China Contributions to SKA-SDP

--Perspectives and Progresses of China SDP Consortium for SKA Challenges

Yongxin ZHU
Shanghai Jiao Tong University, China

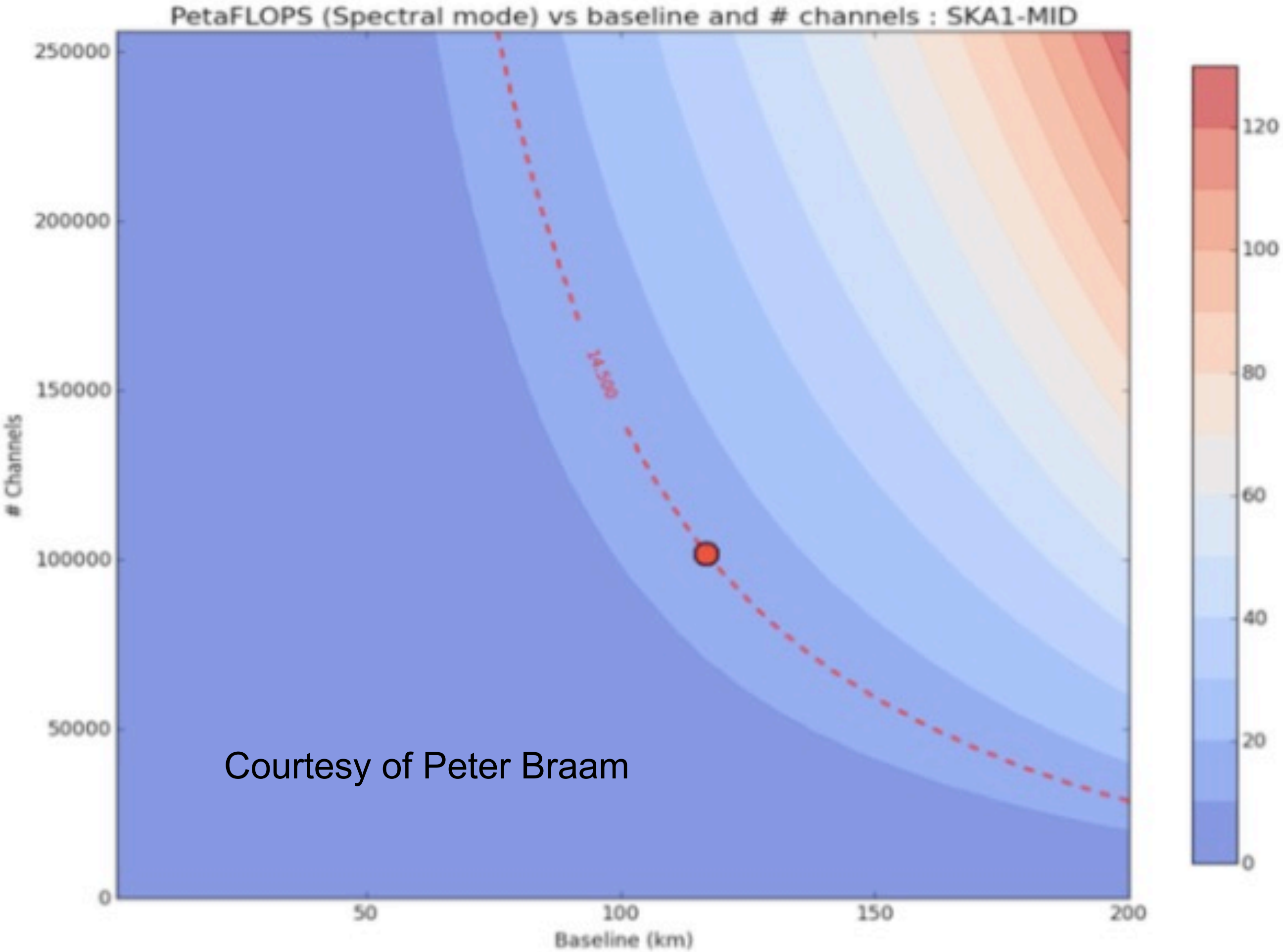
zhuyongxin@sjtu.edu.cn

SKA Engineering Meeting · SDP Consortium Meeting (12-18 June 2017, Rotterdam, Netherlands)

Background: Sustainable performance & power is more challenging than expected

Target: Sustainable Pflops vs #channels @ baseline length

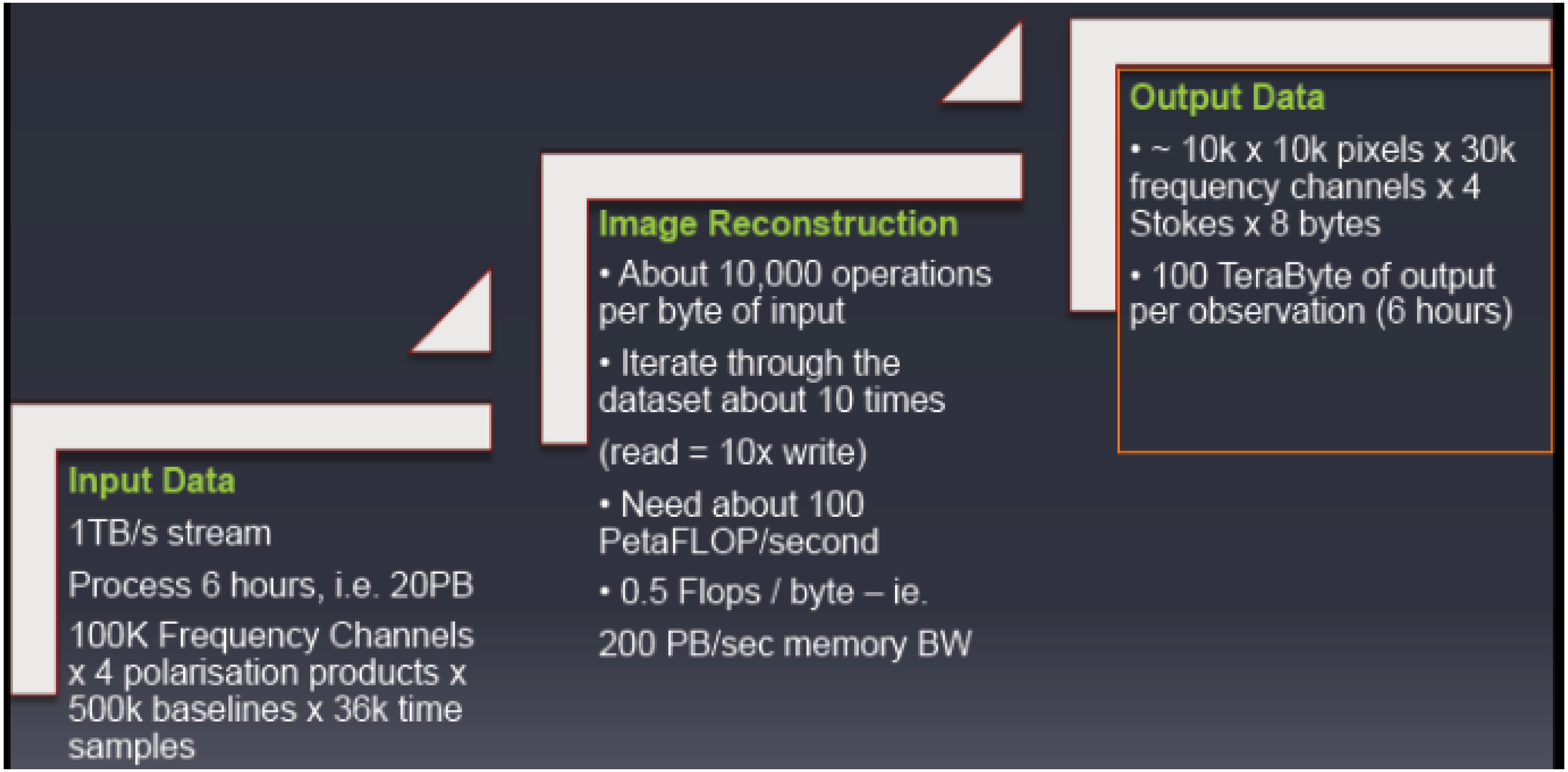
8x of top super-computer performance



1/3 of top super-computer power

Super-computer's efficiency for practical applications is less than 10% of its peak performance

Background: problem size grows much worse with data flow



Courtesy of Peter Braam

Tier1: ingest **Tier2: processing** **Tier3: archive**

Memory bandwidth and storage capacity add more problem dimensions

Outline

I. Overview of SKA-SDP China Consortium
II. Progress of China SDP Consortium
III. Perspectives of China SDP Consortium for SKA Challenges
IV. Future Work

Overview of SKA-SDP China Consortium

SKA-SDP China Consortium

Founded in Jan. 2013

国家天文台

CETC 32

中科院计算所

国家数字程控交换中心

上海交通大学

昆明理工

复旦大学

河南中英联合实验室
Sino-UK Joint lab of Henan Prov.

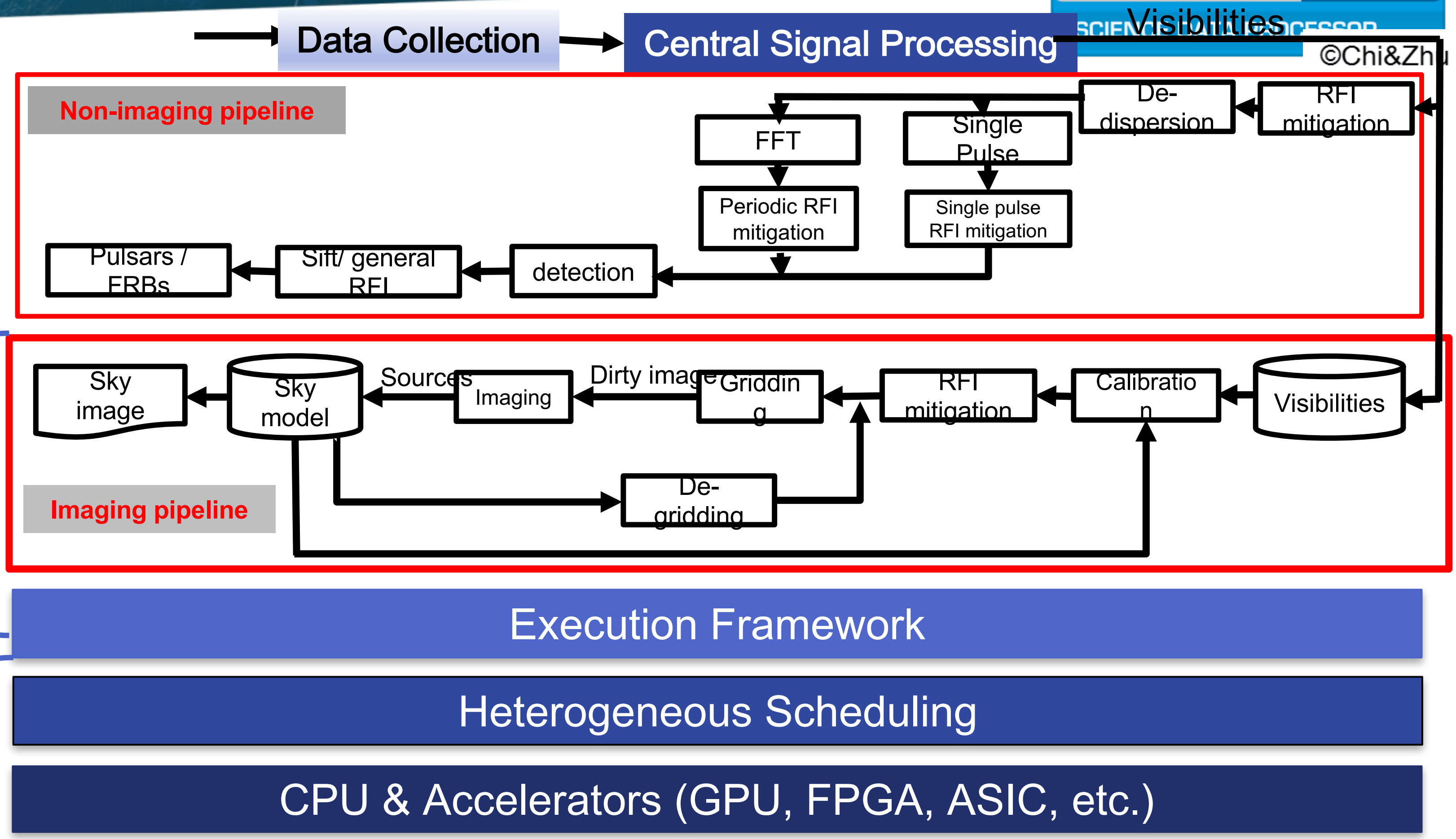
上海天文台
Shanghai A.O.

Total researchers: 77
Faculty & Eng: 27
Students: 50

Sponser :
Shanghai Hongshen Information Technology Ltd.

Beijing Bitmain Ltd.

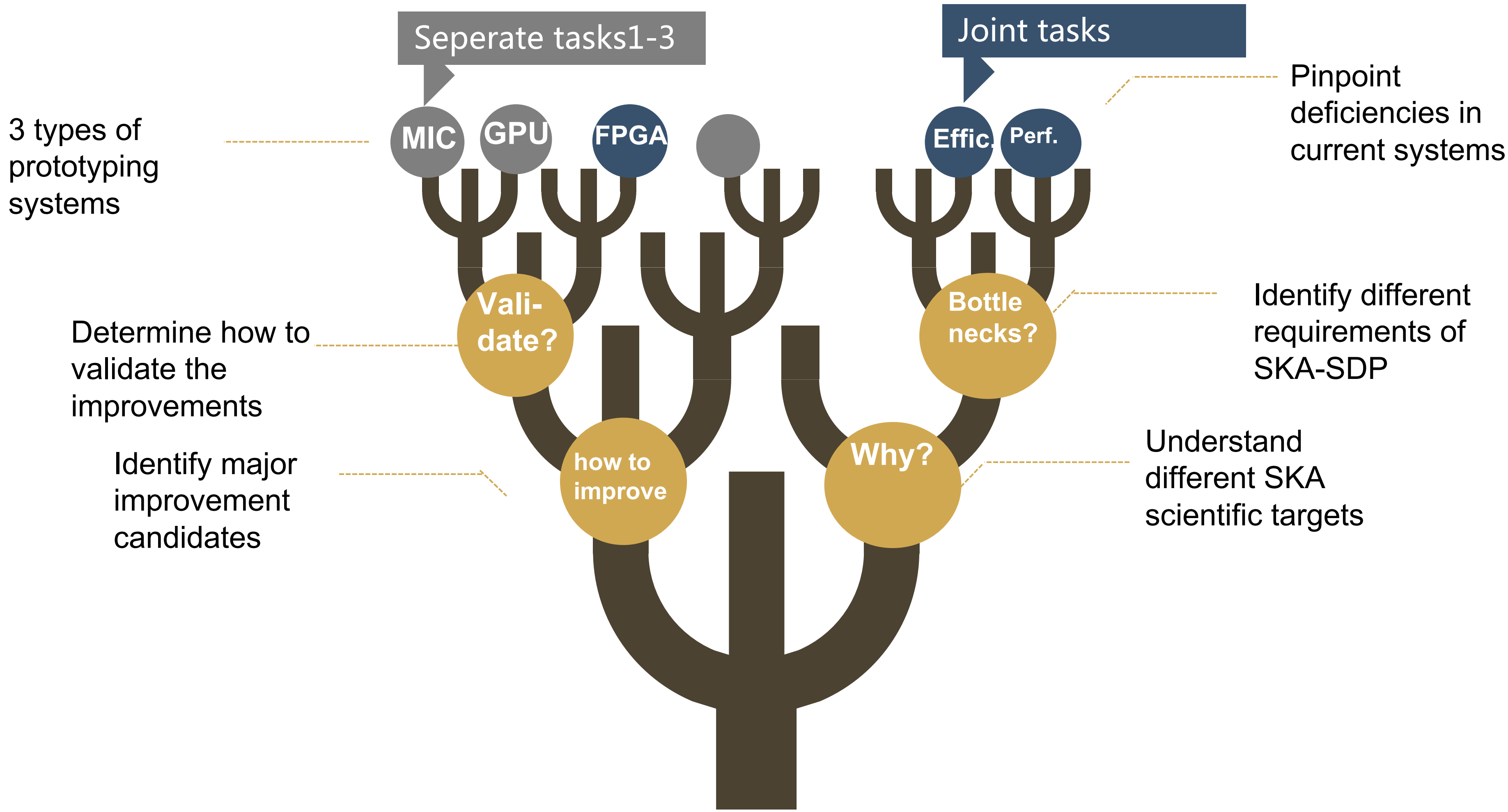
Overview of SKA-SDP China Consortium



Chinese MoST Project on Critical Technologies of SKA-SDP

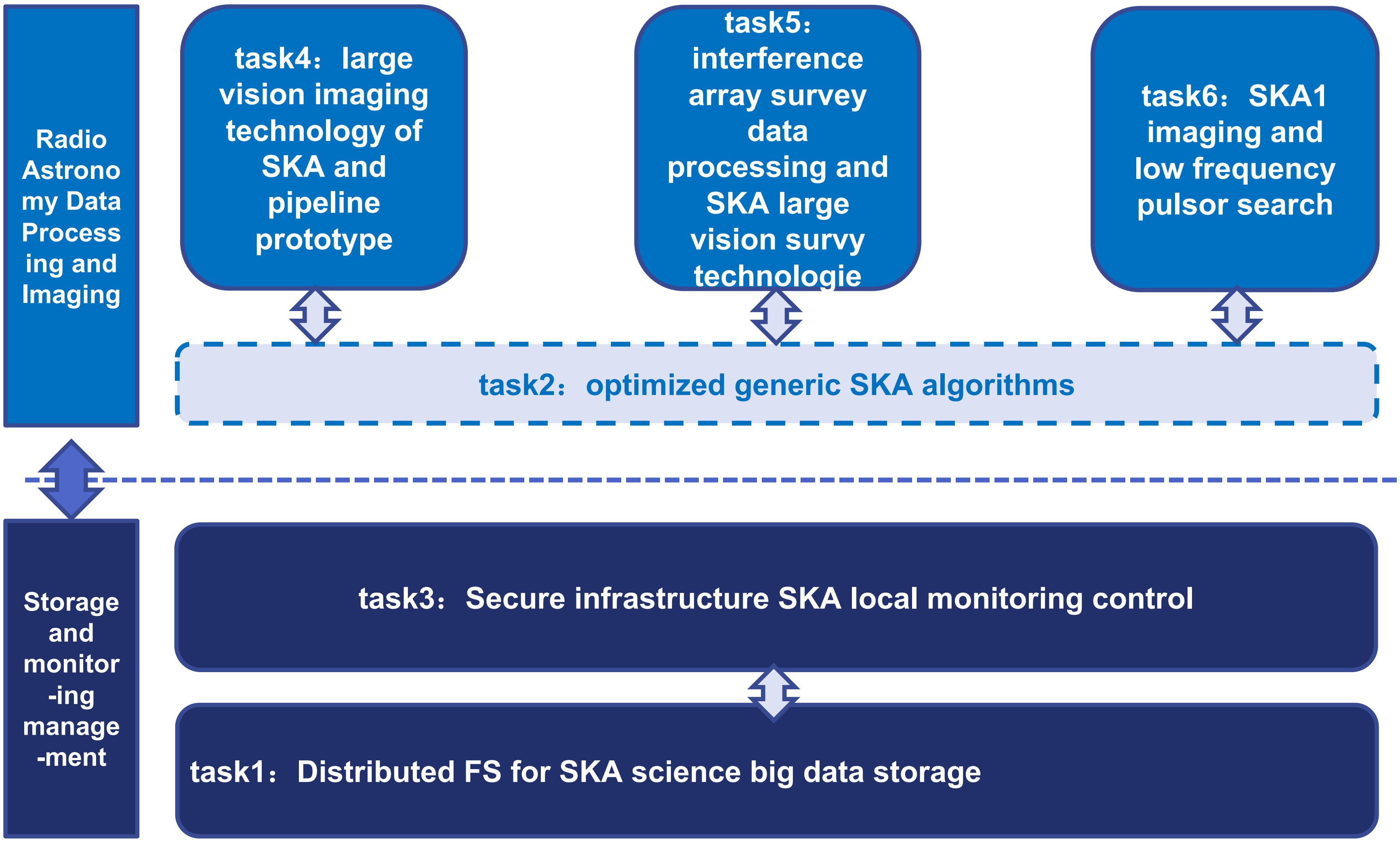
Chinese MoST Project on Prototyping Platform of SKA-SDP

Tasks of China MoST project on SKA-SDP platform prototyping



Existing supercomputers cannot meet the requirements of SKA

Tasks of China MoST project on critical technologies



Outline

I. Overview of SKA-SDP China Consortium
II. Progress of China SDP Consortium
III. Perspectives of China SDP Consortium for SKA Challenges
IV. Future Work

Shanghai Jiao Tong University (SJTU)

- Leading Organization of PRC Consortium
  - Major contact of PRC Consortium
  - PI of China MoST project on prototyping platform
- Design for PDR, Delta-PDR, Product Tree and Sprint Tasks
  - PDR
    - Compute platform: Hardware alternatives and Scheduler Software
    - Local Monitor Control: Control node and Master Controller
  - Product Tree Analysis
    - Owning 4 Tasks: Scheduler, LMC
- Prototyping for Scheduler, Hardware and LMC
  - Data dependency aware scheduler prototyping based on CloudSim
  - Variable Precision FFT prototyping based on FPGA in mimicry computer
  - Experiments on Computer Integrity and Network Control

SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PROCESSOR

✓Ownership

- TSK-8A: Scheduling model, Batch scheduling
- TSK-17: System Scheduling
- Prod_Tree PT-113: Batch Scheduler
- Prod_Tree PT-422: Event Monitoring & Logging
- Prod_Tree PT-423: EM Interface Library
- Prod_Tree PT-425: EM Log Manager
- Prod_Tree PT-426: EM Data Collector
- Prod_Tree PT-401: LMC Control Node

✓Prototyping, Profiling and Benchmarking

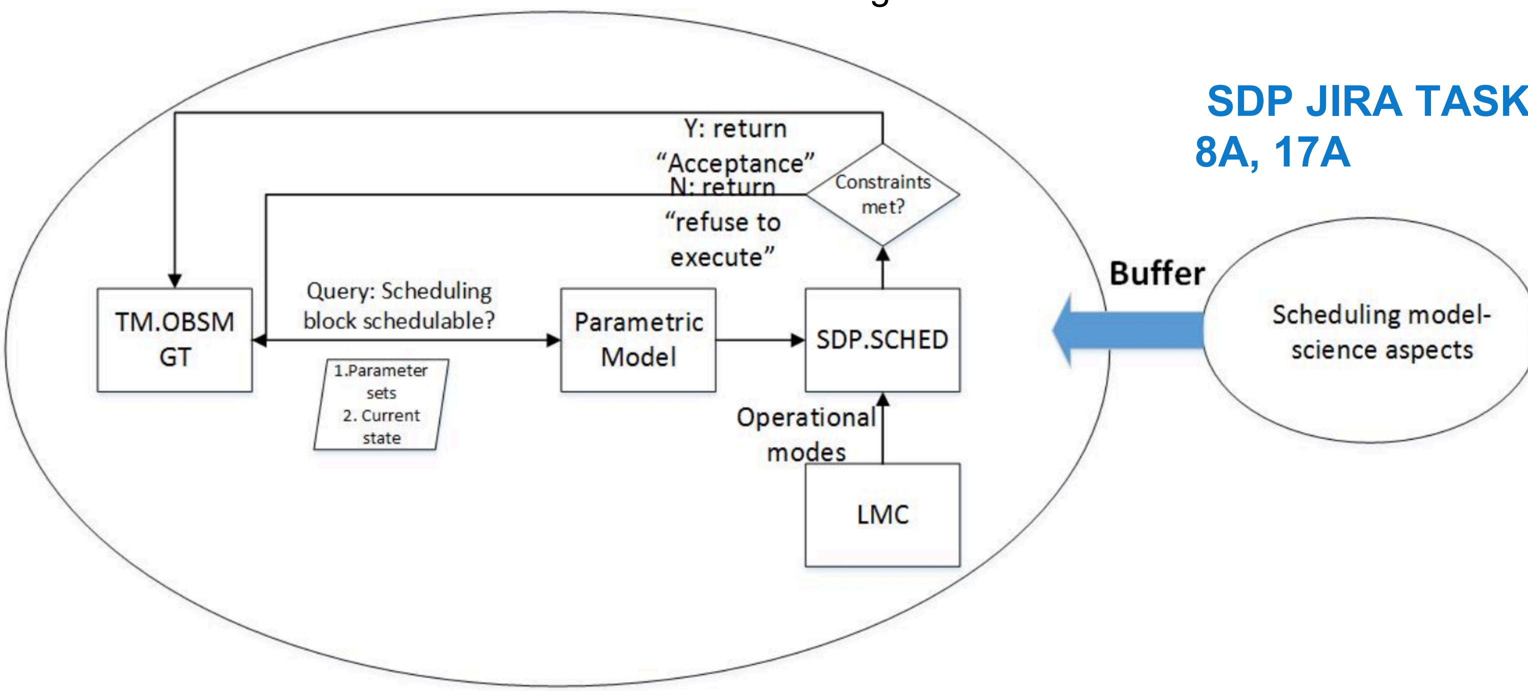
- Heterogeneous Execution Framework
- SKA Key Algorithms Acceleration: FFT, Gridding, Convolution (In process)
- Data Dependency Aware Computation Platform Scheduling
- High performance Floating-Point Unit: Unum Floating-Point Arithmetic (Variable Precision)

SJTU subtask 1: Scheduling Model

SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PROCESSOR

Scheduling Model Coordination: matching observation plan & SDP resources

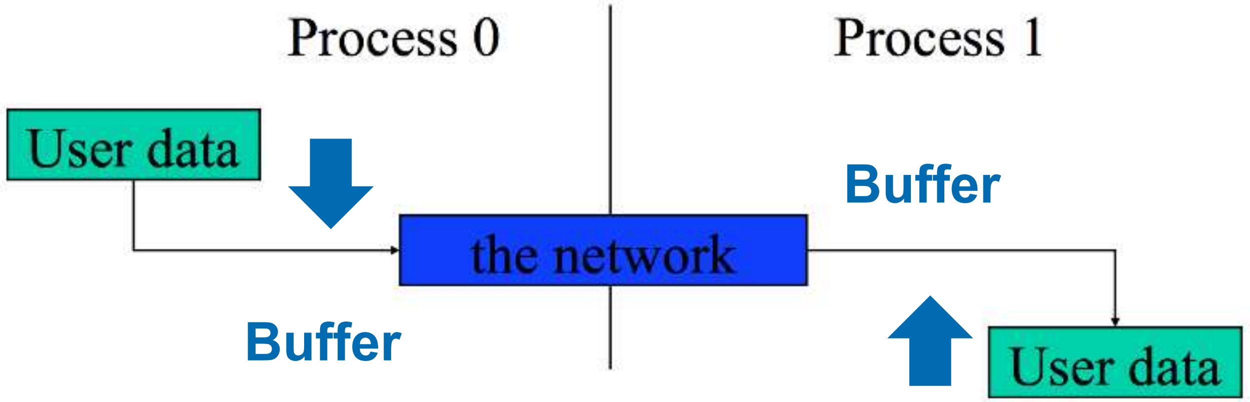
- Interactions among SDP.SCHED, SDP.LMC and TM's planner
- Predict computation and storage requirement from parametric model;
- Determine feasibility of an observation task by scheduling requirements and resources
- Coordination considerations: buffer to host incoming data of an observation



SDP JIRA TASKS 8A, 17A

SJTU subtask 1: Scheduling Model

A verification case: simulation of buffer with MPI point-to-point Communication



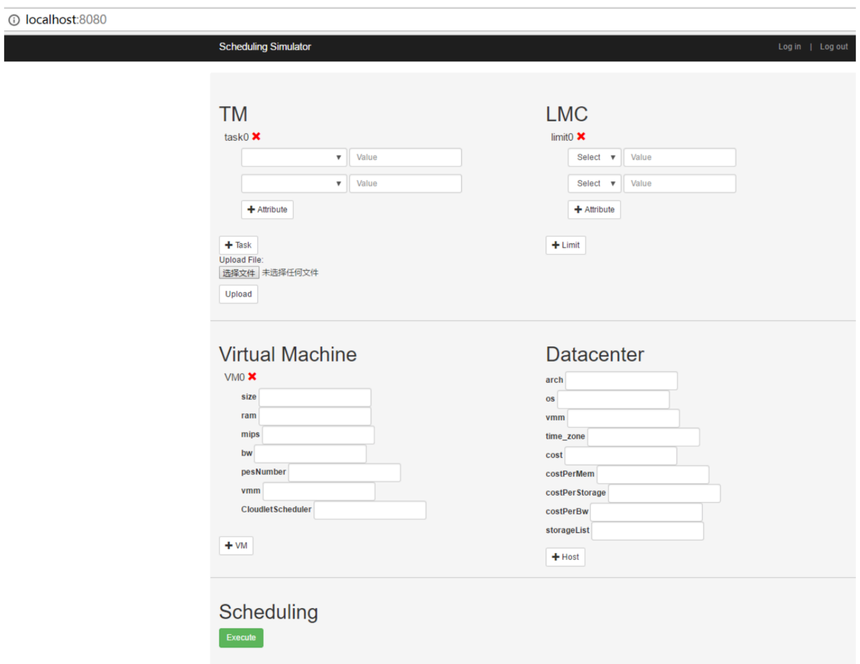
- Message : task size & buffer address
- Blocking send calls
- Three steps
  1. Send message in buffer
  2. Receive message in buffer
  3. Received successfully

```
Process 0 of 2
0 sending 'task size: 2880*17=48960 '
task size: 2880*17=48960 Process 1 of 2
1 receiving
0 receiving
0 received 'task size: 2880*17=48960 '
1 received 'task size: 2880*17=48960 '
1 sent 'task size: 2880*17=48960 '
```

SJTU subtask 1: Scheduling Model

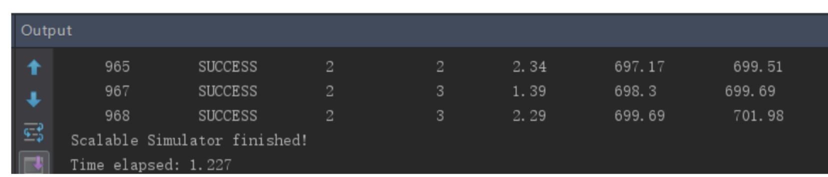
A case of public verification: GUI Wrapper of the SDP.SCHED Prototype

- Interface between webpage and scheduler: JavaScript and Servlet
- Fill in the form on the webpage / upload JSON file



root page

upload task file



Scheduled results log

SJTU subtask2: LMC impact on System Scheduling

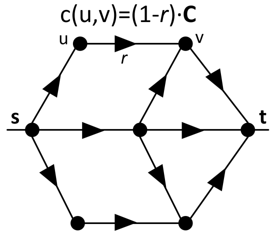
Specified interface parameters between Telescope Management's observation planner (SKAO headquarter) and SDP's Local Monitoring Control.
(TSK-17 T8A, PT-425, PT-426, SDPLMC-5, SDPLMC-6)

1. Estimation on feasible data flow in uncertainty network

- **Network throughput monitoring** is an important issue in LMC. Early warning of network status offers critical information to other tasks of SDP like scheduling.
- **Package loss** happens very often over the fiber and data transmission network in SKA.
- **The maximum capacity :** To provide an efficient scheme of pre-warning system, we introduce the max-flow problem to our monitoring to estimate the maximum capacity that a network can bear.

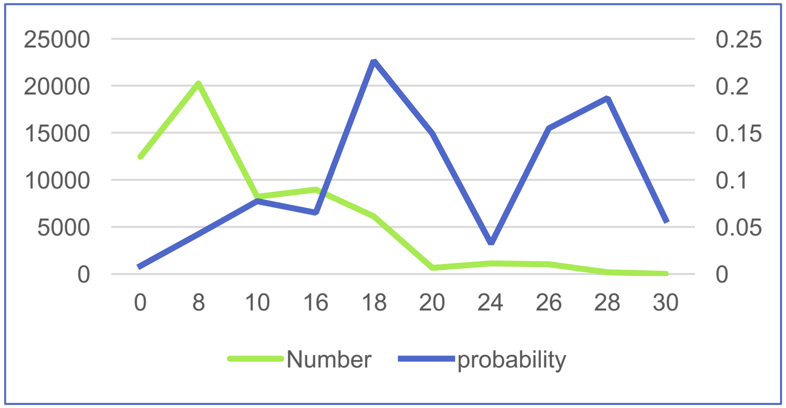
Example:

- **The packet loss rate of a link is r.** The capacity of a link can be considered as $(1-r)\cdot C$, where C denotes the original capacity of the link.
- **The capacity of each edge** and corresponding probability are showed in table:



Capacity of each edge	Probability
0	0.1
8	0.2
10	0.7

Two or three data network capacities have relatively high probabilities but few cases !



SJTU subtask2: LMC impact on System Scheduling

SCIENCE DATA PROCESSOR

Specified interface parameters between Telescope Management's observation planner (SKAO headquarter) and SDP's Local Monitoring Control

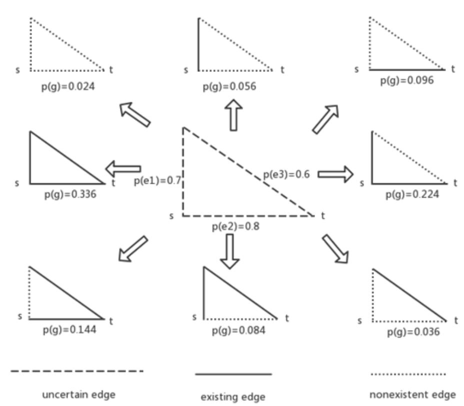
(TSK-17 T8A, PT-425, PT-426, SDPLMC-5, SDPLMC-6)

2. Detection on Uncertain Device Connectivity with Low Cost over internet of things

- **The connectivity detection between devices** is a fundamental problem in the network, and most of the existing works are based on the deterministic network, which ignores the unstable and uncertain nature of the network in the real world.
- **Uncertain device graph:** In order to overcome such limitation, we models the network as an uncertain graph, which means each device e exists independently in the network with some **probability p(e).** For a pair of nodes **s** and **t** in the network, to check whether they are connected or not, it is necessary to detect whether some links of the device network exist or not. And **the cost c(e)** is required to detect the presence or absence of the link **e**.
- **The minimum cost expectation strategy:** Our objective is to propose a detection strategy for a pair of nodes **s** and **t** in the uncertain device network, so that it has the minimum cost expectation under the premise of detecting **s-t** connectivity.

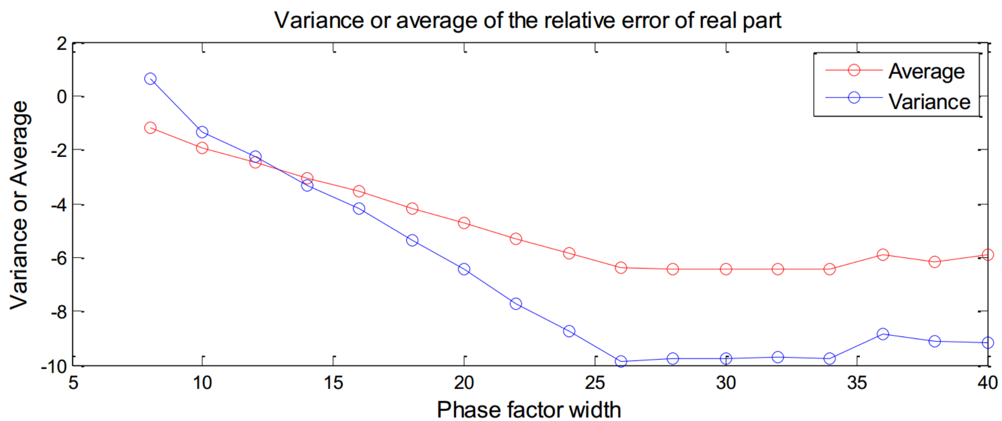
Example.

- Fig1 is **an example of an uncertain network with 3 devices(edges)** and its eight possible underlying graphs. The existence probability of each edge p(e) is labeled beside it.
- We consider what is the best detection strategy with **low cost expectation** in the next step

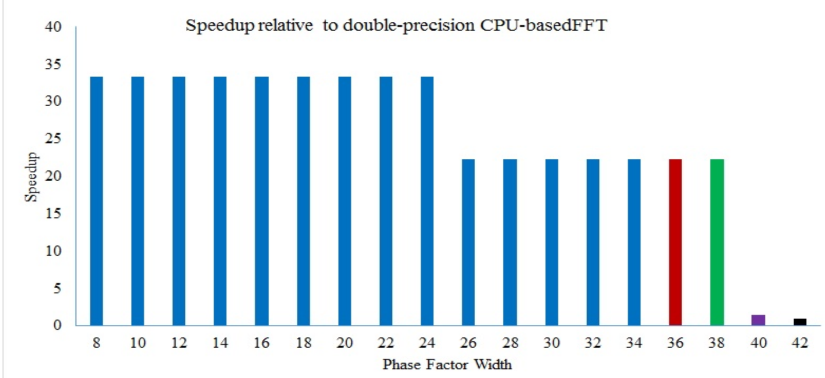


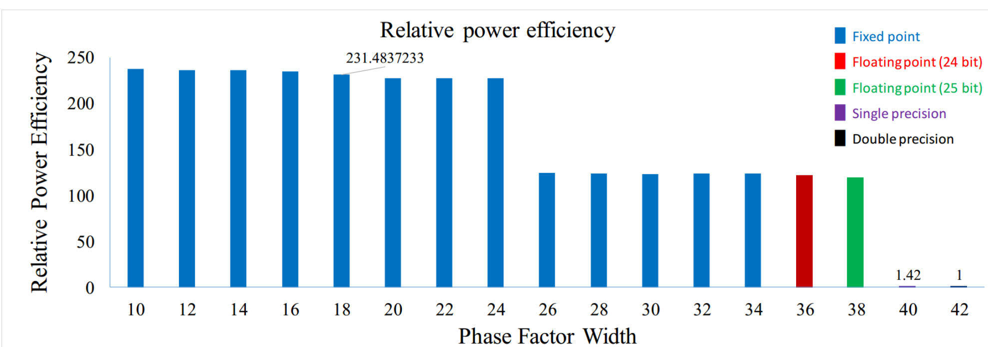
SJTU subtask 3: FFT Algorithm Implementation On FPGA

$$AvgError_{Re} = \lg\left(\frac{1}{N}\sum_{n=0}^{N-1}\left|\frac{Re_{fixed} - Re_{double}}{Re_{double}}\right|\right) \qquad VarError_{Re} = \lg\left(\frac{1}{N}\sum_{n=0}^{N-1}\left|Re_{fixed} - AvgError_{Re}\right|^2\right)$$



Speedup over 30x



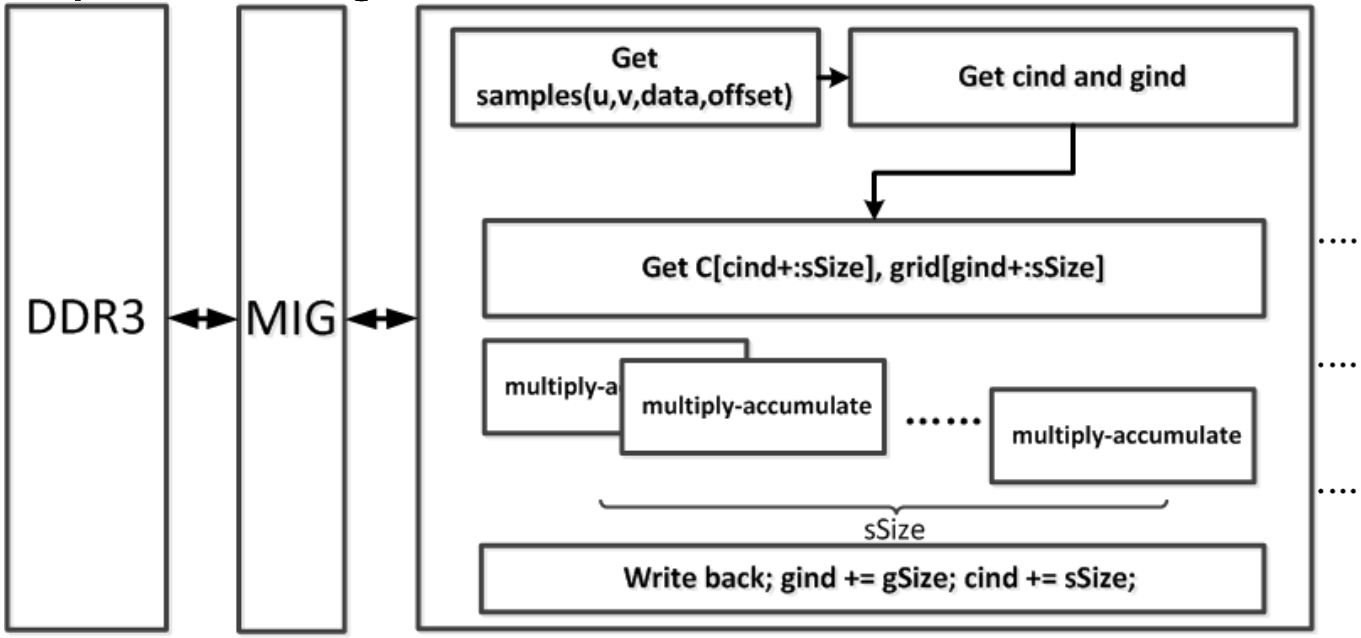


When Phase factor width is 24-bit

1. **Both Power Efficiency and Speedup have a stage;**
2. **Computation precision is at magnitude of 10^{-6};**
3. **For model of Gridding + FFT, there will be precision redundancy about 2 magnitude for Gridding**
4. **Since precision is adequate with 18-bit Phase factor, why is it set to be 24 bit?**

SJTU subtask 4: Gridding Algorithm Implementation On FPGA

- *An efficient hardware accelerator design of Gridding algorithm on FPGA*
  - Loop unrolling
  - Pipeline stages



Pipeline structure

Overall structure

SJTU subtask 4: Gridding Algorithm Implementation On FPGA

SKA
SQUARE KILOMETRE ARRAY
SDP
SCIENCE DATA PROCESSOR

- The functionality and performance are verified on Xilinx Virtex-6 ML605
  - Relative error is under 4.5 × 10−5
  - Speedup: 8.34

	Software on CPU	FPGA
Clock frequency	2.5GHz	156.25MHz
Running cycles	425000	3150
Gridding rate (million points/s)	238.235	2008.93
Number of samples	180	
gSize（The scale of grid）	128	
sSize(Width of convolution function)	15	

SJTU subtask5: Variable Precision Floating-point Unit Implementation On FPGA

SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PROCESSOR

Unum arithmetic is first proposed by Professor John L. Gustafson in 2015

IEEE 754 Floats

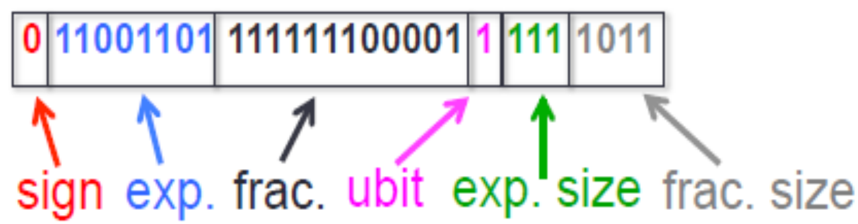
IEEE Floating Point Representation

s	exponent	mantissa
1 bit	8 bits	23 bits

IEEE Double Precision Floating Point Representation

s	exponent	mantissa
1 bit	11 bits	52 bits

Unum Floats



Wasting of Bit Width

No matter a number is **large or small**

No matter a number need **high or low precision to represent**

→ Fixed bit width of components in IEEE 754 floats

High information-per-bit

Bit width of exponent and fraction identified by exp.size and frac.size

Depend on the number to be represented

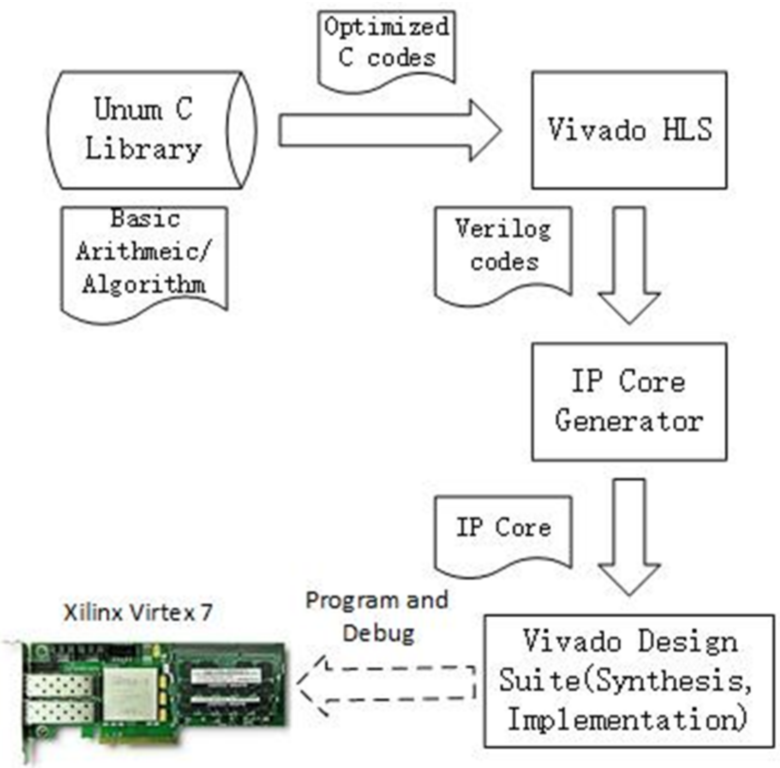
Precision Loss

Constriction of fraction bit width → Actual value round to a approximate value

No Precision Loss

Constriction of fraction bit width → Set ubit bit to 1 to represent the range of accurate value

SJTU subtask5: Variable Precision Floating-point Unit Implementation On FPGA



Contrasting the precision and bit width of addition arithmetic

First Operand	-6.0351562500
Second Operand	6.1201171876
Exact Value	8.49609376e-2
Result(754)	8.49609375...e-2
Bit Width(754)	32 bits
Result(Unum)	(8.49609375...e-2, 8.49609375...e-2)
Bit Width(Unum)	23 bits

Contrasting the precision and bit width of multiplication arithmetic

Operand 1	-3.5477
Operand 2	3.2602
Exact Value	-1.156621154e1
Result(754)	-1.1566211539...e1
Bit Width(754)	64 bits
Result(Unum)	(-1.1566211540...e1, 1.1566211538...e1)
Bit Width(Unum)	45 bits

Fudan University (FDU)

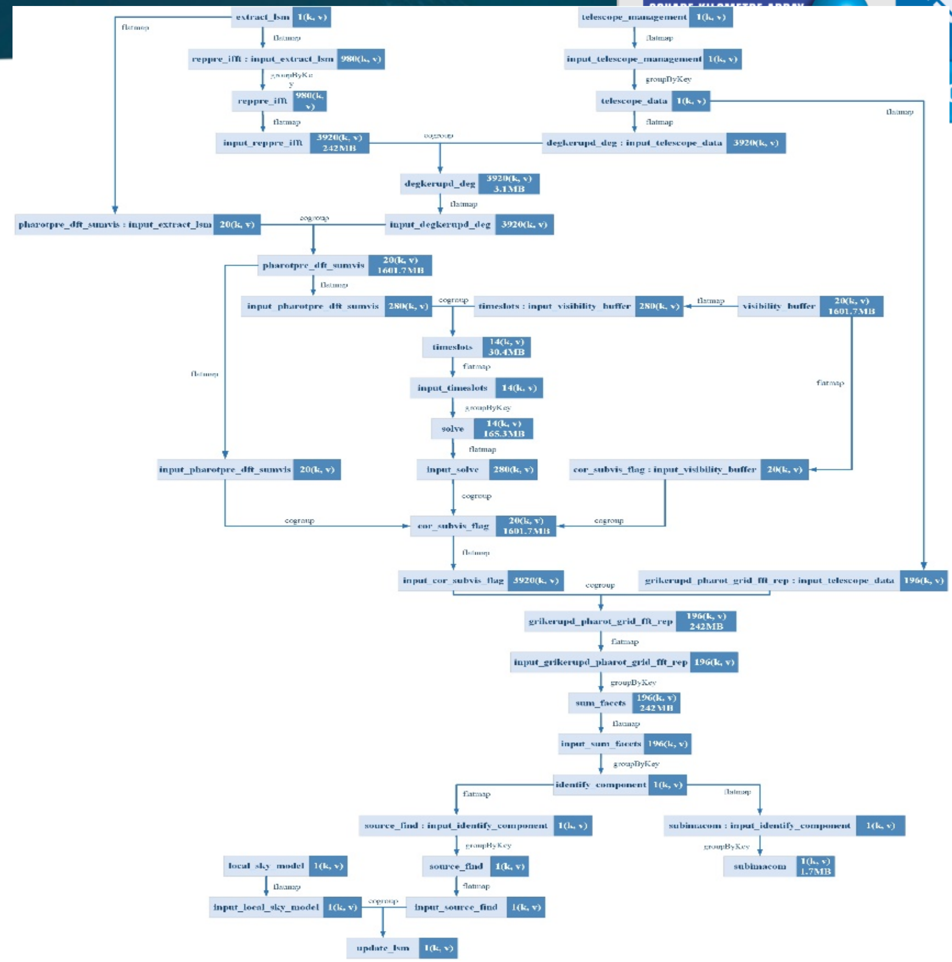
Leading organization of China MoST project on critical technologies of SKA-SDP

- Big data processing
  - COTS: execution framework, i.e., Spark
  - Spark + TensorFlow/Caffe for data accelaration

- Science
  - Pulsar search pipeline by machine learning and artificial intelligence (AI), in particular, deep learning techniques

FDU subtask 1: COTS -- Spark Optimizations on Imaging Pipeline

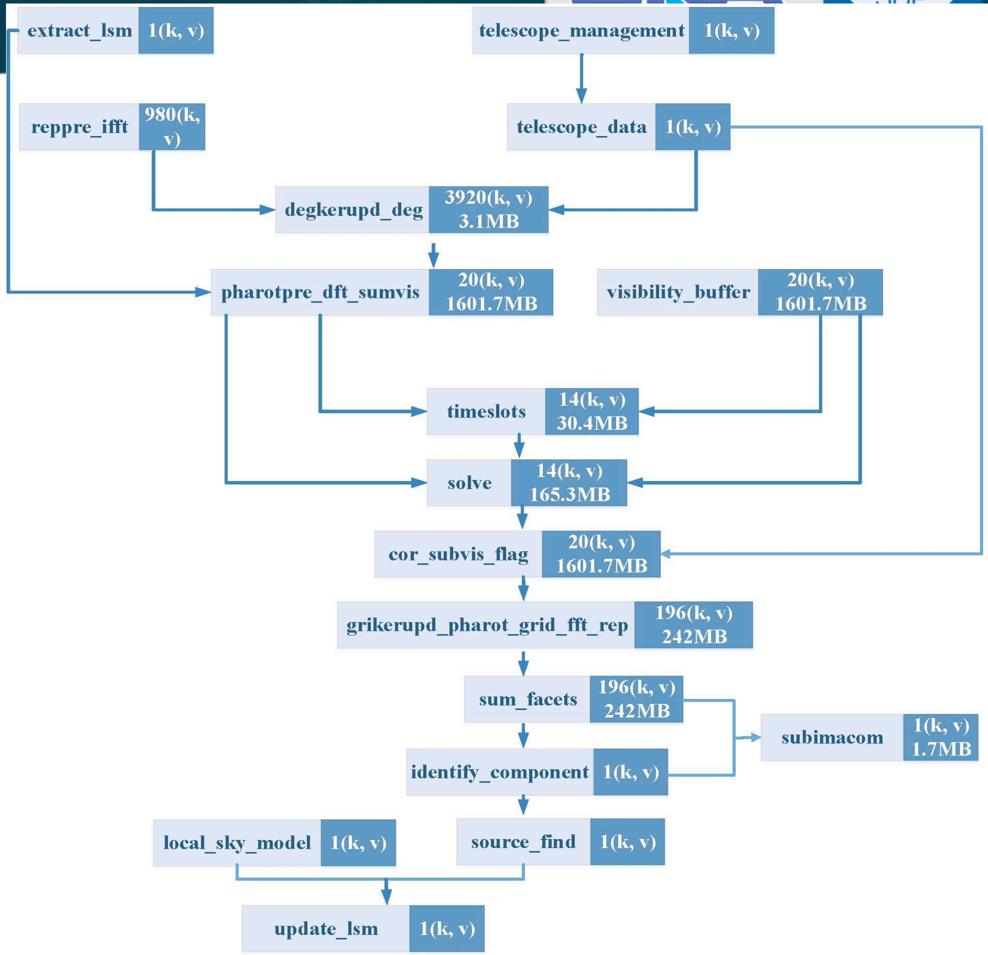
- ➢Selection of Spark as a COTS task
- ➢Bottleneck analysis on Spark
  - ✓All stages of the execution time exceed a minute including cogroup or groupByKey operations which causes shuffle operations
  - ✓Too many RDDs to consume memory resources
  - ✓Too many unnecessary copies by using “flatMap” operations
  - ✓Unnecessary join costs for two or three massive RDDs for specified combination of key



Original Data Model for MID1I CAL pipeline on Spark

FDU subtask 1: Optimizations on Spark for MID1 ICAL pipeline

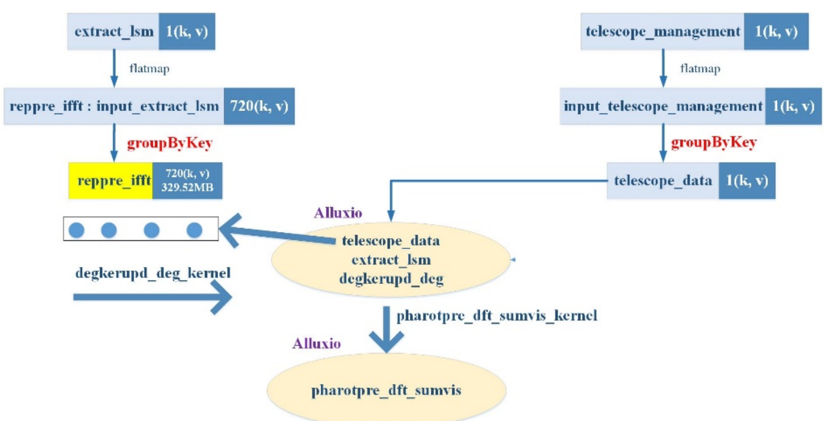
- ✓ Replacing cogroup (broadcast, third-party key-value store serving as distributed memory storage) to avoid shuffle operations
- ✓ using Alluxio as a data sharing tool
- ✓ using Spark partitioning and broadcast to replace “cogroup”
- ✓ merging stages to reduce RDDs



Simplified Data Model for MID1 ICAL pipeline on Spark

FDU subtask 1: COTS -- Initial Optimization Results

- ✓ Test settings: on Inspur clusters, i.e., 5 nodes , memory 214GB/node, 256 cores
- ✓ Speedup of 15x: 289 seconds/ node, 267seconds/3 nodes, 266s/5 nodes VS. the-state-of-the-art with more than 1 hour on memory 1TB
- ✓ This result could be further improved by extending the memory capacity and by adopting an efficient serialization method (e.g., Kryo)



Spark with Alluxio

Job ID	Including Stages	Running Time(Seconds)
0	extract_lsm broadcast local sky model	11.9
1	telescope_data broadcast telescope data	6.0
2	Merge reprojection_predict_ ifft and degridding kernel update degrid as reprojection_predict_ ifft_degridding phase_rotation_predict_dft_sum_visibilities broadcast phase_rotation_predict_dft_sum_visibilities	81.4
3	visibility_data broadcast visibility_data	50.4
4	Merge timelots and slove as timeslots_solve broadcast timeslots_solve	0.9
5	cor_subvis_flag broadcast cor_subvis_flag	54.8
6	identify_component broadcast identify_component	10.7
7	update_lsm	2.9
8	subimacom	0.5
Total		266.1

Execution times for Different Stages (5 nodes)

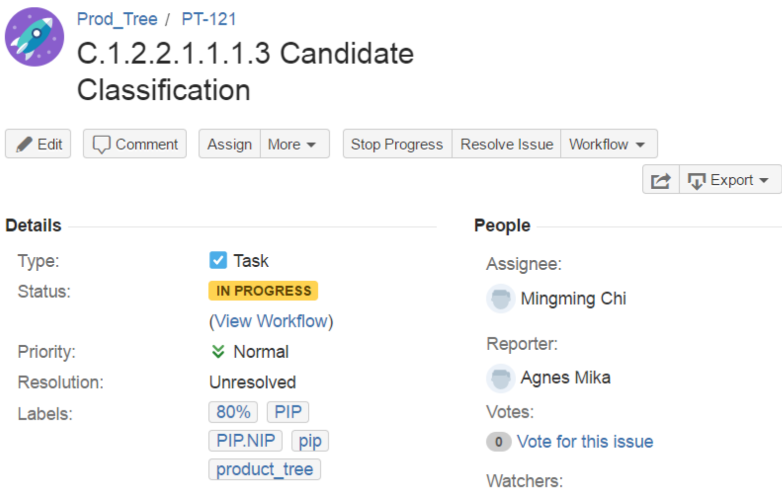
SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PRO

A Pulsar Search Pipeline by Deep Learning Techniques

Mingmin Chi
Baokun Wang, Yunfeng Zhang, Yiqing Qin and Zexin Liao
Fudan University, Shanghai, China
Contact: mmchi@fudan.edu.cn

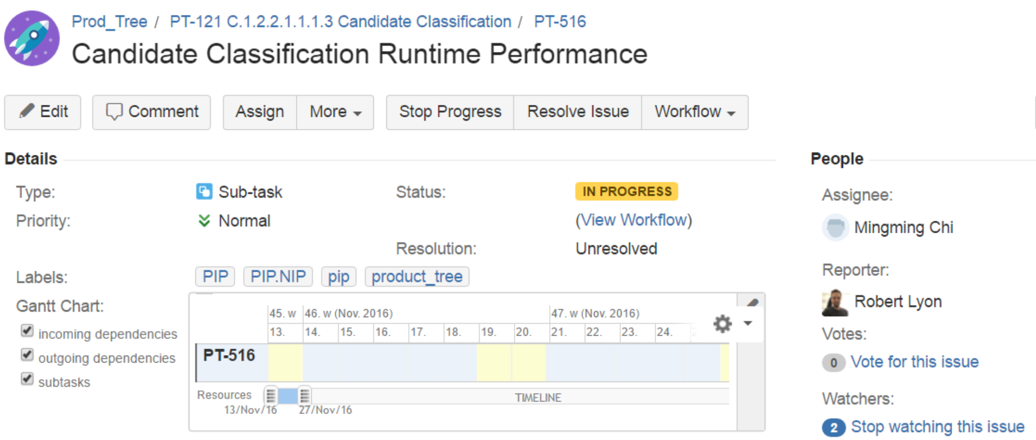
FDU subtask 2: A Pulsar Search Pipeline by Deep Learning Techniques

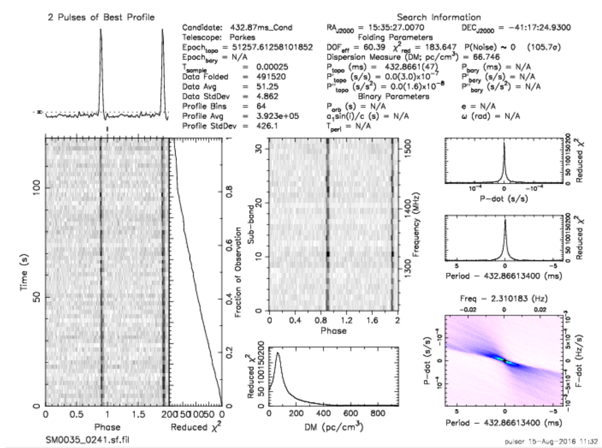
Data volume estimation



After preprocessing, i.e., by Presto, the estimated number of pulsar candidate documents

- SKA: 9M/h, Nbeam * n * (3600 / 600)= 1500 * 1000 * 6, 112TB/h
- FAST: 0.114M/h, 19 * 1000 * 6, 1.4TB/h
- Parkes: 78,000/h, 13 * 1000 * 6 , 0.97TB/h

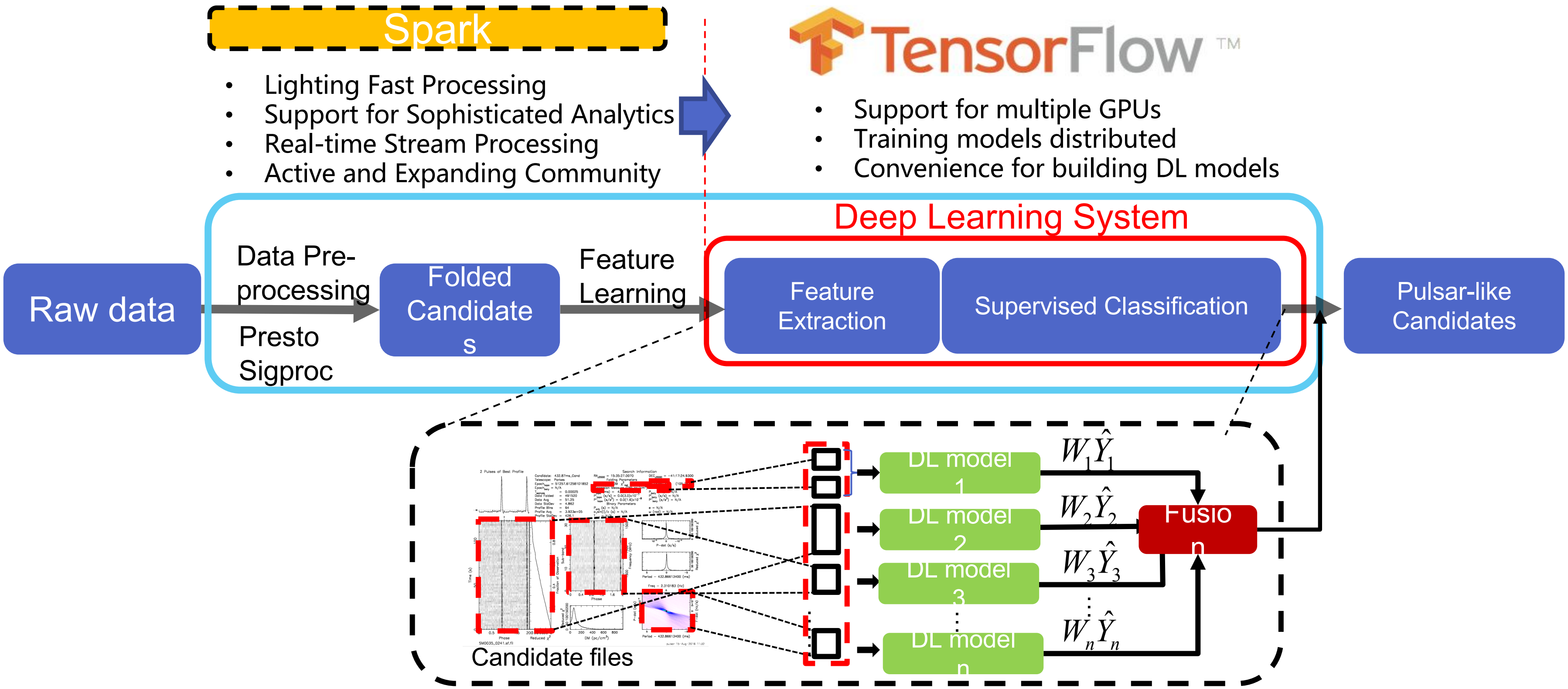




FDU subtask 2: A Pulsar Search Pipeline by Deep Learning Techniques

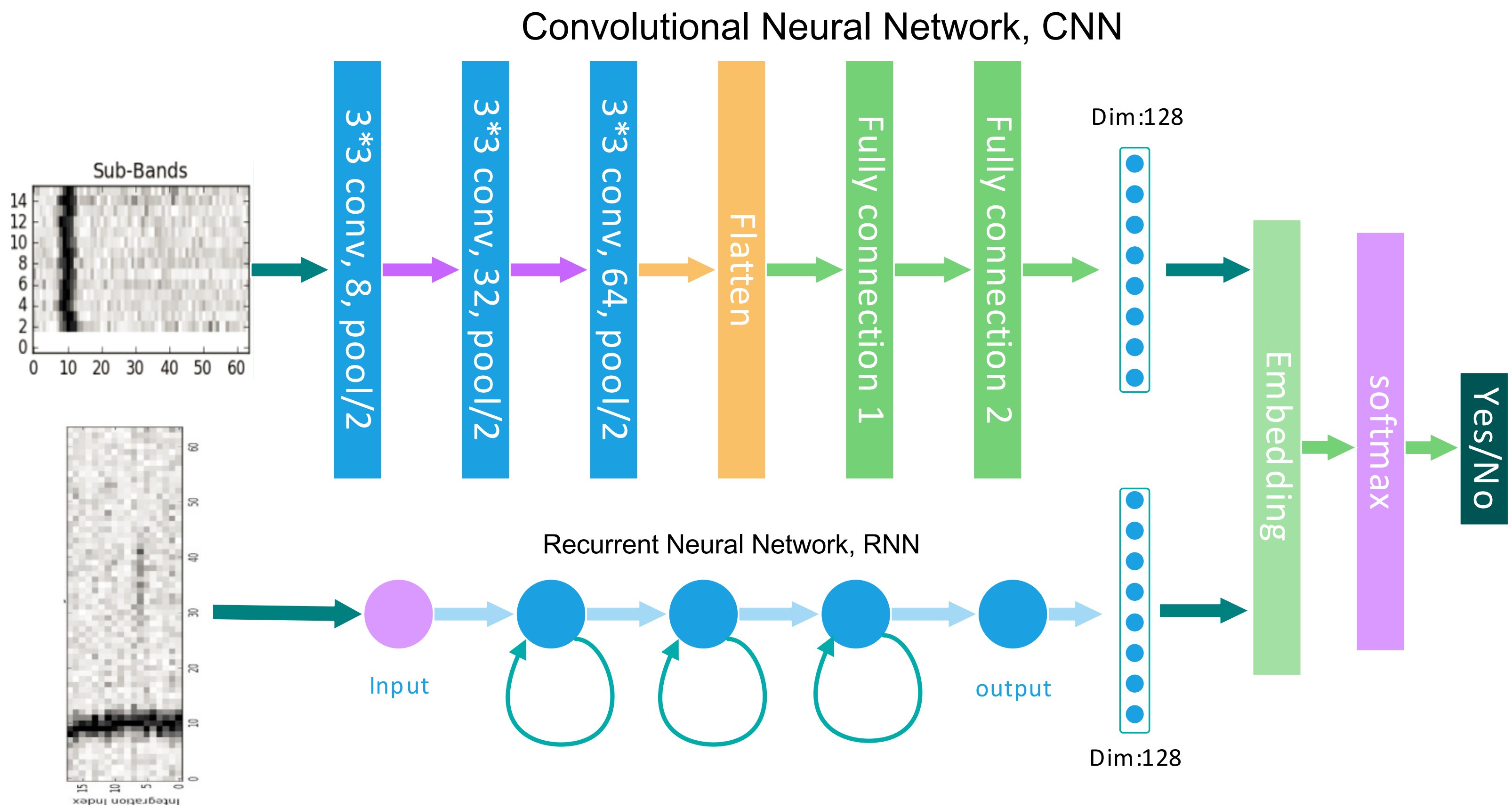
Recent Progress: PSDL V1

Automatic Pulsar Search using Deep Learning (PSDL V1)



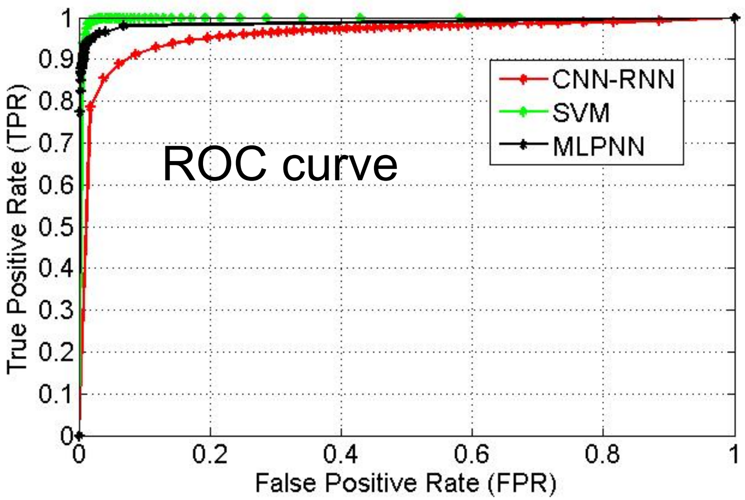
FDU subtask 2: A Pulsar Search Pipeline by Deep Learning Techniques

Hybrid of RNN and CNN

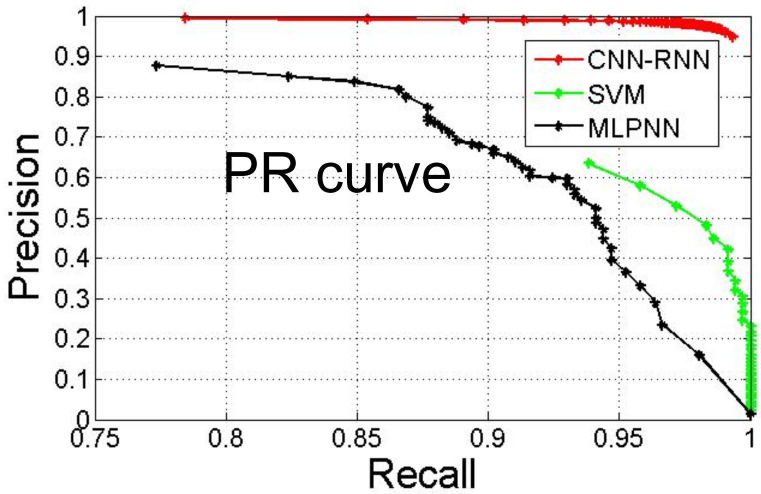


FDU subtask 2: A Pulsar Search Pipeline by Deep Learning Techniques

Empirical Results



MedlatTraingData
Positive 1196
Negative 89996



Reports can be found in
https://jira.ska-sdp.org/browse/PT-121?jql=text%20~%20%22mingming%20chi%22
https://jira.ska-sdp.org/browse/PT-516

The designed hybrid of CNN-RNN model obtains better recall and precision compared to those by the "shallow" machine learning methods, such as support vector machine (SVM) and multi-layer perceptron neural networks (MLPNN) by the hand-designed features proposed in [Lyon et al. 2015]

Inspur Inc. of China

Leading vendor of HPC platform: manufacturer of top 1 (2016) supercomputer Tianhe-2

Contributing areas:

- ➢ Compute platform design
  - ☐ PDR
    - Compute platform: Hardware alternatives and developments
  - ☐ Product Tree
    - Eight elements: Racks, Compute Nodes, Storage, and Network
- ➢ Optimization of the Gridding algorithm
  - ☐ Knights Corner Xeon Phi
    - 3.5x performance improvements
  - ☐ Knights Landing Xeon Phi
    - Current work

Overview of Inspur's tasks

- Inspur built a prototyping cluster based on Xeon Phi MIC accelerator (model code: KNL)
- Continue to keep the KNL cluster open to the SKA community to benchmark the SKA software
- Inspur has completed SKA-SDP tasks:
  - TSK-64 task "Determine the Quality for C.1.1 Processor Platform"
  - A high-level proposal for TSK-1511 "Propose suitable implementation of compute platform architecture from inspur portfolio"
  - we have implemented the gridding algorithm on KNL and achieve a good performance improvement. We will optimize other key algorithms on KNL for the SKA

Inspur subtask1: MIC accelerator based architecture

SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PROCESSOR

Features

- Bootable host processor
- 72 cores，288 threads
- 3+TFLOP/s DP，6+ TFLOP/s SP
- Up to 16GB on-package MCDRAM，400GB/s~500GB/s
- 2VPU pre core
- Binary compatible with Intel Xeon

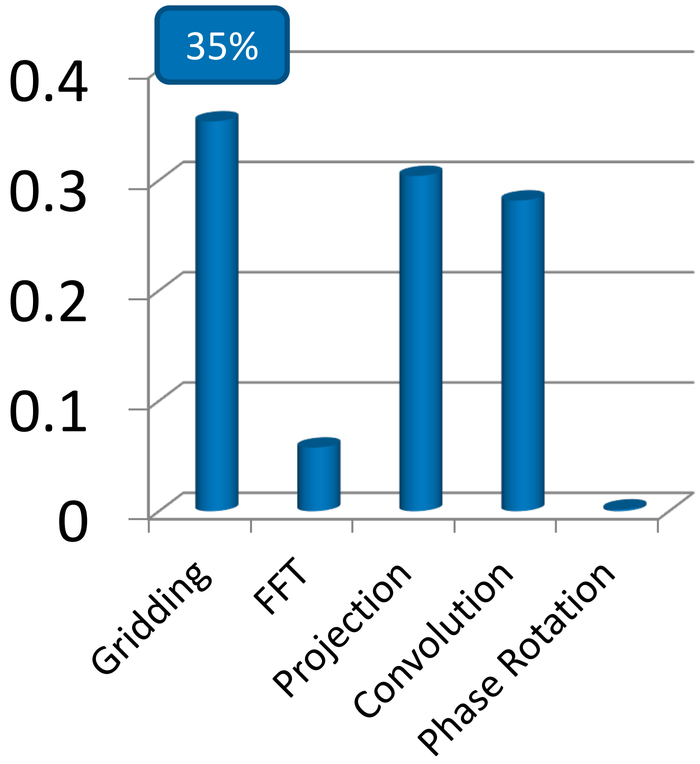
Upsides

- More cores and threads
- 512-bit vector register，supported AVX-512
- High-bandwidth memory MCDRAM
- Easy for programming

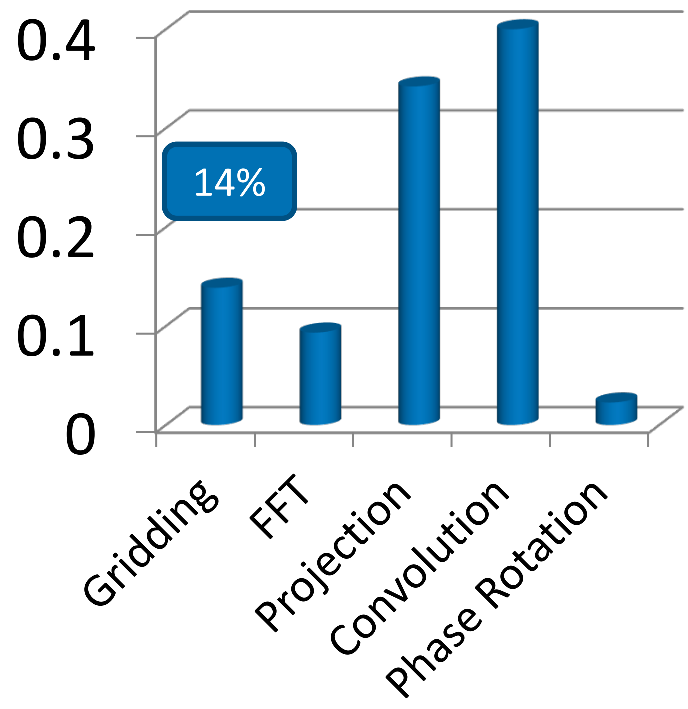
Inspur subtask2： acceleration of Gridding in SKA-SDP

Key algorithms in SKA-SDP

Gridding in the SKA1-Low

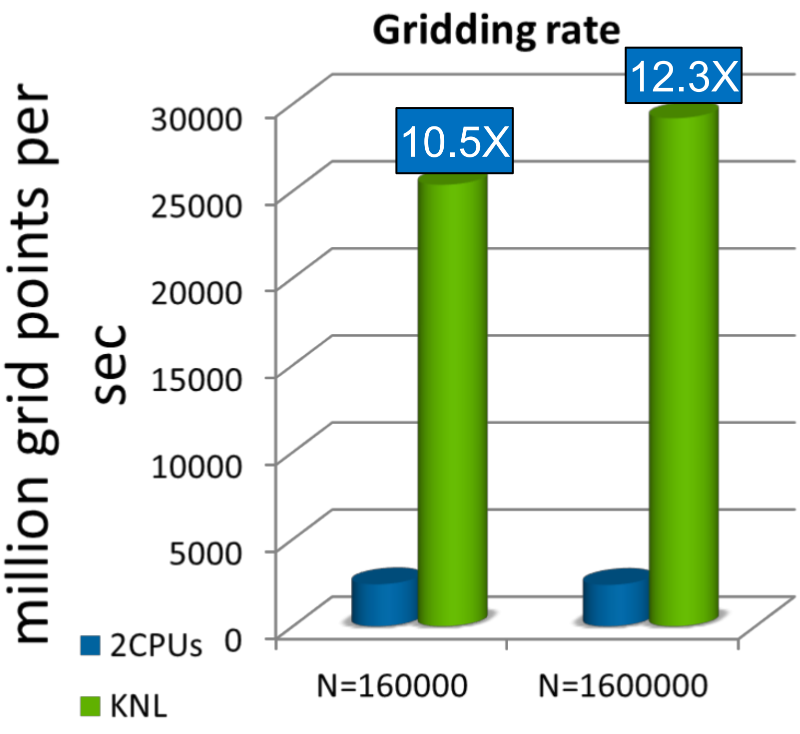


Gridding in the SKA1-Mid



Inspur subtask2： acceleration of Gridding in SKA-SDP

Performance Achievements

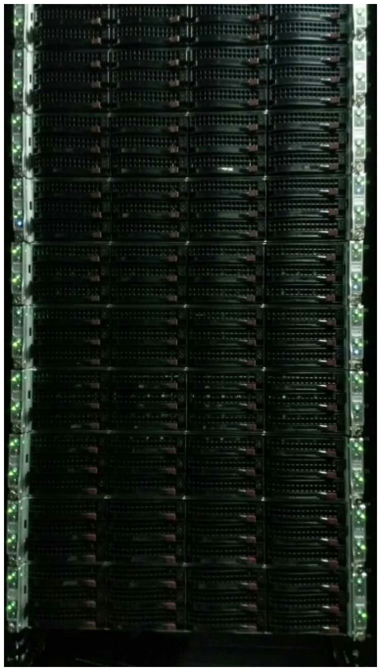


CPU	2*Intel(R) Xeon(R) E5-2620 v3@ 2.4GHz 256GB memory DDR4
KNL(one node of the KNL cluster)	64*Intel(R) Genuine Intel(R) CPU 0000 @ 1.3GHz 48GB memory DDR4 + 16GB MCDRAM
Network	OPA 100Gb/s

Test Platform

Inspur subtask2： acceleration of Gridding in SKA-SDP

A public service: free prototyping platform for key algorithms of SDP



Nodes	64
KNL	64*Intel(R) Genuine Intel(R) CPU 0000 @ 1.30GHz, 16GB MCDRAM, DDR 2133
Storage	Intel Enterprise Edition for Lustre
Network	Intel Omni-Path Architecture
OS	Red Hat Enterprise Linux Server release 7.1 (Maipo)
Compiler	icc, icpc, ifort (version 17.0.0)
MPI	Intel(R) MPI Library for Linux* OS, Version 2017 Build 20160721
Tools	Intel Parallel Studio XE

- A proposed prototype base on KNL Cluster

Inspur subtask2： acceleration of Gridding in SKA-SDP

A public service: free prototyping platform for key algorithms of SDP

- SKA audience are welcome to use the KNL cluster for free
- Apply online through http://inspurhpc.com/KEEP
- If you have any questions, please send email to changxujian@inspur.com
- Applications will be reviewed

Kunming University of Science and Technology (KUST) `

- Astronomical information technology joint laboratory, cooperating with National Astronomical Observatory, Chinese Academy of Sciences
- A astronomical technology research team of nearly 40 people
- Dedicated in development of astronomical data processing software and technology
- Virtualization integration and control technology for heterogeneous devices of telescope

LOVELL

Projects involved:

- Data processing for MingantU Spectral Radioheliograph (high temporal, high spatial, and high spectral resolution almost simultaneously)
- Real-time data acquisition and CCD control for the 1meter New Vacuum Solar Telescope—NVST
- Observation control for LAMOST (The Large sky Area Multi-Object fiber Spectroscopic Telescope)
- High speed data collection and observation for the 40 m radio telescope in Kunming (Chang'E1/2)

KUST tasks overview

- TSK-167: Analysis the execution framework — DALiuGE and wirting the cook-book for DALiuGE deployment and execution.
- TSK-168: Generate input for PIP <-> EF interface document (Control layer)
- TSK-169: Generate input for PIP <-> EF interface document (Data I/O layer)

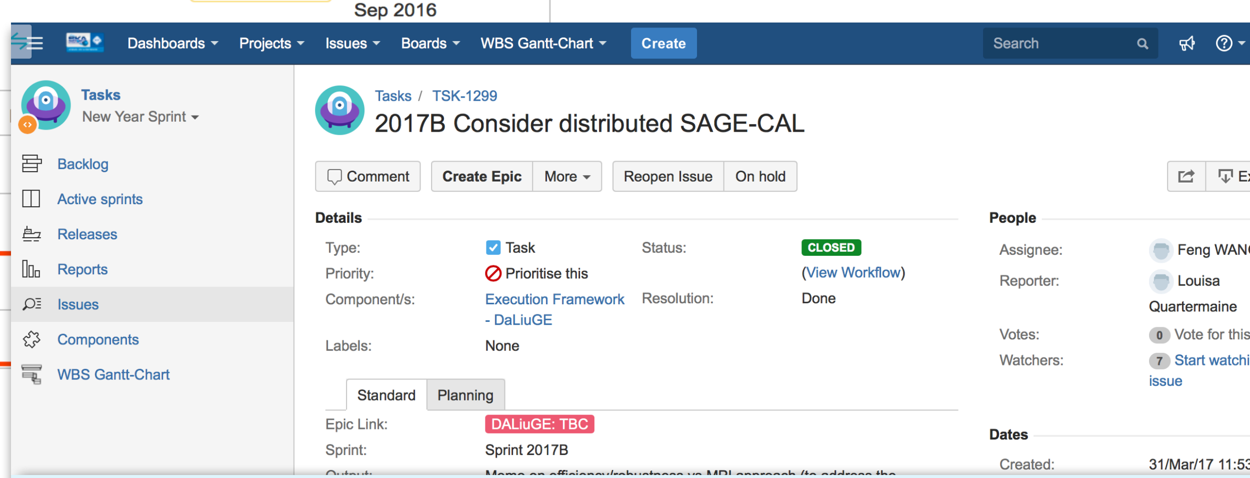
Prototyping and Analysis Development Tasks

Key	Summary	P	Assignee	Stat	
DATA-232	DATA-162 / Magnus support for large scale test		Mohsin Ahmed Shaikh	CL	
DATA-210	Support multiple Drop islands in Physical Graph generation		Chen Wu	CL	
DATA-171	Analyse results of deployment tests		Andreas Wicenec	RE	
DATA-170	Support deployment tests on Tianhe-2		Andreas Wicenec	RESOLVED	SDP High Risk - Sep 2016
DATA-167	Write cook-book		Feng WANG	RESOLVED	SDP High Risk - Sep 2016
DATA-166	Organise access to Tianhe2		Tao An	IN PROGRESS	SDP High Risk - Sep 2016

Related tasks:
- TSK-340: Execution Framework – DALiuGE
- TSK-342: Assessment of technical risk associated with MSMFS and distributed calibration
- TSK-343: Consider distributed SAGECAL

System Engineering Development Tasks

Key	Summary
DATA-174	EF:Identify main risk areas to be addressed during this sprint
DATA-173	EF: Differential Risk Analysis and Report
DATA-169	Generate input for PIP <-> EF interface document (Data I/O layer)
DATA-168	Generate input for PIP <-> EF interface document (Control layer)



- TSK-1299: 2017B Consider distributed SAGECAL

KUST subtask1: Case study I for DALiuGE— MUSER

SQUARE KILOMETRE ARRAY
SCIENCE DATA PROCESSOR

ScienceDirect Journals Books Register Sign in

Download PDF Export Search ScienceDirect Advanced search

Article outline Show full outline

ELSEVIER
Astronomy and Computing
Volume 20, July 2017, Pages 1–15

Full length article

DALiuGE: A graph execution framework for harnessing the astronomical data deluge

C. Wu[a], R. Tobar[a], K. Vinsen[a], A. Wicenec[a], D. Pallot[a], B. Lao[b], R. Wang[c], T. An[b], M. Boulton[a], I. Cooper[a], R. Dodson[a], M. Dolensky[a], Y. Mei[d, e], F. Wang[d, e]

Cornell University Library

arXiv.org > astro-ph > arXiv:1701.04907

Search or Article (Help | Advanced sea

Astrophysics > Instrumentation and Methods for Astrophysics

OpenCluster: A Flexible Distributed Computing Framework for Astronomical Data Processing

Shoulin Wei, Feng Wang, Hui Deng, Cuiyin Liu, Wei Dai, Bo Liang, Ying Mei, Congming Shi, Yingbo Liu, Jingping Wu

(Submitted on 18 Jan 2017)

The volume of data generated by modern astronomical telescopes is extremely large and rapidly growing. However, current high-performance data processing

Cornell University Library

arXiv.org > astro-ph > arXiv:1705.06067

Search or (Help | Advan

Astrophysics > Instrumentation and Methods for Astrophysics

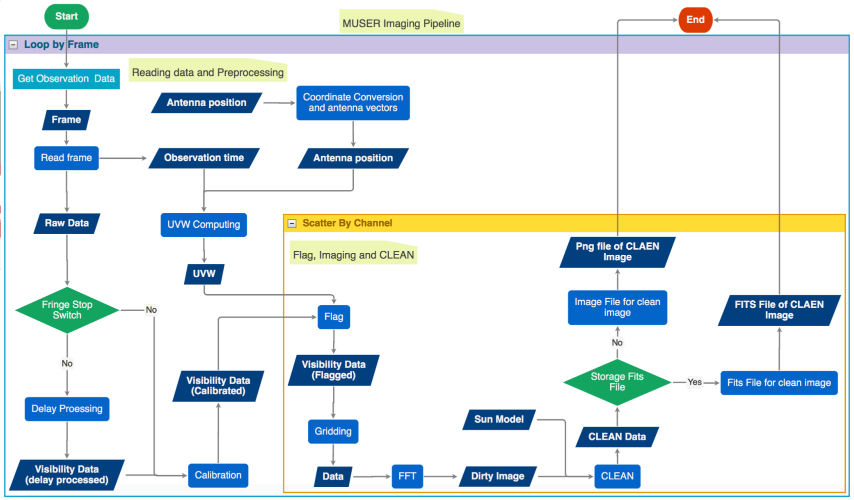
High Performance Negative Database for Massive Data Management System of The Mingantu Spectral Radioheliograph

Congming Shi, Feng Wang, Hui Deng, Yingbo Liu, Cuiyin Liu, Shoulin Wei

(Submitted on 17 May 2017)

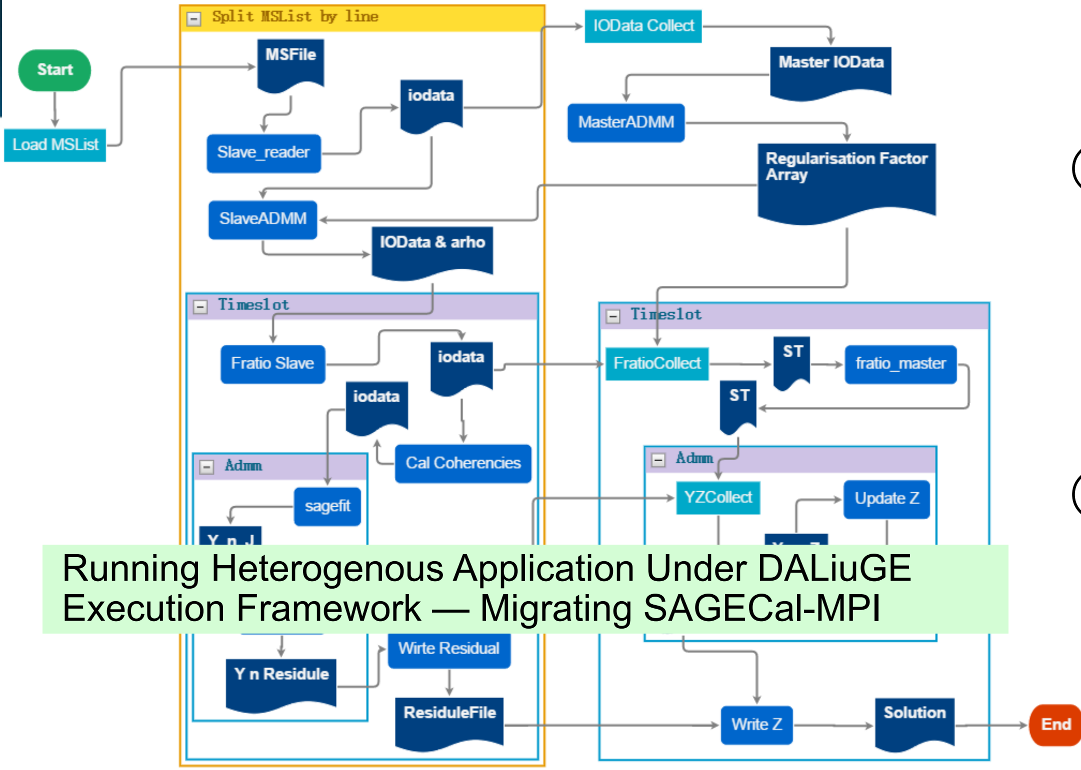
As a dedicated synthetic aperture radio interferometer, the MingantU SpEctral Radioheliograph (MUSER), initially known as the Chinese Spectral RadioHeliograph (CSRH), has entered the stage of routine observation. More than 23 million data records per day need to be effectively managed to provide high performance data query and retrieval for scientific data reduction. In light of these massive amounts of data generated by the MUSER, in this paper, a novel data management technique called the negative database (ND) is proposed and used to implement a data management system for the MUSER. Based on the key-value database, the ND technique makes complete utilization of the complement set of observational data to derive the requisite information. Experimental results showed that the proposed ND can significantly reduce storage volume in comparison with a relational database management system (RDBMS). Even when considering the time needed to derive records that were absent, its overall performance, including querying and deriving the data of the ND, is comparable with that of an RDBMS. The ND technique effectively solves the problem of massive data storage for the MUSER, and is a valuable reference for the massive data management required in next-generation telescopes.

To evaluate the usability and performance of DALiuGE, we have "migrated" all pipeline components to DALiuGE Drops. We created 12 pipeline components such as raw data acquisition, frame data distribution, dirty image processing, CLEAN, and so on as shown in the figure on the right.



MUSER Imaging --- Logical Graph in DALiuGE

KUST subtask2: Case study II for DALiuGE — Migrating SAGECal from MPI to DALiuGE



Running Heterogenous Application Under DALiuGE Execution Framework — Migrating SAGECal-MPI

Work Breakdown

Create

SPACE SHORTCUTS

- Towards CDR
- SDP Product Tree
- Sprints
- Meeting notes

- Work Breakdown Home
  - Non-product WBS Breakdown
  - Work Breakdown Structure (WBS...
    - Application Development and ...
    - Architecture WBS
    - Ancillary Platforms WBS
    - Compute system and intercon...
    - Compute System Software an...
    - Data Lifecycle Manager WBS
    - Delivery.Software WBS

Pages /... / Execution Framework Sprint Status Report

Edit

1 JIRA link

Technical Report: Migrate Sagecal From MPI to DALiuGE

Created by Feng WANG, last modified yesterday at 10:20 PM

Version 1.00 (Beta)

@Feng WANG, @Shoulin WEI, @Ying MEI, @Hui DENG

Motivation

SAGECal is a fast, distributed and GPU accelerated radio astronomial calibration package. The many optimization algorithms in SAGECal are implemented in a computationally efficient way and can be used in many other applications.

SAGECal is a self-calibration technique that uses the Expectation-Maximization (EM) algorithm to obtain the maximum likelihood estimation of the instrument and sky parameters. While the plain EM algorithm has been applied in estimating parameters for non-linear, superimposed signals, SAGECal uses an improved EM algorithm known as the SAGE algorithm in order to: 1. Speed up the convergence (compared to the plain EM algorithm); 2. Reduce the computational complexity (compared to least square estimation methods; 3. Improve the quality of calibration (compared to the Peeling calibration

Technical Report: Migrate Sagecal From MPI to DALiuGE

① SAGECal is a fast, distributed and GPU accelerated radio astronomical calibration package. Migrating the codes of SAGECal-MPI to DALiuGE is running a real astronomical software under DALiuGE.

② When using NFS shared directory on hard disk, MPI version took about 13 minutes and DALiuGE version took about 19 minutes as Sagecal-DALiuGE has to output a series of temporary files which apparently reduced the performance.

③ When using tmpfs and shared the directory with NFS, the DALiuGE version only took about 13 minutes to perform the full calculation.

④ This application proves the availability and usability of the DALiuGE execution framework.

Institute of Computing Technology, Chinese Academy of Sciences (ICT)

The Tasks on JIRA system

Haihang You

youhaihang@ict.ac.cn

Completed Task:

TSK-1284 SFFT algorithm optimization

a) Analyze the bottleneck of SFFT, Optimize SFFT algorithm
b) Proposed a new fast two-dimensional Fourier transform based on Image sparsity (2D-SFFT)
c) Proposed an Adaptive Tuning Sparse Fourier Transform (ATSFFT)

The tasks in progress:

1. TSK-434 Algorithm Library Development

a) Learn Python language and study the SKA algorithm reference library. (https://github.com/SKA-ScienceDataProcessor/algorithm-reference-library)
b) Develop the calibration and imaging algorithms in C form
c) Analyze the bottleneck of algorithms and optimize them

2. TSK-1441 ARL Imaging Pipeline on TensorFlow

a) ARL is a reference library including algorithms of simplified process of imaging
b) The numpy library have been implemented in tensor flow in GPU version
c) Tensorflow can speed up ARL significantly

3. TSK-1540 Apply the optimized SFFT algorithm to Radio Astronomy

a) Study the application of SFFT in SKA
b) Apply the optimized SFFT algorithm to SKA

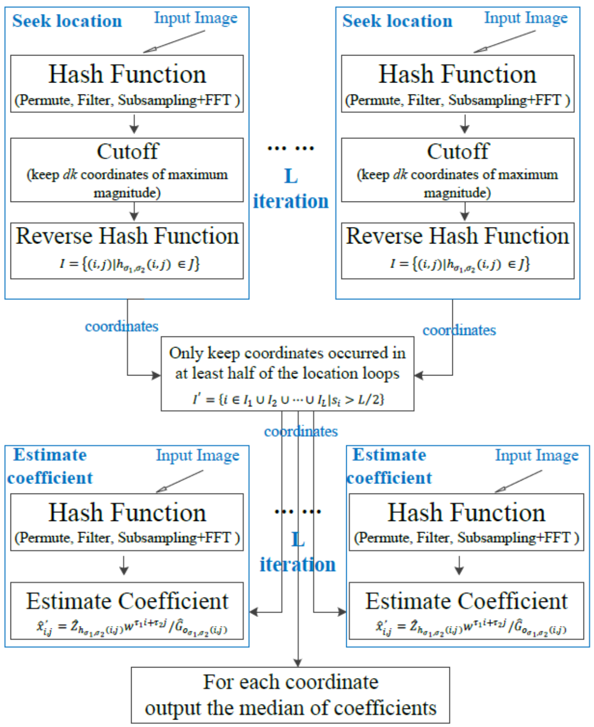
ICT subtask1: Algorithm Reference Library

Completed Tasks

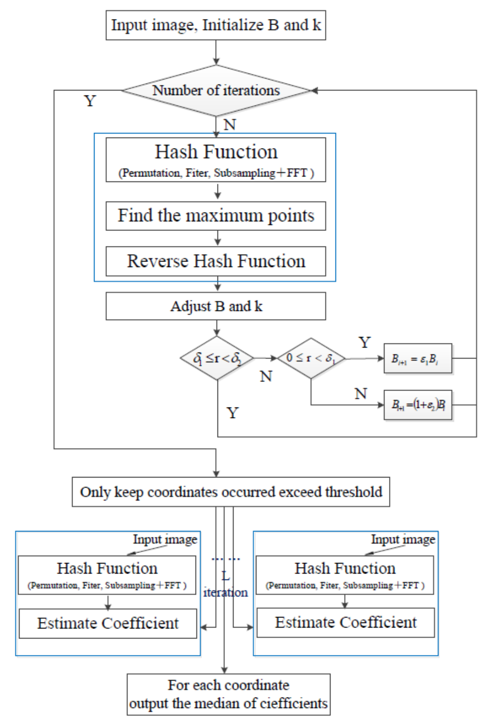
TSK-1284 SFFT algorithm optimization

1. Proposed a new fast two-dimensional Fourier transform based on Image sparsity (2D-SFFT)

2. Proposed an Adaptive Tuning Sparse Fourier Transform (ATSFFT)



(a) 2D-SFFT



(b) ATSFFT

S. Shi, R. Yang, and H. You, "A New Two-Dimensional Fourier Transform Algorithm Based on Image Sparsity," 2017 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, USA, 2017

ICT subtask1: Algorithm Reference Library

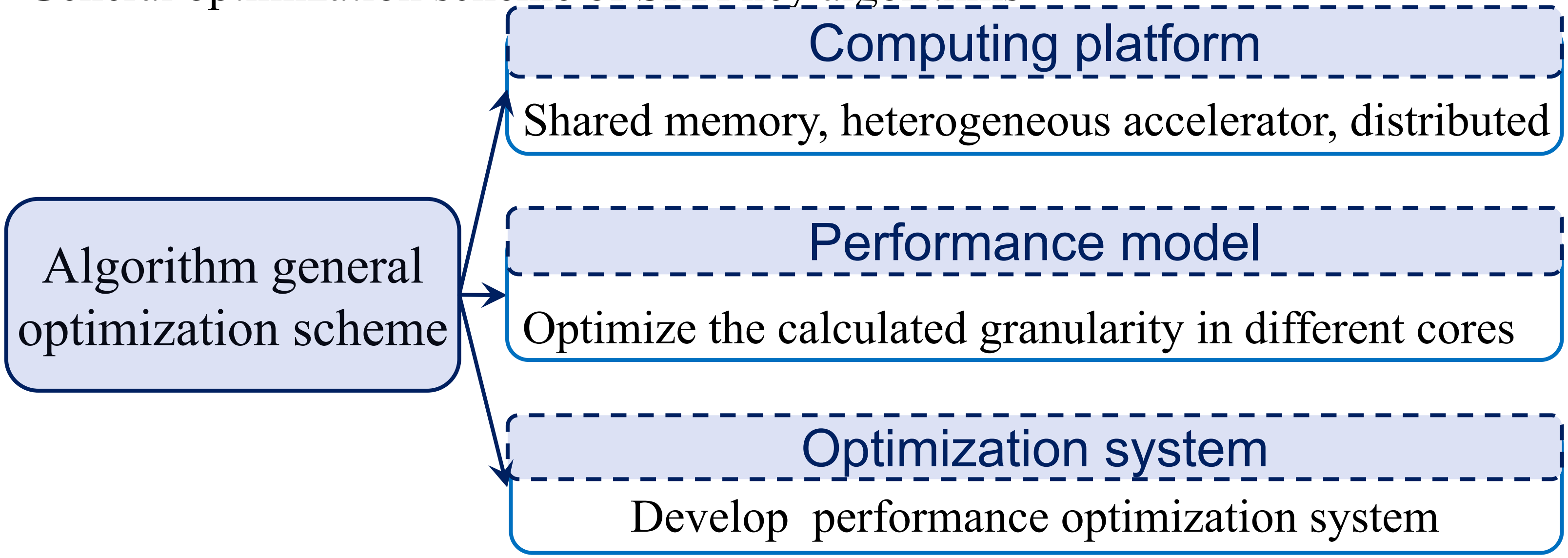
The Tasks in Progress

TSK-434 Algorithm Library Development

- Learn Python language and study the SKA algorithm reference library;
- Develop the calibration and imaging algorithms in C form;
- Analyze the bottleneck of algorithms and optimize them.

Final purpose:

General optimization scheme of SKA key algorithms



Complete the optimized version of different system

ICT subtask1: Algorithm Reference Library

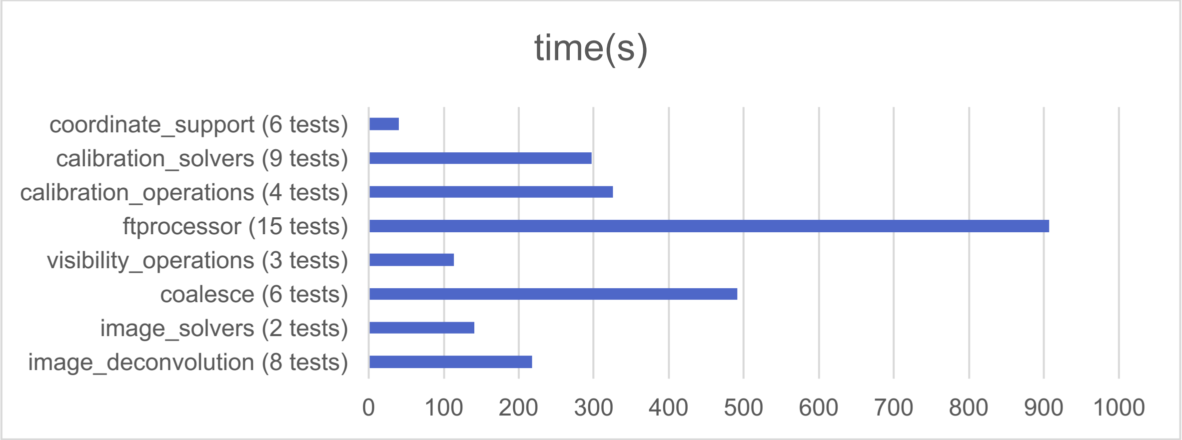
The Tasks in Progress

TSK-1441 ARL Imaging Pipeline on TensorFlow

The algorithm reference library (ARL) is designed to present calibration and imaging algorithms in a simple Python-based form.

CARL1.1	CARL1.2	CARL1.3	CARL1.4	CARL1.5	CARL1.6	CARL1.7	CARL1.8
Data	Image	Visibility	Fourier transforms	Sky components	Calibration	Util	Pipelines

Run unittest cases with given data of ARL. Show some time consuming result.



Main time consuming process: image, visibility, fft, calibration

The Tasks in Progress

TSK-1441 ARL Imaging Pipeline on TensorFlow

Reason: nested "for" loops of matrices operations. eg.

```
for ant1 in range(nants):
    for ant2 in range(nants):
        for chan in range(nchan):
            for rec1 in range(nrec):
                for rec2 in range(nrec):
                    error = x[ant2, ant1, chan, rec2, rec1] - \
                            gain[ant1, chan, rec2, rec1] * \
                            numpy.conjugate(gain[ant2, chan, rec2, rec1])
                    residual[chan, rec2, rec1] += (error * \
                        xwt[ant2, ant1, chan, rec2, rec1] * numpy.conjugate(
                        error)).real
                    sumwt[chan, rec2, rec1] += xwt[ant2, ant1, chan, rec2, rec1]

residual[sumwt>0.0] = numpy.sqrt(residual[sumwt>0.0] / sumwt[sumwt>0.0])
residual[sumwt <= 0.0] = 0.0
```

```
xshape = (nrows, nants, nants, nchan, nrec, nrec)
x = numpy.zeros(xshape, dtype='complex')
xwt = numpy.zeros(xshape)
for row in range(nrows):
    for ant1 in range(nants):
        for ant2 in range(ant1+1, nants):
            for chan in range(nchan):
                ovis = numpy.matrix(vis[row, ant2, ant1, chan].reshape([2,2]))
                mvis = numpy.matrix(modelvis[row, ant2, ant1, chan].reshape([2,2]))
                wt = numpy.matrix(weight[row, ant2, ant1, chan].reshape([2,2]))
                x[row, ant2, ant1, chan] = numpy.matmul(numpy.linalg.inv(mvis), ovis)
                xwt[row, ant2, ant1, chan] = numpy.dot(mvis, numpy.multiply(wt, mvis.H)).real
```

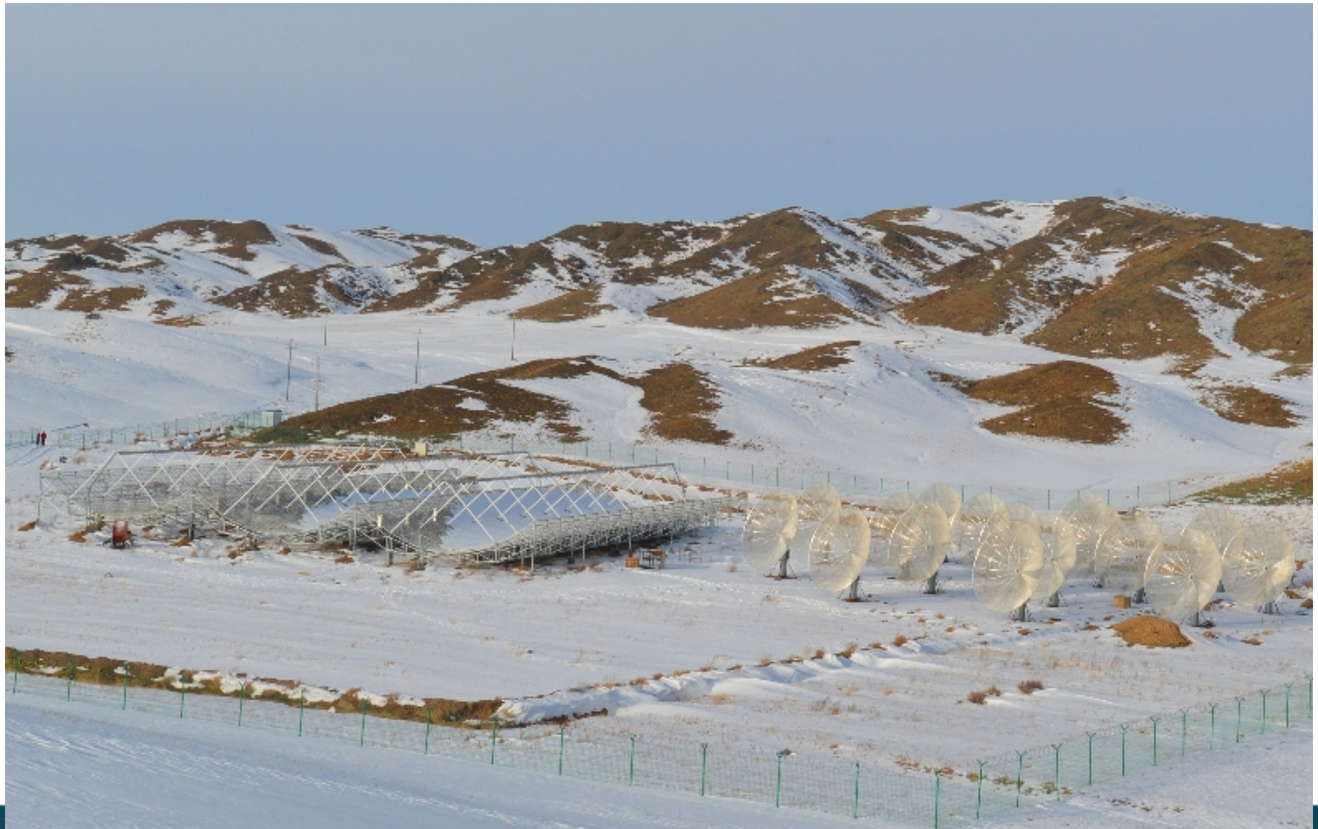
Slover: numpy operations——build-in optimization and vectorization

- The main strengths of TensorFlow are very fast dot products and matrix exponents. The dot product is approximately 8 and 7 times faster respectively with Tensorflow compared to NumPy for the largest matrices.
- Numpy has Ndarray support, but doesn't offer methods to create tensor functions and automatically compute derivatives (+ no GPU support).

National Astronomy Observatory of China (NAOC)

Overview of Tianlai pathfinder experiment

- Interferometer Array for 21cm intensity mapping and dark energy
- 3x15mx40m cylinders, 96 dual polarization receiver units
- 16x 6m dishes
- Frequency: 400-1400MHz (Redshift z=0-2.5)



Similarity between Tianlai and SKA & What NAOC could do for SKA SDP

- **Similar scientific goal:** Neutral hydrogen(21cm)
- **Similar hardware:** Interferometer and multi-beam
- **Similar Software:** RFI/Calibration/Imaging
- **Similar challenge:** Mass data processing, novel processing method for next generation interferometer array
- **What could we do for SKA SDP?**

(1) Astronomical methodology/algorithm development, rather than optimizing and accelerating algorithm itself.

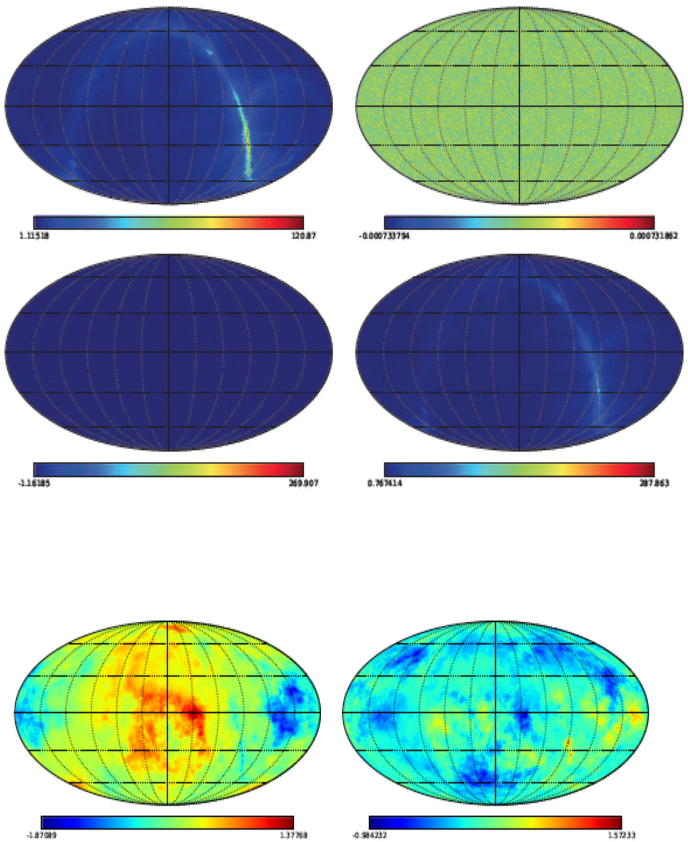
(2) Working on issue of drift scan sky survey, include RFI mitigation/excision, calibration and whole sky imaging and so on, and aim to provide the community with a whole set of basic pipeline which would be optimized and implemented in the future SKA drift scan survey.

(3) Developing the technique of extraction of 21cm signal from the galaxy synchrotron emission and developing the model-independent method to do it.

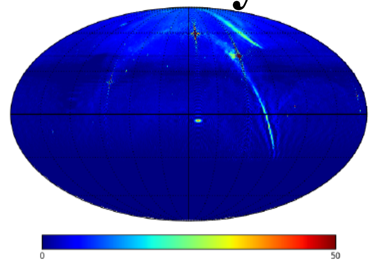
NAOC subtask1: data processing simulation system

SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PROCESSOR

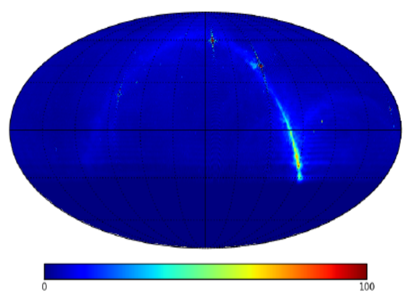
Whole sky imaging



Sky map simulation

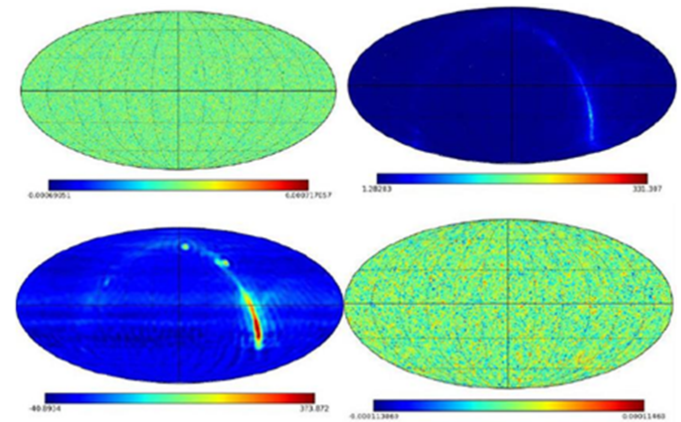


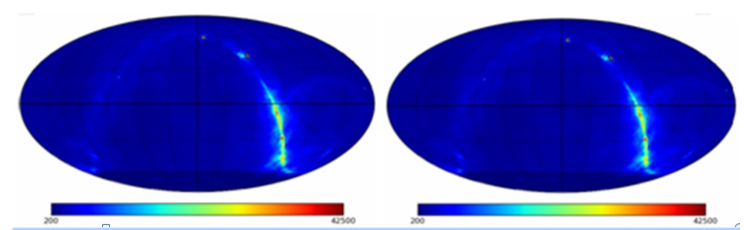
Sky image obtained by equal space of feeds



Sky image obtained by unequal space of feeds alone 3 cylinders

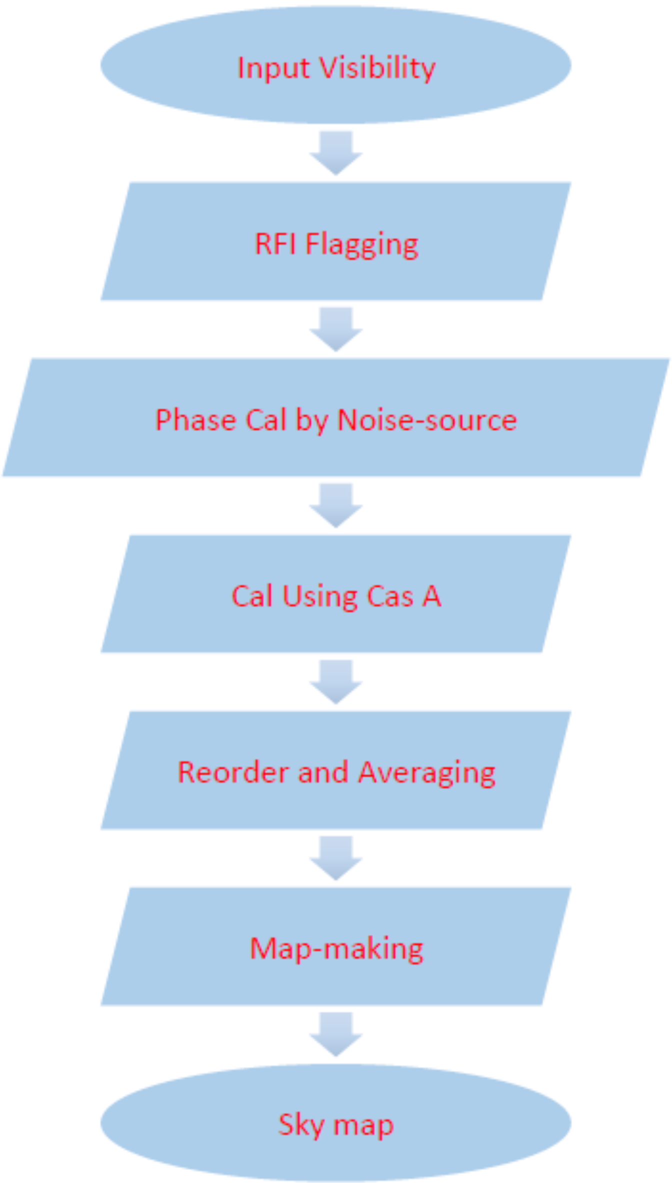
Foreground subtraction



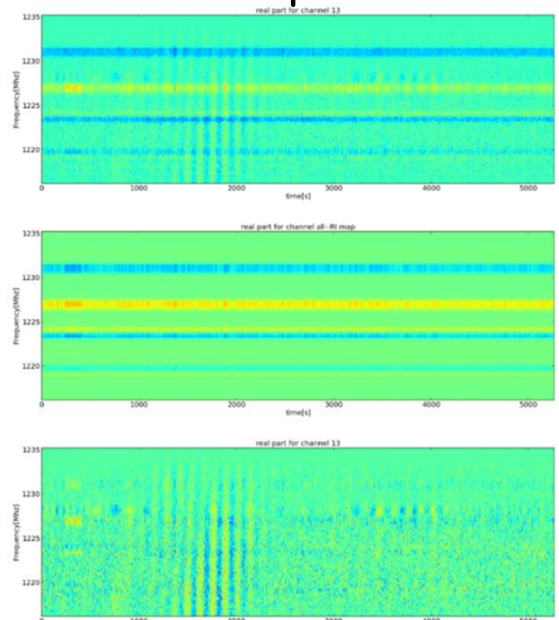


Data compression

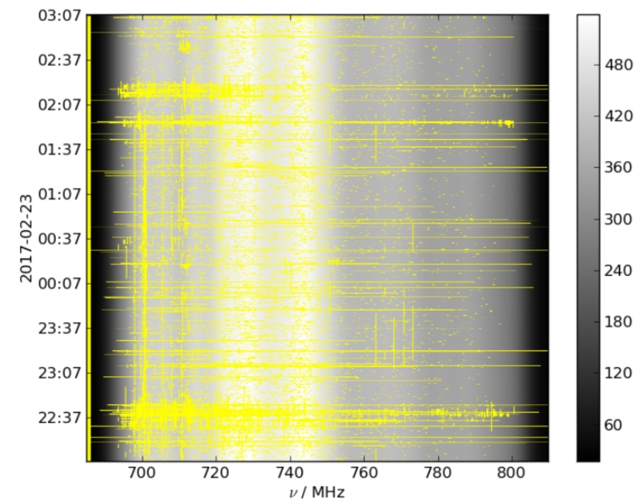
NAOC subtask2: Parallel pipeline system for Interferometer



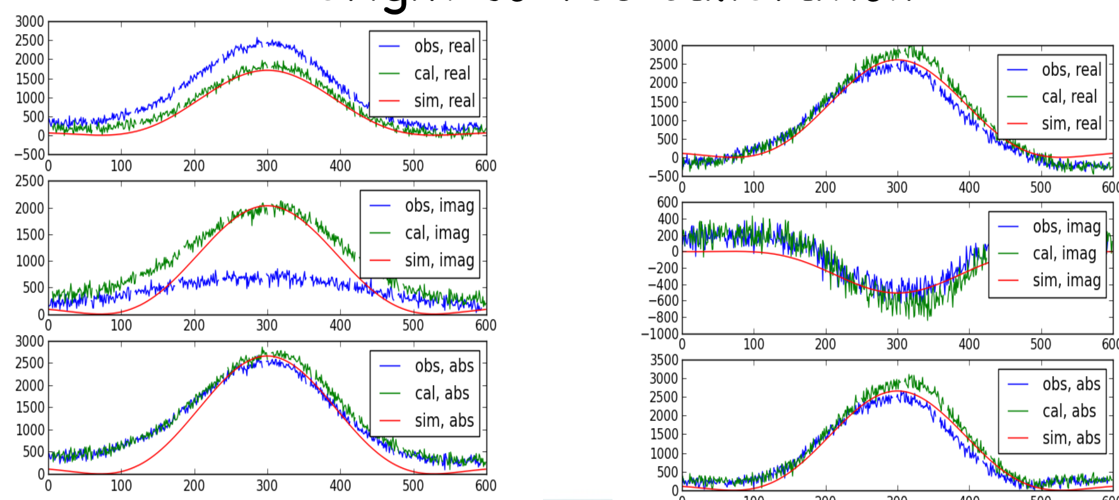
SVD RFI separation



2D var-threshold flagging



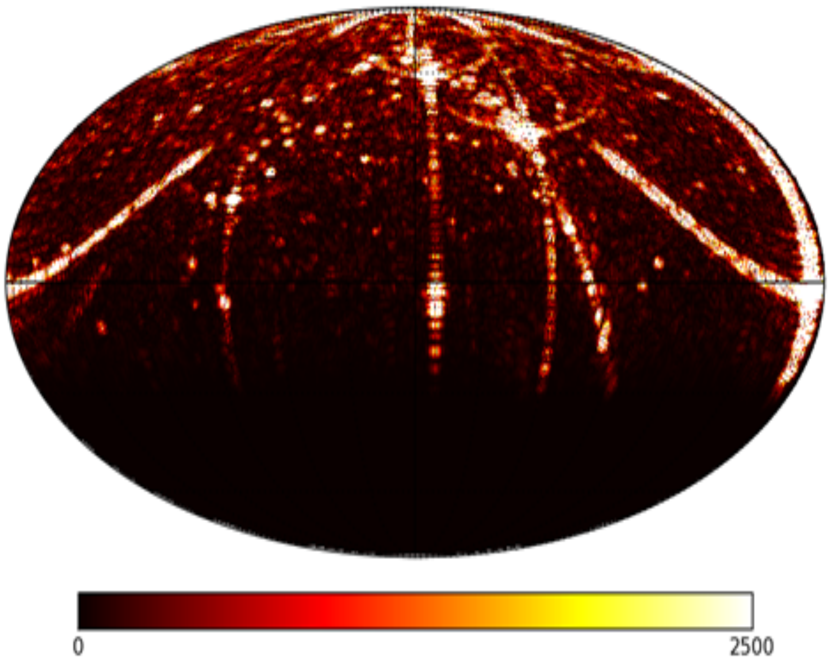
bright source calibration

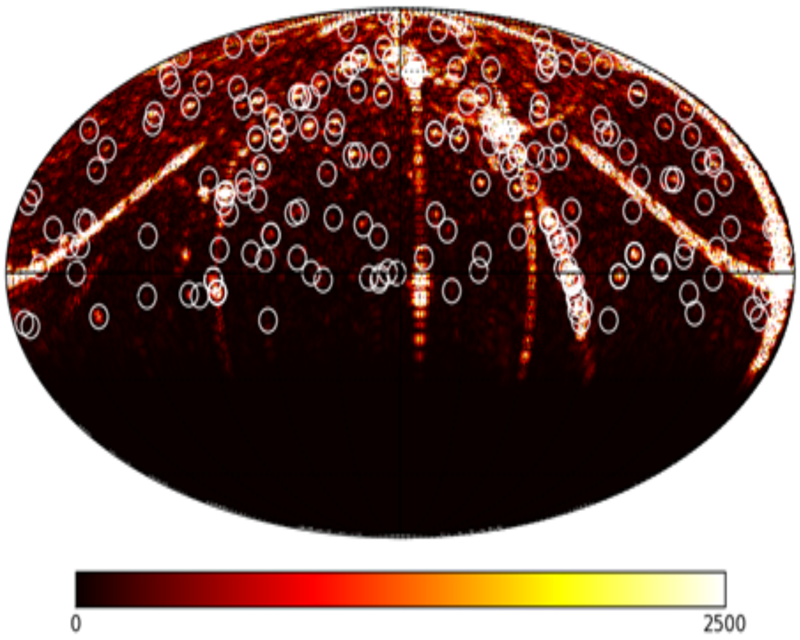


NAOC subtask2: Parallel pipeline system for Interferometer

SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PROCESSOR

Whole sky imaging for real observation data





Sky image made by 3-day data of cylinder array, only 5 frequency channels around 750 MHz is used.

Shanghai astronomy observatory (SHAO) tasks

- Completed task：

Complete production of SKA1-scale simulated data on Tianhe-2; implemented imaging software package on Tianhe-2 and demonstrated preliminary results on 2017 AU/CH SKA big data workshop.

- Task in progress：

Leading Sprint task TSK-344: Run DALiuGE on Tianhe-2

- Task to take:

To execute and verify SKA1-scale data processing simulation by integrating imaging pipeline into execution framework.

SHAO subtask: Regional Science Centre

SKA SDP Workshop, 2016 Shanghai

100+ researchers (20+ international), cross astronomy, HPC, industry
Sessions: SKA science, Regional Science Centre, Science Data Processor, and Prototyping

- Shanghai Observatory first proposed **SKA Asia Regional Centre** concept in the workshop

- PD (DG) - 1. data processing 2. synergistic support among members

- To strengthen **bilateral collaborations** in the framework of multinational project

SHAO subtask2: Prototyping - SKA data flow management

- Data-Activated Flow (流Liu) Graph Engine (DALiuGE) – Australia-China collaboration achievement !
- Deployed on Tianhe-2 1500 nodes, multiple computing islands, verifying the scalability of DALiuGE to 10 million tasks/drops => first-time large-scale SDP test, strong supporting for further integration and prototyping
- SHAO SKA team awarded 2016 "Milky Way Star"

World's largest telescope meets the second fastest computer


上海观察
Shanghai Observer 2016-9-9 星期五
首页 政情 财经 区情 城事 文化 天下

世界上最大射电望远镜核心数据管理软件首次集成测试完成

分享至： (1) (0) 收藏 作者:黄海华 2016-08-30 17:41:07

上海天文台安涛研究员说，下一步将考虑最高用10000节点（注：天河2号的极限能力是16000计算节点）开展全规模验证实验。

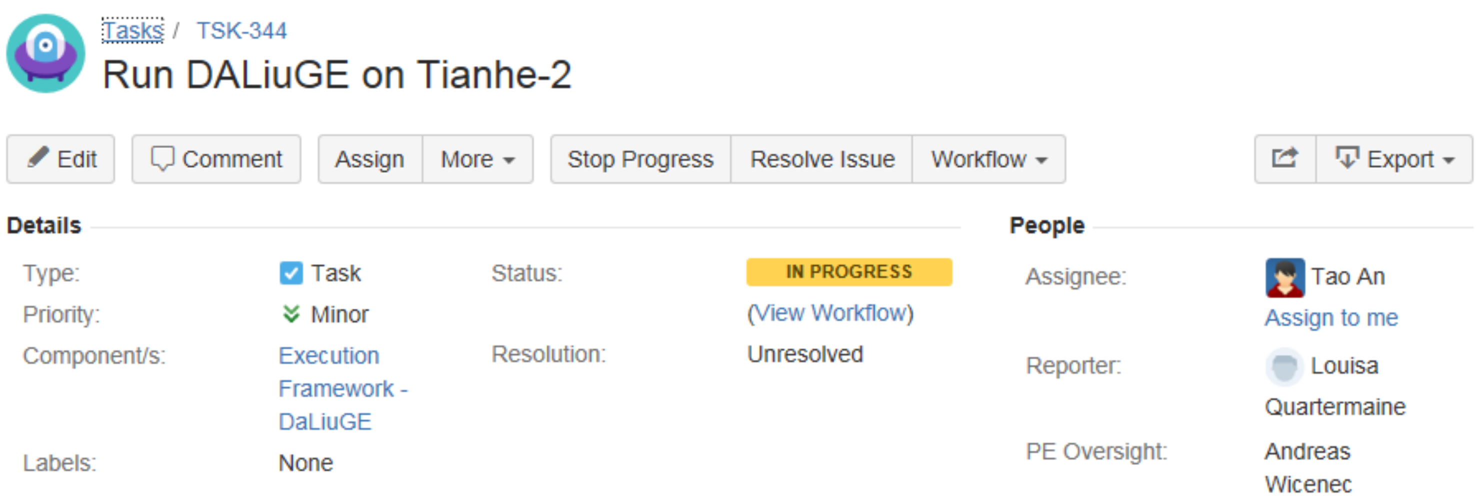
日前，上海天文台安涛研究员带领的SKA团队，在澳大利亚射电天文国际联合研究所和广州超算中心的协作下，在天河-2超级计算平台上成功部署了SKA数据流管理系统并完成了1000个计算节点的大规模集成测试，这是SKA核心软件首次完成大规模集成测试，为将来工程化验证提供了强有力的技术支撑，在国际上引起了广泛的关注和积极的反响，也得到SKA总部赞扬。

SHAO subtask2 Leading Sprint task TSK-344

Run DALiuGE on Tianhe-2

- Complete production of SKA1-scale simulated data on Tianhe-2; implemented imaging software package on Tianhe-2 and demonstrated preliminary results on 2017 AU/CH SKA big data workshop.

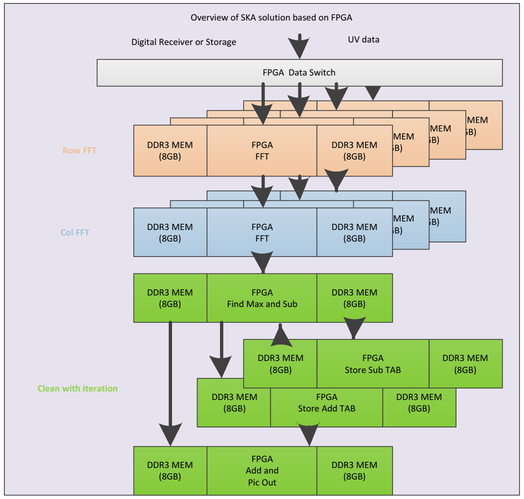
- **Next step:** To execute and verify SKA1-scale data processing simulation by integrating imaging pipeline into execution framework.



No.32 Institute of China Electronics Technology Groups Corporation (CETC 32)

Work involved: FPGA platform prototyping

- Upgrading from XILINX VIRTEX-6 to VIRTEX-7 to achieve double Power Efficiency
- Developing circuits of high performance buffer and network protocol circuits, improve storage and network performance
- Evaluation of FFT implementation on Mimicry computer
- SKA-SDP solution design on mimicry Computers



Sino-UK Joint lab of Henan Prov. (SUJLHP)

SKA
SQUARE KILOMETRE ARRAY
SDP
SCIENCE DATA PROCESSOR

Work involved

- **Analysis of algorithms on CPU+ GPGPU**
  - Convolution
  - 1D clFFT, 2D clFFT, 3D clFFT
  - SFFT
  - CUFFT
  - Reprojection
  - Gridding, DeGridding

Sino-UK Joint lab
中英联合实验室

NNEAC

- **ARL Imaging Pipeline on TensorFlow(TSK-1441)**
  - Cooperate to implement key operation of image pipeline under tensorflow framework
  - Mainly refer to crocodile
  - Building key operation library on GPGPUs

SUJLHP subtask: Algorithm analysis

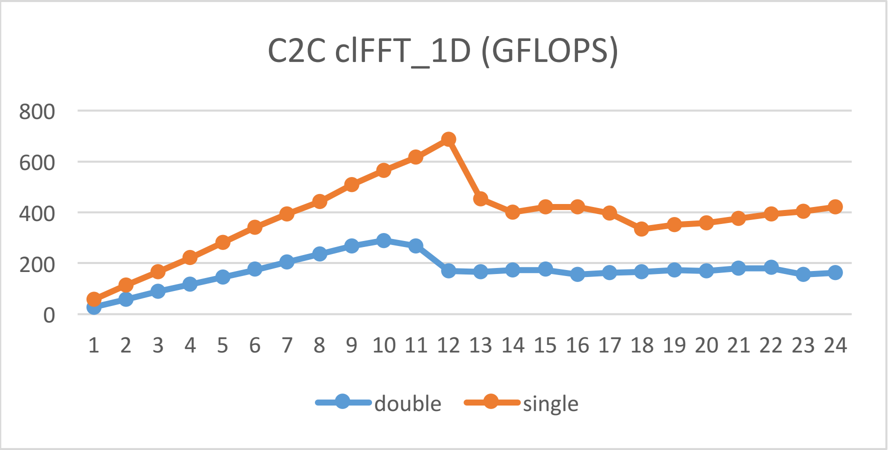
SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PROCESSOR

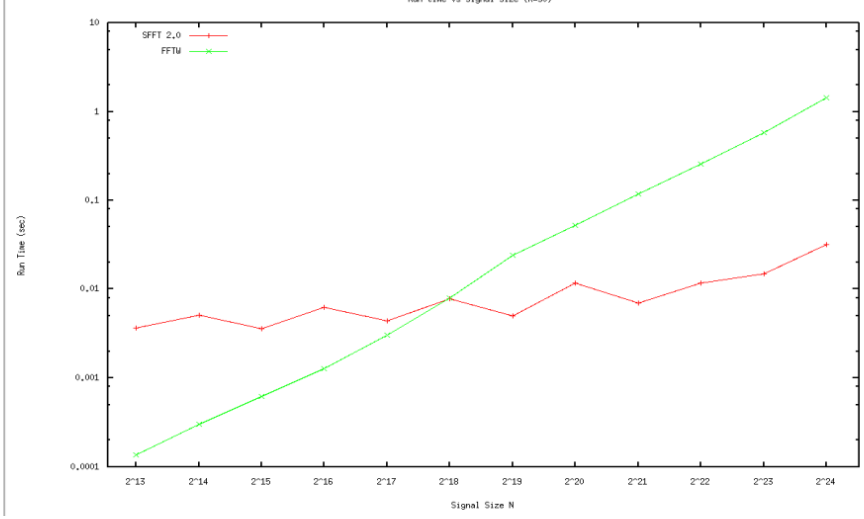
Performance measurement

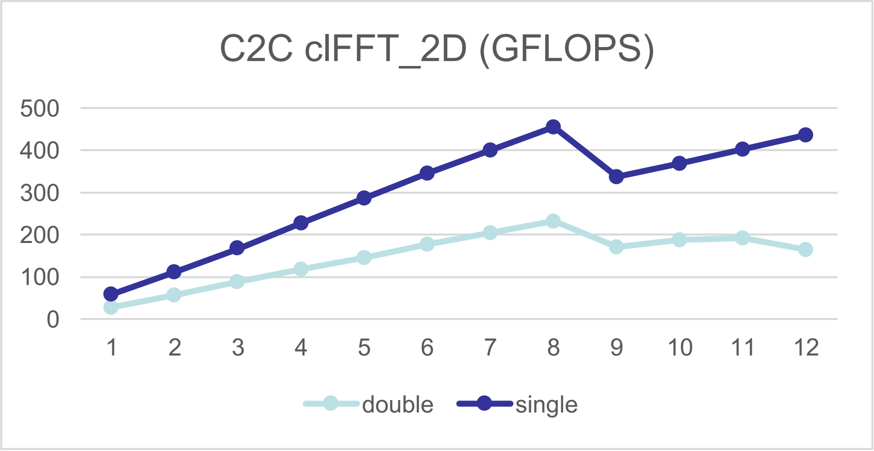
- Nvidia cuFFT （cuFFTBenchMark by Nvidia）
- OpenCL FFT (BenchMark with pycLFFT）
- FFTW 3.0
- SFFT 1.0

We found:

- **GPU's efficiency for FFT is less than 10% of its peak performance**
- **NVLink is not supported properly by intel/AMD**







Outline

I. Overview of SKA-SDP China Consortium
II. Progress of China SDP Consortium
III. Perspectives of China SDP Consortium for SKA Challenges
IV. Future Work

Perspectives of China SDP Consortium for SKA Challenges

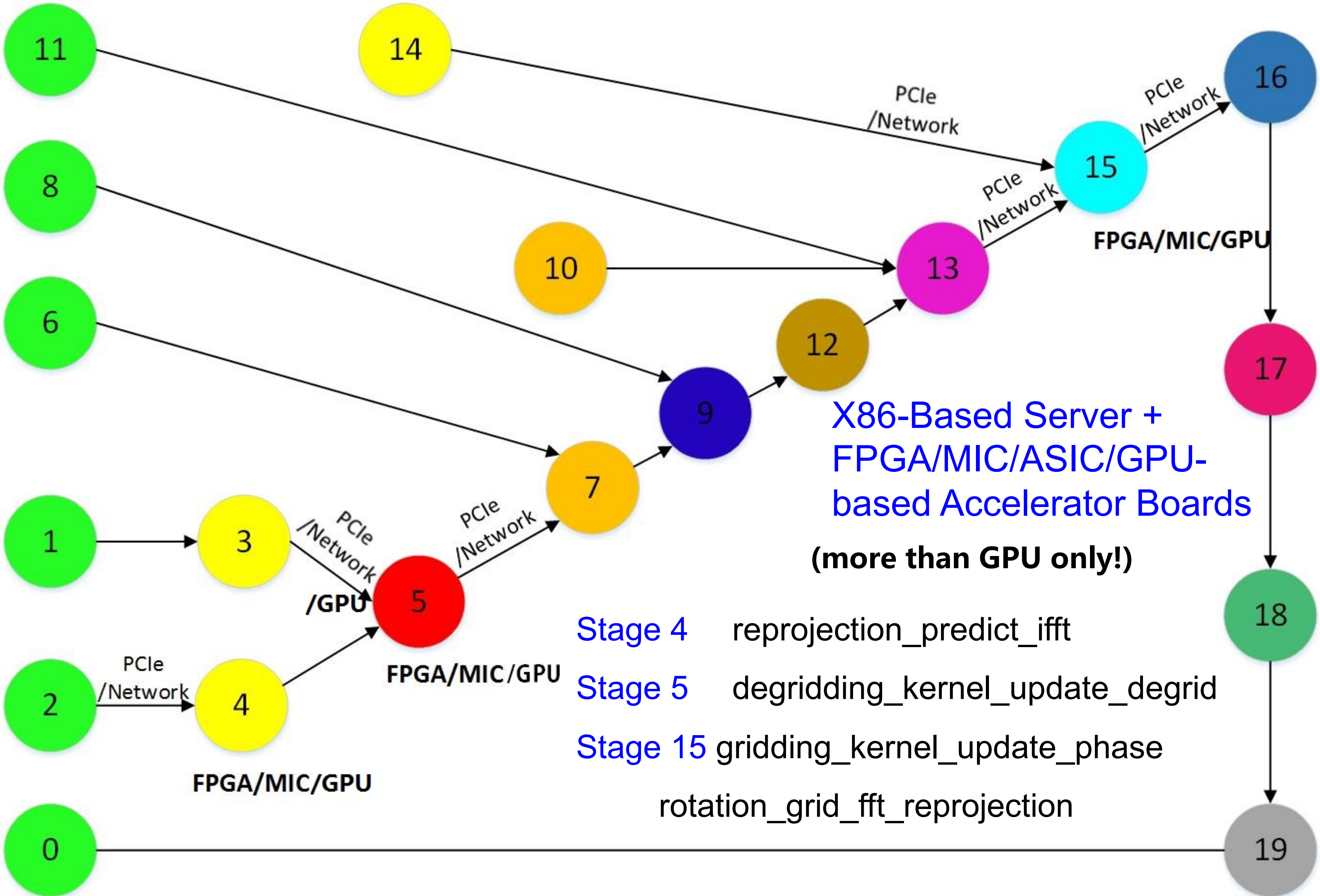
- SDP architecture has to be open since existing supercomputers cannot meet the requirements
  - Computation requirement is over 8x of top SC
  - Power budget is below 1/3 of top SC
  - Data bandwidth is over 1TB/s
- A science-software-hardware co-design approach is required
  - SDP imaging and non-imaging algorithms need to be refined with awareness of platform constraints
  - Software and hardware must be re-designed in a fusion way to meet the toughest requirement

Outline

I. Overview of SKA-SDP China Consortium
II. Progress of China SDP Consortium
III. Perspectives of China SDP Consortium for SKA Challenges
IV. Future Work

Implementing with a heterogeneous execution framework

SKA SDP
SQUARE KILOMETRE ARRAY
SCIENCE DATA PROCESSOR



X86-Based Server + FPGA/MIC/ASIC/GPU-based Accelerator Boards

(more than GPU only!)

Stage 4 reprojection_predict_ifft

Stage 5 degridding_kernel_update_degrid

Stage 15 gridding_kernel_update_phase

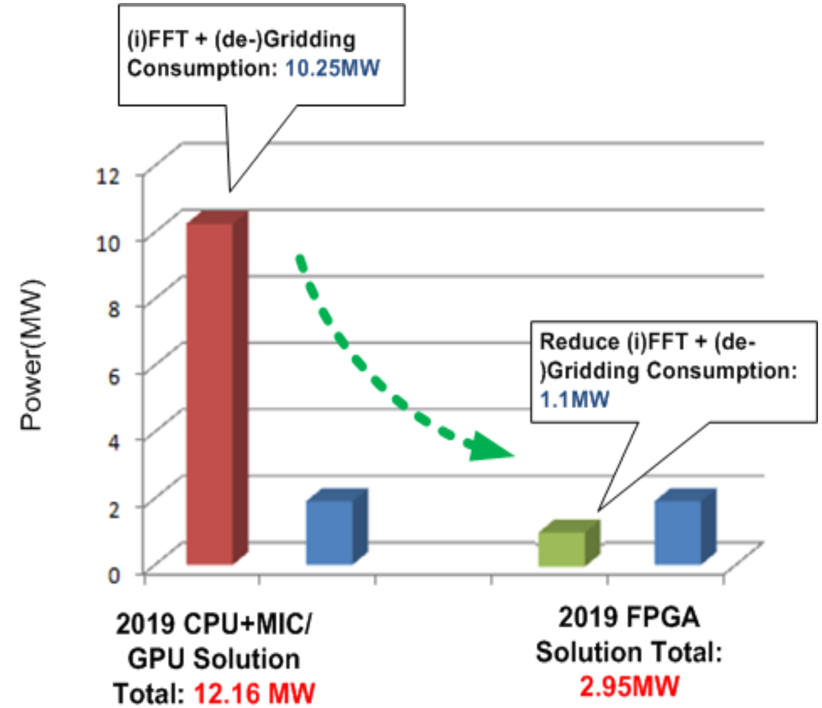
rotation_grid_fft_reprojection

Improving power efficiency with heterogeneous execution

Power Efficiency Improvements

	Currently TianHe-2 Status	Currently CPU+MIC/GPU Solution	2019 CPU+MIC/GPU Solution	2019 FPGA Solution
Computing Speed **(PFlop/s)**	54.9 PFlop/s	300 **PFlop/s**	300 **PFlop/s**	300 **PFlop/s**
Total Power **(MW)**	17.8 MW	97.3 **MW**	12.16 **MW**	2.95 **MW**

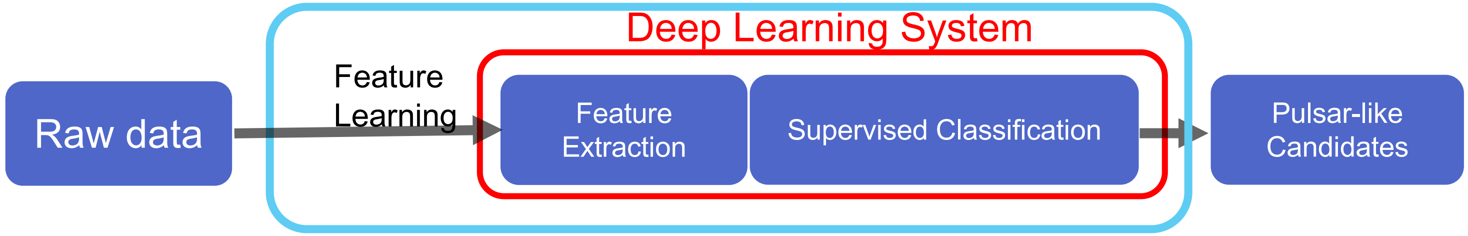
FPGA may reduce power within the budget



Estimated power of FPGA based architecture

A Case of Science-Software-Hardware Co-design

Automatic Pulsar Search using Deep Learning (PSDL V2)



Spark TensorFlow™

- Directly input raw data after FFT or other transforms to the DL system
- Design of more complex deep learning algorithms
- Execution framework: integration of Spark with the GPU platform, i.e., TensorFlow, Caffe, etc.

FPGA/MIC/ASIC/GPU-based Accelerators:

- Direct support of Tensorflow
- Direct support of SPARK

Thank you!