非常同意

1.2 講師具備主題專業知識 *

	1	2	3	4	5	
非常不同意	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	非常同意

1.3 講師的實務經驗能帶給您收穫 *

	1	2	3	4	5	
非常不同意	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	非常同意

1.4 講師能有效地引導我學習 *

	1	2	3	4	5	
非常不同意	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	非常同意

1.5 講師有良好控制課程時間及進度 *

	1	2	3	4	5	
非常不同意	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	非常同意

1.6 教材內容豐富且有效地呈現教學內容 *

	1	2	3	4	5	
非常不同意	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	非常同意

1.7 我對於此課程的吸收程度 *

	1	2	3	4	5	
非常不同意	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	非常同意

[返回](#) [繼續](#) [清除表單](#)

Final Project Evaluation (1/2)

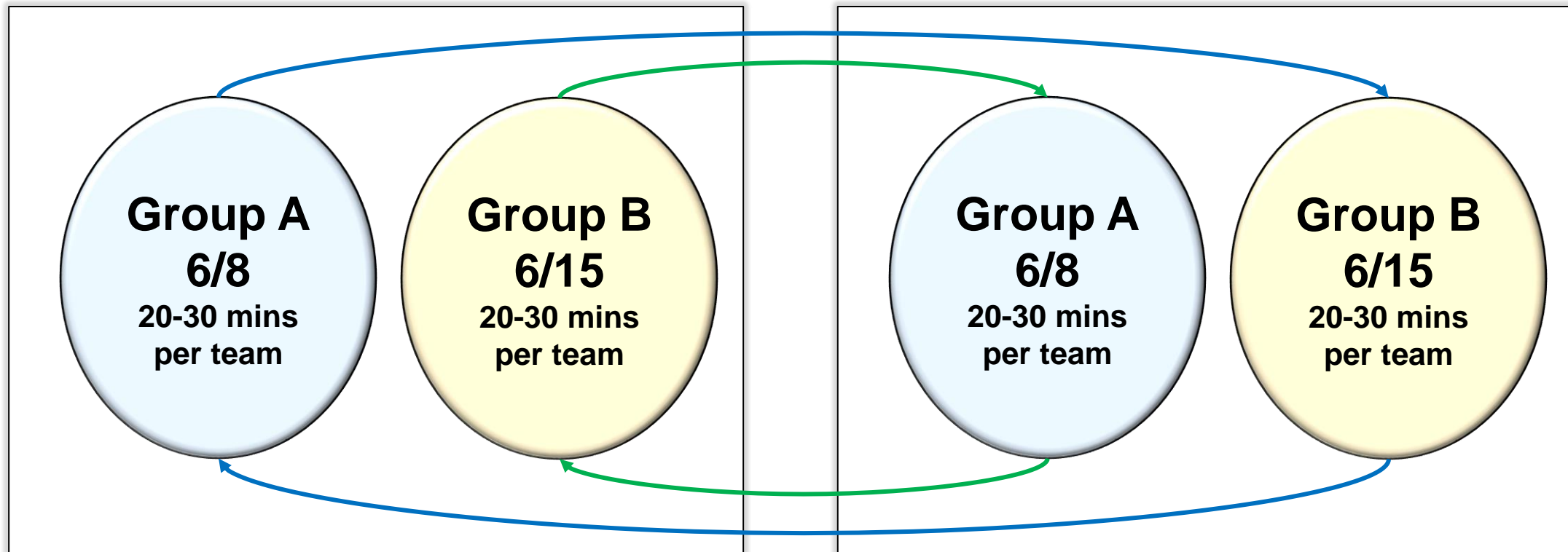
Type	Category	Score %	Deliverables & Evaluation	Remark
Basic (100%)	Application meet requirements	60%	Upload codes to Github and deploy to GKE	Provide (4/30) application codes with unit test, deployment docker image
	Testing Coverage (Junit, Pytest .. With Cobertura)			
	Application Architecture	20%	Provide PPT wi/ final presentation	Provide sample of architecture
	Code Optimization (adopting design patterns to keep code maintainable, extensible or improve reliability or faster trouble shooting) , CI/CD	20%	1) Upload codes to Github and deploy to GKE 2) Provide PPT wi/ final presentation	Enhance and optimize template codes
Option (30%)	Application Monitor (Metrics, Approach ..)	10%	Provide PPT wi/ final presentation	5/11 lectures
	Code Quality (configurable design, password encryption..) + 增加 unit test/logging, Use DB+ code contribution	20%	Provide PPT wi/ final presentation	

Final Project Evaluation (2/2)

- (Option) Final Project 準備 + 助教諮詢 at 6/1
- Github code frozen at 6/7
- Two presentations at 6/8, 6/15 and welcome to join other group at free week.

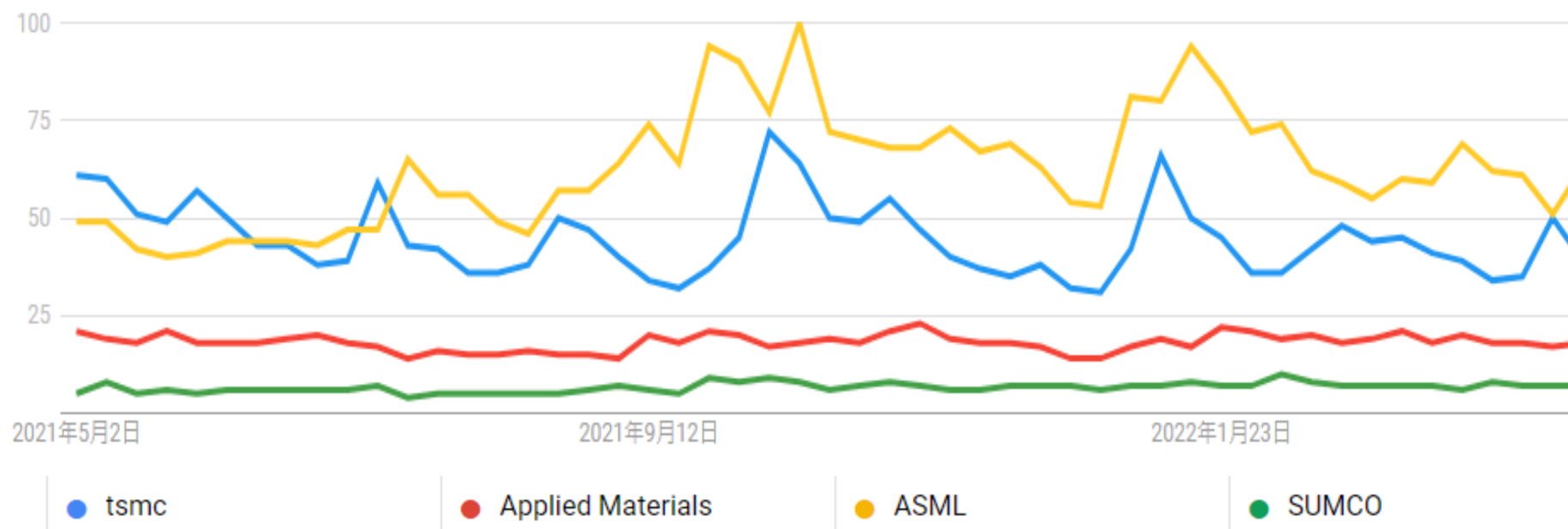
Digital Business

Smart Manufacture

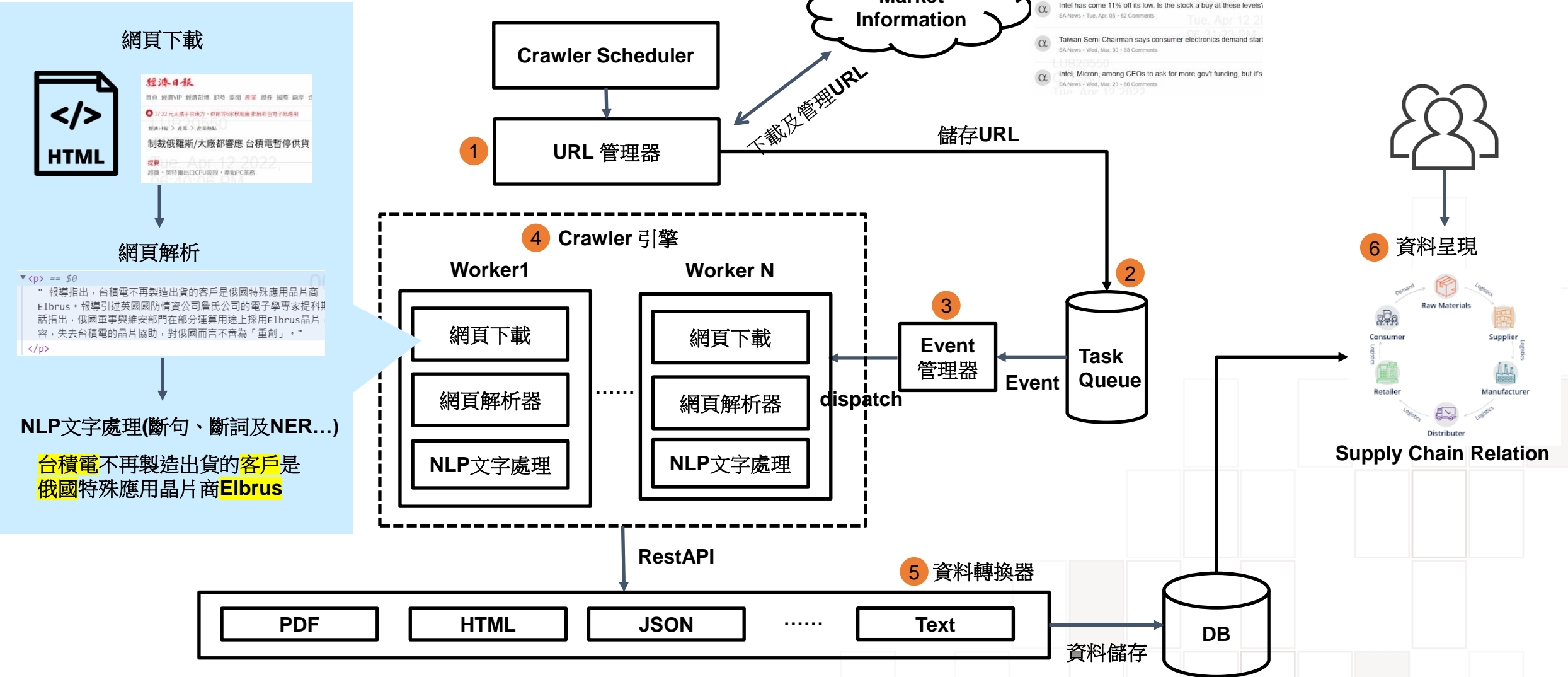


交大雲原生課程 – 期末專題 Digital Business

- 請透過網路爬蟲，找出台積電與3家供應商的網路聲量變化。
 - Applied Materials (應用材料)
 - ASML (艾司摩爾)
 - SUMCO (勝高株式會社)



Crawler Reference Architecture



交大雲原生課程 – 期末專題 Digital Business

- 講師
 - SCPM / 粘怡祥 Hugo (yhnien@tsmc.com)
 - CAPD / 蔡易陵 Rophy (yltsaize@tsmc.com)
- 助教
 - CAPD / 方子睿 (zrfang@tsmc.com)
 - SCPM / 蔡仲庭 (cttsait@tsmc.com)
 - SCPM / 徐希平 (hphsuj@tsmc.com)



期末專題範例程式

程式使用情境

1. 用程式去呼叫Google網頁伺服器，然後將顯示出來的URL，抓出來。(支援換頁及限縮搜尋時間)
2. 對我們要的URL，進行萃取，取得文字內容。
3. 將文字內容，進行word count計算。
4. 將word count結果儲存到Excel，以利分析之使用。

Step1 透過程式呼叫Google網頁伺服器，並把URL抓出來

1. 在Google上輸入關鍵字，然後將顯示出來的URL，抓出來。(支援換頁及限縮收尋時間)

使用google_search 這支Function

- 參數使用 timeline：搜尋時間 => 參數可以參考qdr:h (past hour), qdr:d (past day), qdr:w (past week), qdr:m (past month), qdr:y (past year)
- 參數使用 page：換頁

```
def google_search(self, query, timeline='', page='0'):
    url = self.url + query + '&tbs={timeline}&start={page}'.format(timeline=timeline, page=page)
    print('[Check][URL] URL : {url}'.format(url=url))
    response = self.get_source(self.url + query)
    return self.parse_googleResults(response)
```

這邊用到get_source及parse_googleResults，兩支Function，為了解析Google Search網路資源，並將URL/標題/內文，找出來。

```
def get_source(self, url):
    try:
        session = HTMLSession()
        response = session.get(url)
        return response
    except requests.exceptions.RequestException as e:
        print(e)
def parse_googleResults(self, response):
    css_identifier_result = "tF2Cxc"
    css_identifier_title = "h3"
    css_identifier_link = "yuRUbf"
    css_identifier_text = "VwiC3b"
    soup = BeautifulSoup(response.text, 'html.parser')
    results = soup.findAll("div", {"class": css_identifier_result})
    output = []
    for result in results:
        item = {
            'title': result.find(css_identifier_title).get_text(),
            'link': result.find("div", {"class": css_identifier_link}).find(href=True)['href'],
            'text': result.find("div", {"class": css_identifier_text}).get_text()
        }
        output.append(item)
    return output
```

```
query = "TSMC ASML"
crawler = GoogleCrawler()
results = crawler.google_search(query, 'qdr:w', '10')
print(results[:3])
```

```
[Check][URL] URL : https://www.google.com/search?q=TSMC ASML
[{'title': 'Better Semiconductor Stock: ASML vs. TSMC | The
etter-semiconductor-stock-asml-vs-tsmc/', 'text': '21 Apr 20
r than ASML's forward P/E ratio of 45. TSMC's forward divide
der to step up TSMC competition - Taipei ...', 'link': 'http
t': '20 Jan 2022 — Intel Corp yesterday said it has placed
industry's first TWINSKAN EXE:\xa0...'}, {'title': 'Chip mac
s://www.cnbc.com/2021/10/12/chip-machine-maker-asml-will-gro
L provides chip makers with essential hardware ... likes of
```

Step2 對我們要的URL，將內文萃取下來

2. 對我們要的URL，進行萃取，取得文字內容。

- 使用get_source 這支Function，取得網路資源後，
- 使用html_parser 這支Function，將網路資源進行解析。
- 使用html_getText 這支Function，將我要的區塊p tag的文字取出來。

```
def get_source(self,url):
    try:
        session = HTMLSession()
        response = session.get(url)
        return response
    except requests.exceptions.RequestException as e:
        print(e)
def html_parser(self,htmlText):
    soup = BeautifulSoup(htmlText, 'html.parser')
    return soup
def html_getText(self,soup):
    original_text = ''
    for el in soup.find_all('p'):
        original_text += ''.join(el.find_all(text=True))
    return original_text
```

```
Target_URL = 'https://taipeitimes.com/News/biz/archives/2022/01/20/2003771688'
response = crawler.get_source(Target_URL)
soup = crawler.html_parser(response.text)
original_text = crawler.html_getText(soup)
print(original_text[:100])
```

Intel Corp yesterday said it has placed its first order with ASML Holding NV to

Step3 將文字內容，進行word count計算

3. 將文字內容，進行word count計算

使用word_count function，將一篇文章內容，進行各文字的count個數

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
def word_count(self, text):
    counts = dict()
    stop_words = set(stopwords.words('english'))
    words = word_tokenize(text)
    #words = text.replace(',','').split()
    for word in words:
        if word not in stop_words:
            if word in counts:
                counts[word] += 1
            else:
                counts[word] = 1
    return counts
```

NLTK文字處理:

- 去除冗餘字
- 斷詞斷句
-

<https://www.nltk.org/>

```
In [209]: result_wordcount = crawler.word_count(original_text)
result_wordcount
```

```
Out[209]: {'Intel': 5,
'Corp': 1,
'yesterday': 2,
'said': 14,
'it': 8,
'has': 3,
'placed': 1,
'its': 7,
'first': 7,
```

Step4 將計算結果，存到Excel中

4. 將聲量結果儲存到Excel

使用get_wordcount_json function 濾掉不要的word count dict, 取自己要的(whitelist)

同時利用jsonarray_toexcel function 將結果，落地於Excel，以做聲量分析

```
def get_wordcount_json(self, whitelist, dict_data):
    data_array = []
    for i in whitelist:
        json_data = {
            'Date': 'Week1',
            'Company': i,
            'Count': dict_data[i]
        }
        data_array.append(json_data)
    return data_array

def jsonarray_toexcel(self, data_array):
    df = pd.DataFrame(data=data_array)
    df.to_excel('result.xlsx', index=False)
    return
```

	A	B	C
1	Company	Count	Date
2	ASML	6	Week1
3	Intel	5	Week1
4			

```
: whitelist = ['ASML', 'Intel']
end_result = crawler.get_wordcount_json(whitelist, result_wordcount)
print(end_result)
crawler.jsonarray_toexcel(end_result)
print('Excel is OK')
```

```
[{'Date': 'Week1', 'Company': 'ASML', 'Count': 6}, {'Date': 'Week1', 'Company': 'Intel', 'Count': 5}]
Excel is OK
```

Build Container Image of Sample Crawler

- Sample Dockerfile is also included in Source code zip

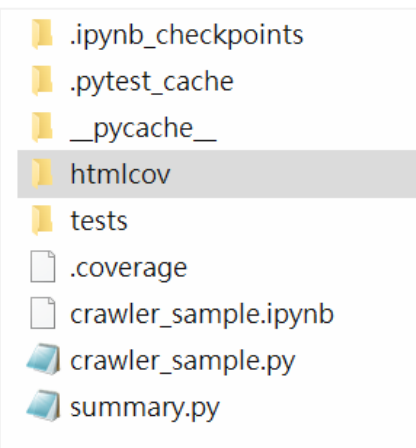
```
1 FROM python:3.6.15-slim
2 COPY . /sample_crawler
3 WORKDIR /sample_crawler
4 RUN pip install -r ./requirements.txt
5 CMD ["python", "crawler_sample.py"]
```

Build Crawler Image

```
docker build -t mycralwer:test .
```


Pytest

- 將測試程式先放在指定的**tests**資料夾
- 下指令: `python -m pytest . tests --doctest-modules --junitxml=test-results.xml --cov-config=.coveragerc --cov=. --cov-report=html`



htmlcov: coverage report位址
tests: unit test

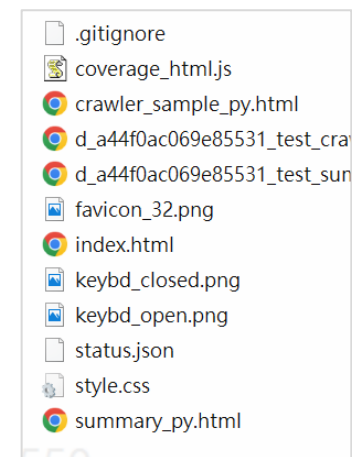
```
PS C:\Users\bapadmin\Desktop\sample_crawler> python -m pytest . tests --doctest-modules --cov-config=.coveragerc --cov=. --cov-report=html
===== test session starts =====
platform win32 -- Python 3.6.5, pytest-3.10.1, py-1.11.0, pluggy-1.0.0
rootdir: C:\Users\bapadmin\Desktop\sample_crawler, inifile:
plugins: arraydiff-0.2, cov-3.0.0, doctestplus-0.1.3, openfiles-0.3.0, remotedata-0.2.1
collected 6 items

tests\test_crawler.py .....
tests\test_summary.py .

[ 83%]
[100%]
```

Pytest執行過程

在htmlcov folder找到對應.html可以看到程式碼的Coverage狀況
Ex: htmlcov/crawler_sample_py.html



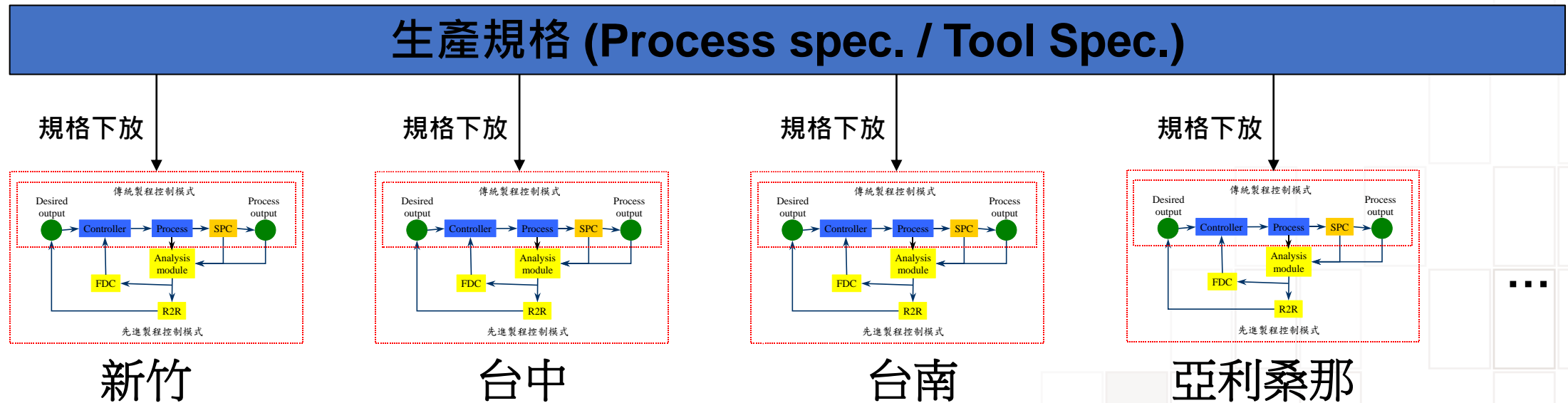


Unleash Innovation

期末實作 – Smart Manufacturing

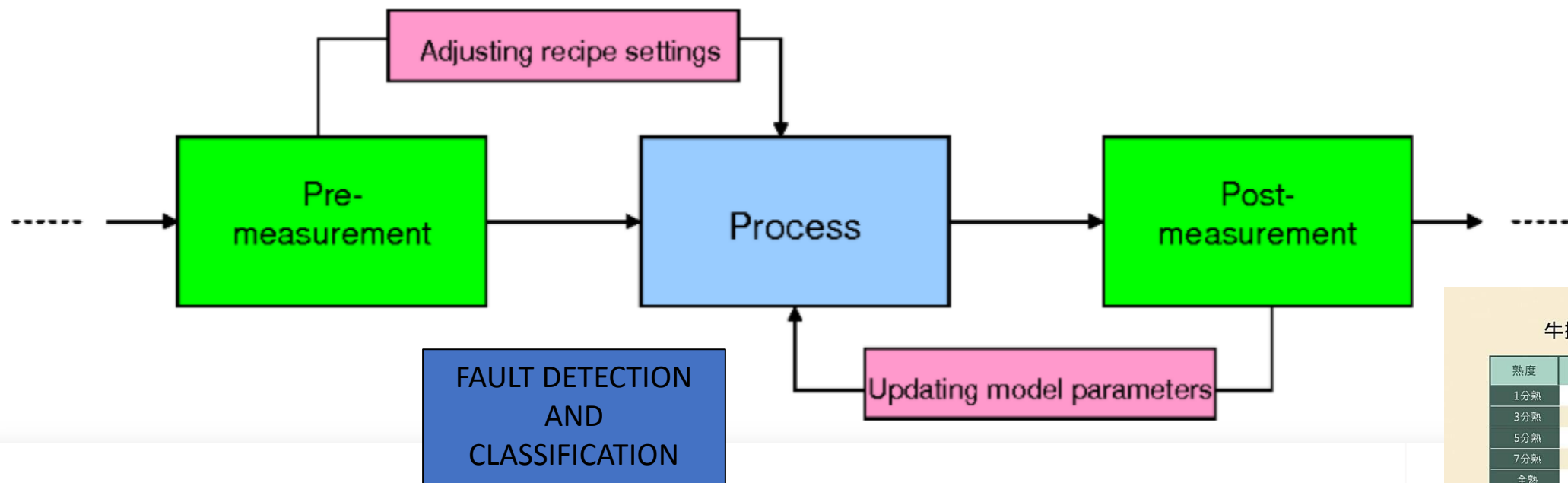
課堂實作 – Smart Manufacturing

- 為了達到不同廠區生產出來的產品規格一致, 產品規格由中央訂定, 規格下放到各個獨立生產的廠區
- 廠區的生產穩定由APC (Advanced Process Control) 控管 (不同產品的Spec. 在各個不同獨立廠區都相同)



...

課堂實作 – Smart Manufacturing



水分比例
厚度



溫度
時間



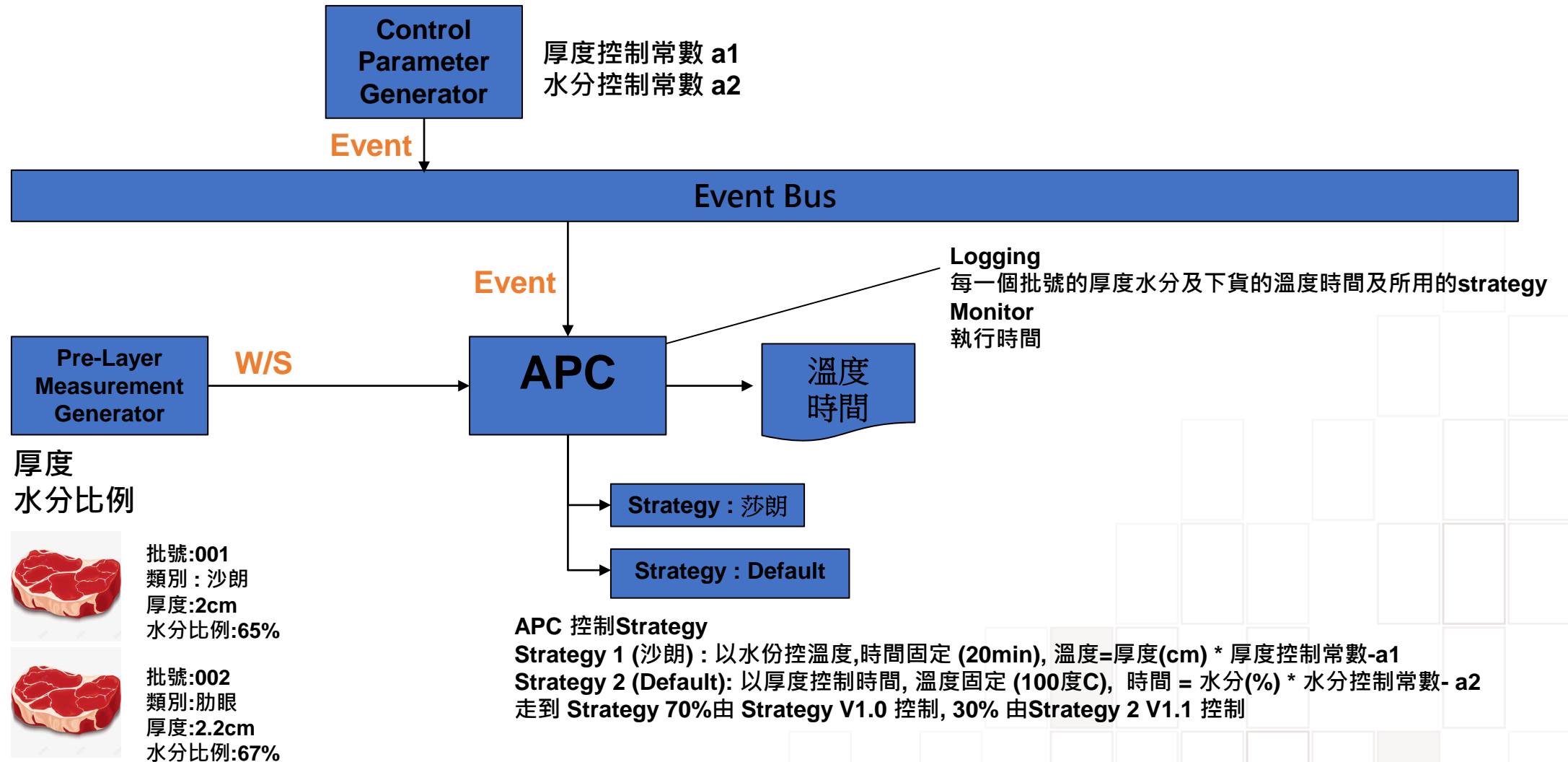
核心溫度

食力 foodNEXT
牛非常見的熟度判斷基準

熟度	切面顏色	中心溫度
1分熟	90%血紅色	45°C
3分熟	60%血紅色	50°C
5分熟	30%血紅色	55°C
7分熟	粉紅色	65°C
全熟	灰色	70°C

內容來源 = 《專業肉舖的牛肉料理教本》· 幸福文化出版

課堂實作 – Smart Manufacturing Reference Architecture



交大雲原生課程 – 期末專題 Smart Manufacturing

- 講師

- 羅子威 (twloc@tsmc.com)
- 趙瑞祥 (jhchao@tsmc.com)
- 沈冠廷 (gtshen@tsmc.com)
- 楊捷凱 (jkyangc@tsmc.com)

- 助教

- 鄭清斌 (cpchengb@tsmc.com)
- 范競勻 (cyfanj@tsmc.com)
- 吳振愷 (ckwum@tsmc.com)