

# DISPY: DISTRIBUTED AND PARALLEL COMPUTING WITH/FOR PYTHON

Presenter: Jing Ge



# dispy Setup

---

- Official documentation

- <http://dispy.sourceforge.net/index.html>

- Installation and Setup

- *Make sure each node of your cluster is in a same network*

- *Start dispynode.py on each node:*

- `python3 dispynode.py -d --clean -i <ip_addr>`

- <http://dispy.sourceforge.net/dispynode.html>

- *Start dispyscheduler.py (shared execution):*

- `python3 dispyscheduler.py -d - --clean -i <ip_addr>`

- <http://dispy.sourceforge.net/dispyscheduler.html>

# dispy Code Example

---

```
# simple program that distributes 'compute' function' to each node running 'dispynode'  
def compute(n):  
    import time, socket  
    time.sleep(n)  
    host = socket.gethostname()  
    return (host, n)
```

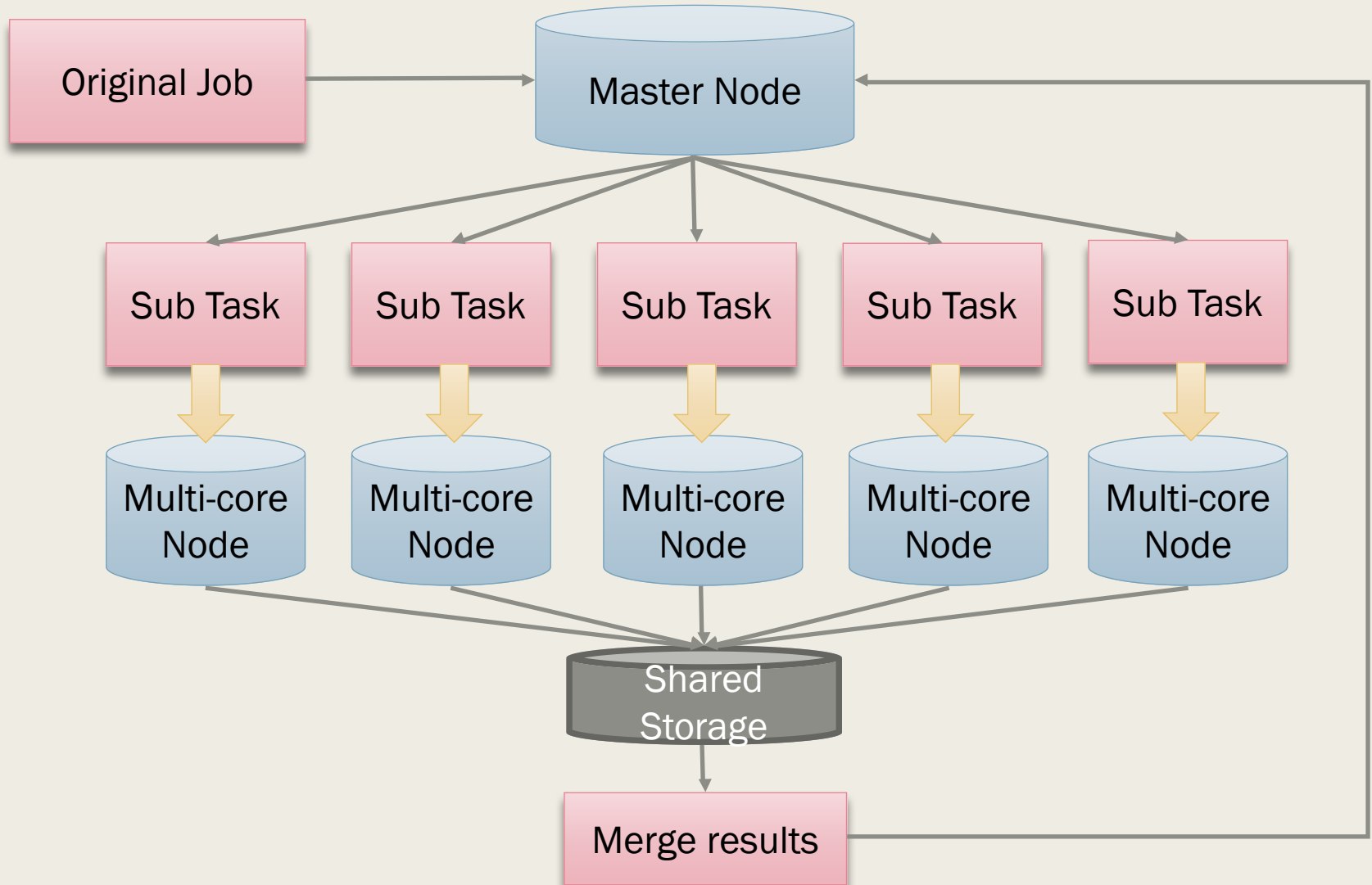
*Note: import module in individual function.*

# dispy Code Example

```
if __name__ == '__main__':
    import dispy, random

    cluster = dispy.JobCluster(compute)
    jobs = []
    for i in range(10):
        # schedule execution of 'compute' on a node (running 'dispynode')
        # with a parameter (random number in this case)
        job = cluster.submit(random.randint(5, 20))
        job.id = i # optionally associate an ID to job (if needed later)
        jobs.append(job)
    # cluster.wait() # wait for all scheduled jobs to finish
    for job in jobs:
        host, n = job() # waits for job to finish and returns results
        print('%s executed job %s at %s with %s' % (host, job.id, job.start_time, n))
        # other fields of 'job' that may be useful:
        # print(job.stdout, job.stderr, job.exception, job.ip_addr, job.start_time,
        job.end_time)
    cluster.print_status()
```

# dispy Usage In KnowEnG



# dispy Usage In KnowEnG

---

- knpackage.distributed\_computing\_utils
  - [\*distributed\\_computing\\_utils.py\*](#)
- Example usage in Samples\_Clustering\_Pipeline:
  - [\*sample\\_clustering\\_toolbox.py\*](#)
- Performance testing:
  - [\*knoweng\\_performance\\_testing\*](#)

Thank You For Watching!