3DPAS April Meeting

Wiki: http://wiki.esi.ac.uk/3DPAS_Workshop

Participants:
Keith Jackson (LBL),
Jon Weissman (UMN),
Jessica Chen-Burger (University of Edinburgh),
Neil Chue Hong (EPCC),
Vinay Sudhakaran (University of Edinburgh) - vinaysudhakaran@gmail.com,
Yogesh Simmhan (USC),
Shantenu Jha (LSU, Rutgers)
Adam Barker (St Andrews)
Adam Carter (University of Edinburgh),
Omer Rana (Cardiff Univ.),
Dan Katz (University of Chicago, Argonne),
Kees Nieuwenhuis (D-CIS)
Steve Fisher (Rutherford Appleton Lab / STFC)
Gayathri Nadarajan (University of Edinburgh)
Paolo Besana (University of Edinburgh)

Day 1, April 4th, 2011

Efficient Processing in Real-time Complex Distributed Information Systems
Kees Nieuwenhuis
Thales Research & Technology

Challenges DIADEM project:

- create an ICT system for collaborative situation assessment and decision support that delivers more effective protection of the population and the environment against chemical hazards in industrial areas
- Methods and tool facilitating collaborative situation assessment and decision making involving automated reasoning and human domain experts
- SOA approach
- advanced GAS detection

Use Case: Greater area of Rotterdam

- A lot of industry, but also a lot of people in the same area
- There are some sensors in the area (but not a lot)
- Correlate sensor data with observations from people
- How is the gas spreading?
- Where are the zones with critical concentration?

- Use cases:
    - Acquire and process large quantities of heterogeneous information
    - cognitive capabilities of a single expert are not sufficient
    - full automation of assessment and decision making process is not feasible
- Information arrive via call center
    - “you have to ask the right questions”
    - Computer should generate questions according to a certain hypothesis
- Control room operators are not chemical specialist
- Construction Requirements:
    - professional bureaucracy
        - division of labor => heterogeneous services
            - start to automatically generate workflows...
        - decentralized control => distributed processing
        - generate domain knowledge is neither shared centrally nor standardized
            - you don’t know which expert you need
            - experts are not part of the same organization
            - health office, fire office - all have their expert - general no sharing of all knowledge
    - Data rich and dynamic environment
        - use of phone/email/sensors => quality, dynamics, data-intensive
            - no one sensor that is going to tell you all
        - avoid communication
- Process problem capture
    - We need:
        - a method for robust heterogeneous and hybrid fusion of information
        - the processing of which can be distributed
            - needs to take potential new data sources in account
        - dynamically adapts to changes
        - scale-free in both resourcing and processing
        - thus 3D
    - Environmental agency needs to process information within 30 minutes to potentially inform mayor (by law)
    - Different granularities of models?
        - coarse grained to get a fast analysis?
            - e.g. gas detection
    - There are gas leaks everyday - there is always some sort of escalation
    - Fukushima: “There are not a lot of models available that helps them”
- Dynamic Process Integration Framework (DPIF)
    - agents make resources autonomous and composable
        - asynchronous, data-driven processing in dynamically constructed workflows
    - communication engine
        - has to work with more abundant available telecommunication technology
        - Internet, gsm, cell phones

    - reasoning engine, graphical user interface
    - A software assistant for every expert
    - From local knowledge to global workflow
        - yellow page service
    - Collaborative reasoning in workflows
        - Workflow is basically a routing scheme included in the agents
        - Agents determine in what things they are interested in
    - Capabilities:
        - support very heterogeneous services
        - new services can be added to system continuously
            - no harm if there are two agents with the same knowledge
            - contradictions? Detecting faulty knowledge? No completely eliminated!
        - no need to know which services will be added prior to the operation
        - No centralized overview of service dependencies is necessary (and not practical and likely intractable)
        - Idea: dynamic, distributed decision support system
- Example:
    - Distributed Perception Networks (DPN)
        - breaking big model up into several smaller models that can be calculated separately
    - Distributed Detection System:
        - compound observation system models
- Future Work
    - Automated Pruning
    - Data mining and knowledge extraction
    - System-level protocols (yellow pages)
    - Querying, matching, ranking
    - Faulty resource detection
    - MILS security & deployment strategies
        - make sure security is handled on information item level
    - decision making interface
    - different communication kernels

Discussion Points:

- How does the dynamic data influence the inference process?
- Is this a completely new class of application?
- Problem with deadline-driven approaches:
    - simulated data is for some models good enough
    - not all data is always useful
- No everything can be automated - human knowledge for certain decisions necessary
- Data privacy sufficiently considerred?:
    - privacy policies of owner are taking into account during processing
- Auditing?

    - logging in place:
        - What agents are active?
        - …
    - Playback of datasets that have been in place at a certain time
- State:
    - one version operating at the harbor of Rotterdam
    - roll-out to Denmark planned
    - project will end at the end of 2012
- Relation to 3D:
    - No as large scale as LHC, but...
    - 3D not solved in this kind of decentralized environment
    - a lot of local autonomy
- Standardized data formats:
    - no - first need to standardize communication between human experts

Talk 2
How to efficiently process large videos? A workflow perspective?
Gayathri Nadarajan
http://fish4knowledge.org

Link to Gaya's thesis: http://homepages.inf.ed.ac.uk/gnadaraj/thesis/nadarajan-thesis.pdf

- Partners: U. Edinburgh, CWI, Catania, NCHC (Taiwan)
- Aims:
    - automatic storage and extraction of information of obsverved marine life from database
    - provide interface for non-expert users (marine scientists)
- Facts:
    - >3000 species identified
    - continous collection: 24-25k videos & counting, 8fps, 20-50 MB, 20 GB per hour, 100 TB per year
    - 10 static cameras, resolution 680x480
- 20 questions:
    - How many species appears and their abundance and body size in day and night include sunrise and sunset period?
    - … in certain periods of time (day, week, month, season per year)?
    - rank species (most frequently seen, very rare)
    - fish activities with one day
    - population size changes
    - relationship with environmental factors
    - fish size (requires a stereo camera)
- very challenging question also with respect to image processing demands
    - very colorful

- How can the project influence the used technical infrastructure?
    - for some questions technical infrastructure can be defined?
- System Architecture
    - Database
    - Fish processing (not the kind that leads to fish in cans)
        - Fish tracking (counting)
        - Fish description (show me fish that look like...)
        - Fish recognition (species identification)
        - Fish clustering
    - Workflows
    - Query engine
    - Capture device
- Workflow Component:
    - deal with special queries
    - manage/schedule jobs/processes
    - live streams
    - considerations:
        - distributed/parallel processing
        - batch processing
        - data streaming instead of job scheduling to minimise volume
    - special queries:
        - queries need to be formalised into machine readable VIP goals
        - done via ontology and planning
            - ontology has a list of goals
            - constraints: can you do this goal quickly?
        - human is not completely removed from the loop
            - the amount of data a human needs to process is reduced
            - semi-automated approach
        - Specification:
            - Here is a list of goals? which ones do you want?
    - Optimal job scheduling
        - VMWare on Demand platform
        - Which algorithms should be parallelized?
            - Fish detection takes the longest: 20 min (expected), 60 mins (worst case)
        - Algorithms are both data- and compute-intensive
    - compute options:
        - shared memore architecture (48 cpu smp system)
        - run code on one VM system
        - run code on several VM systems: individual and distributed memory
        - replicated data? yes
    - Adaptation for dynamic data:
        - live streams
        - adaptive workflows

        - cloud solutions, e.g. Hadoop for dynamic scalability

Discussion:

- What can we reuse in another context? e.g. with a different sensor source
    - Planner is domain-independent
- What is dynamic?
    - What can be done offline? Does it need to be done in real-time?
        - a bulk can be done offline
    - correlations between queries, queries could be refined with each iteration
    - mobile & multiple cameras
    - Analogy with digital sky survey?
- What is distributed?
    - Camera are distributed?
        - distributed vs. remote
        - just remote because they are feeding into the same storage
    - Is compute co-located with data-source?
    - Can queries be run remotely?

Application Discussion Session
Dan Katz

Application from White Paper:

- Apps w/ local data
- w/ local and remote data
- w/ remote data (files)
- w/ remote data (streams)

Local Data

- Sivia's Bioscience app
    - data not really dynamic
    - data comes from sequencer
    - pipeline of processing steps (workflows)
    - Discussion:
        - dynamic due to fast generation of data?
            - analysis must be done quickly
        - are there differences between the stages?
            - raw data is annotated
            - data is not changed, but the relationships to other data

- Simon Sensor network app
    - Data is generated and stored in a central archive
    - i.e. distributed part is not essential for the way users use the app
    - dynamic queries

- similar to fish app
- Discussion
  - dynamic aspects:
    - generation of data
    - infrastructure might need to dynamically scale
    - Is dynamic infrastructure different from scalable infrastructure?
      - dynamic == you don’t know which capacity you need
      - timescale: what is the response time required?

Local/Remote Data

- Jon’s Environmental App
  - data from sensors and/or simulations
- Power Grids (Shantenu)
  - needs to combine streaming data arriving from sensors
- Astro (CMB)
  - data from detectors
  - data perform monte-carlo simulations
  - combinations from raw detector data and simulations
- Astro (LSST)
  - data from 1 or more telescopes
  - quick pre-analysis
  - data that is taken is archived

Remote data (files)

- ESGF (Don/Dan)
  - data is distributed across multiple stores
  - data is transferred to a local resource where analysis takes places
- WLCG
  - data is stored in a hierarchy of infrastructure
    - Level 1, 2 - 3
- Adam SOA astronomy application
  - data comes from remote files/db
- Bob’s astrophysics (VO)
  - data stored in registries w/ metadata
  - pub/sub for dynamic events
- Bioinformatics
  - compare O(1000) of terabyte-scale data sets

Remote data (streams)

- Fusion
  - multiple pyhiscs codes running concurrently on a distributed set of parallel computers
  - data from some codes is streamed to other codes to link them into a single simulation

    - data transformed inflight
    - data examined in situ
    - Discussion:
        - At what state must data be co-located to perform a simulation
    - DOE assumes that all the data that is generated cannot be written out to disk (too expensive)
    - real-time analysis of sensor data necessary
- Kee’s SOA app:
    - all data is coming from real-time sensor
    - how much of the data you should store unchanged?

Discussion

What’s important in these applications?

- streams vs. file
- remote vs. local vs. both
- processing data as streams vs. processing archived data
- files vs. databases
- data trains vs. stream standing queries
- compute intensive vs. data movement intensive
- for compute intensive work: who pays?
    - user, service-provider
    - institution pays

What are the three axes?

- dynamic, distributed, data-intensive
- diversity of data (how many types of data are produced/consumed)?

Railway schedule re-scheduling:

- potentially distributed processing
- but results need to brought back and aggregated
- local vs. global optimizations

Security Log Analysis / Intrusion Detection:

- TODO: Jon to fill in questions for app

Number of concurrent streams (could be same source)

Questions:
What are secondary questions that drive whether I want remote or local access?
What drives the users? What is their concern? What restricts their scalability?
Policies? Are budget restrictions constrains?

Omar

Discussion
Application Vectors

Dynamic data:

- rapidly changing (vs. query evolution)
  - both content and structure
  - rate of change of data
- variability of data rate

Distributed data

- out of control
- not in one place
- same as DPA theme

Data-Intensive

- Large datasets

Proposed Vectors/Stages/Clownfish:

- Data Ingest
- Data Analytics
- Post-Processing

This is primarily a workflow view.

Do these vectors make sense?

- Example: Ingest could be described as follows:
  - manage multiple streams (of events)
  - rate of data, jitter

These vectors are just phases: input - processing - output?

- not phases => component

Captures only data-intensive - not dynamic
What is part of the system?

- Is the storage part of the system?
- live streams are not part of the system

Shantenu: Vectors should spawn orthogonal characteristics
It might be the case that not all applications that don't have all 3Ds? Do these vectors make sense in this case?

Are vectors associated with 3D or with application?

Example applications:

- **Kees - emergency response**
    - Ingest
        - sensor data (physical sensors and people)
        - dynamic:
            - you cannot predict which sensor fires when
            - you don't know how accurate the signal is (signal-noise ratio)
            - you don't know whether sensor is available in the next moment
            - new sensors suddenly become part of your workflow
        - events not raw data
        - concurrent streams
        - How does the ingest work?
            - physical sensors come in via some kind of electronic media
            - data is stored in DB
            - People data comes in via telephone and is transcribed and filtered by the operator. Data is then stored in DB as well.
            - Querys are generated by system
                - Processing
                    - Questions for caller is generated by system. Retrieved by processing agent
                    - automated processing system generates query that can be sent to an individual with cell phone (YES/NO answer) - automated analysis
                    - Also stored in central DB
        - XML, SQL + proprietary for http://opencv.willowgarage.com/wiki/ mats
    - Analytics
        - Situation assessment (reasoning network) to formulate hypothesis
        - use of a distributed perception network with nodes being data from ingest
        - Diversity of data not pre-determined (data models may vary depending on the experts involved)
        - Dynamism:
            - dynamically constructed network based on hypothesis
            - dynamic interaction (coordination) between agents (expert agents, humans etc.)
            - dynamic: number of components involved unknown, location of nodes unknown
        - Coupling between data ingest and analytics (to improve accuracy - "cooperative capability" of the system - to focus attention on a specific geo-spatial location - managed through an operator (i.e. manage more sensors, location of sensors)
        - AgentSpace/AgentScape??? software is used, http://www.agentspace.org/ ??

        - DPIF (Belief Networks) - e.g. Bayesian Networks & other knowledge encoding networks
            - update frequency?

    - Post-Processing
        - Plume detection and origin tracing (identify location of plume, coverage of area, calculate concentrations, advise on possible actions)
        - identify more actions - decision support:
            - gather more data
            - evacuate people

- Fish4Knowledge
    - Ingest
        - Multiple concurrent streams (FLVs -> MPEGs)
        - variable rates (8 fps -30 fps), resolutions
        - deal with raw data (no pre-processing or events)
            - 1-3 min long video clips
        - centrally managed NAS file system (move to local disk or remote disk)
        - database planned in future
        - camera type can very (CCTV, HDV) - variable quality (color, brightness, size etc.)

    - Analytics
        - OpenCV

Neil
Infrastructure

- What are the application requirements for each stage?
- Take instances of applications currently collected and for each stage show how it utilises the infrastructure
- what does the infrastructure provide at the moment
- What should the infrastructure provide in the future:
    - what does it do
    - what does it provide
    - what the application would the infrastructure to provide?
- tabulate data
- identify areas of missing infrastructure

- identify infrastructure needs on app basis or on basis of app vectors?
- what is the right level to abstract the infrastructure for the 3DPAS?
    - programming abstraction, storage abstractions
    - development, deployment and execution abstractions (covers programming/

    storage abstraction which would be in development)
        - deployment:
            - "setting up", "(re-)?? configuring"
            - just stuff that happens at t=0, i.e., no reconfiguration?
            - constraint to be applied before execution
            - e.g. setting up a cloud-like execution environment in HPC environment: message queues, tabulated storage,...
    - reconcile level of abstraction with whatever we are calling vectors
    - collect from applications by understanding not just the systems but what they are doing
        - infrastructure abstractions
        - must capture where applications have rolled their own?
            - e.g. pilot jobs for sensor type environments?
        - architecture diagram for each application?

Infrastructure Taxonomy

- Data
    - Movement
        - Partitioning / planning
        - Staging options
    - Data placement
    - Code placement
    - Communication (whilst program is executing)
        - Messaging
    - Streams
    - Storage
        - Durability
    - Access
- Logging
- Security
- Computing
    - Event detection
    - Stream processing
    - Semantic Information Integration
    - Filtering
    - Image processing
- Management / Choreography
    - Discovery?
    - Information Services
    - Workflow
    - Adaptive sensing?
    - Scheduling ability (infrastructure) / Scheduling approach (application)

How do we relate particularly relevant patterns observed in the applications and systems to vectors and reinstantiate this at the infrastructure level?

DPA Vectors

- Execution Unit
- Communication
- Coordination

3DPAS Vectors

- “Data planning / shape”: how the data is coordinated from ingest through analysis to output Communication + Coordination
    - e.g. LHC experiments exhibit a single point source to multiple analysis followed by mass exchange
    - Sensor grids have multiple sources to single analysis followed by constrained point of reuse
- “Application Composition” “Adaptation” - Execution Unit + Coordination
    - Adaptive applications, choice of data sources
    - Mobile agents
    - Late binding

Look at the data connectivity pattern...
Shape gives sense of distributedness; thickness of edges gives sense of data intensiveness; change in the shape and thickness (and rate) and number of edges gives sense of dynamism