"Embryo": an autonomic co-operative service management framework

Fabrice Saffre[1] and Mark Shackleton[1]

[1] BT Group plc, Pervasive ICT Research Centre, Adastral Park, Martlesham Heath, Ipswich IP5 3RE, United Kingdom
fabrice.saffre@bt.com

Abstract

In this paper, we present "Embryo", a fully decentralized service management framework inspired by morphogenesis and capable of installing components and modifying the topology of a peer-to-peer (P2P) interaction overlay network so as to meet the needs of the majority of all participating peers. Co-operation is an emergent property of the self-organisation process, which is underpinned by purely "selfish" decision-making based on incomplete information gathered through gossiping (local messaging). We provide a detailed description of the local reasoning loop governing the behavior of individual peers, as well as Monte Carlo simulation results that demonstrate the system's ability to converge to a stable state in which most peers have direct access to all the components they require via one of their first neighbors.

Introduction

The last few years have seen a huge number of papers on so-called "complex networks" in both natural (e.g. social, ecological, genetic) and artificial (e.g. power grid, communication) systems [1]. The main reason for the "vitality" of the field is arguably that many scientists have come to realise that the combination of graph theory and complexity science could provide them with powerful tools, provided that the specific problem they are trying to solve is translated into a set of vertices/nodes connected by edges/links. To a large extent, whether the nodes are genes, species or routers and the links chemical reactions, predator-prey interactions or fibre optic cables can be considered as irrelevant when the system is described as a complex network, which makes the associated investigation techniques very widely useful indeed.

It has been argued however that the dynamical properties of complex networks have been somewhat neglected and that maybe too much emphasis was being put on descriptive analysis (e.g. the search for power laws [2]). It is a fact that the mechanisms leading to the emergence of the various network structures have not always been studied in appropriate detail, and that some fundamental aspects of network growth, evolution and decay have yet to be explored. To some extent, the influence of dynamic node properties on network genesis is the focus of this paper.

Understanding the "history" of networked systems (i.e. of how they came to exhibit a particular structure) is of particular significance in the case of overlays, i.e. virtual or logical networks superimposed on (and usually not paralleling) the underlying physical infrastructure. Indeed, being "immaterial" entities, overlays can reorganise themselves very quickly and easily, potentially resulting in macroscopic topological changes occurring over a short period of time. Since in many cases, rewiring in overlays is based exclusively on locally available information and only involves P2P interactions, understanding how a given structure can emerge requires careful examination of the local rules governing decision making and/or information sharing. If the objective is to build an overlay network to serve a particular purpose, then the problem becomes to engineer those rules so as to generate and maintain a desirable configuration.

Some remarkable work has recently been done in this area by Jelasity and Babaoglu [3]. These authors propose a fully decentralised, gossip-based rewiring method (christened "T-Man") to organise a vast set of nodes so that each one of them rapidly "migrates" to the correct place in the whole (initially random) web of relationships. In T-Man, the "correct place" is entirely determined from the start by some static, node-specific information (which can be assimilated to a unique ID in an addressing system). There are many situations however in which the local property that the self-organising process is concerned with will simply not be a constant, due to individual elements changing all or part of their characteristics at the same time that the web of interactions is being reconfigured.

This problem is very similar to that found in morphogenesis (biological development), where individual stem-cells simultaneously differentiate (i.e. specialise) and move in space (equivalent to rewiring in a network) until cells of the right type occupy the right location in the developing organism. Indications are that the type of each cell is not assigned from the onset, but that differentiation is the result of signalling: in effect, neighbours influence each other's "choice" through a dynamic web of positive and negative feedbacks, the structure of which varies due to physical relocation of individual cells [4].

We used this aspect of the developmental process as a source of inspiration because we find it to share many characteristics with co-operative peer-to-peer (P2P) service provision, a type of application that is very different from content-sharing, as it is typically characterised by lower diversity (there are fewer different service components than there are unique files in a typical file-sharing community) and more subtle interaction patterns (peer activity isn't limited to propagating queries and uploading/downloading content). Note that biologically inspired approaches have been applied to content distribution by other authors (see e.g. [5]), but that this problem is outside the scope of our paper.

In P2P service provision, components are distributed across a collection of processing nodes, each of them hosting only a sub-set of all services locally and relying on different nodes to satisfy other needs, via remote access. In terms of the developmental biology metaphor, deciding which service to host is equivalent to differentiation (with the same complex, non-deterministic nature, due to the quality of a choice being primarily a function of other units' decisions), while identifying and selecting suitable providers through rewiring of the overlay network is somewhat similar to physical migration.

Motivation and related work

The service management framework presented in this paper is motivated by the need for autonomic (self-managing) solutions that support future service component-based systems. Two areas where such autonomic solutions are required include Web Services and Pervasive Computing.

Web Services [6, 7] encapsulate both a technology and an industry trend towards distributed, component-based business solutions, where an application is realised by linking many individual services together into complicated workflows and higher-level composite services.

Related to the Web Services domain is the Open Grid Services Architecture (OGSA) architecture that is being developed by the Global Grid Forum (GGF). The aspiration of this work is to support the distributed interaction and interoperability of large numbers of component services in order to meet the needs of users, especially in the eCommerce and eScience domains [8]. Some of the technical challenges arising in realising such solutions are highlighted in Ian Foster's seminal paper "The Physiology of the Grid" [9].

Pervasive Computing is similarly a combination of an overarching vision [9], industry trends and supporting underpinning technologies, and is of sufficient importance to merit its own journal [11]. The vision here is that of huge numbers of communicating devices embedded into the physical fabric of everyday life, all of which need to be marshaled to provide services effectively and efficiently. An interesting example of an adaptive approach to realising a service provision framework in a pervasive computing context is given in [12, 13].

Taken together, these technology trends provide huge opportunities, but they also exacerbate problems which already plague ICT systems, namely the complexity of such systems arising from huge numbers of interacting component parts and the cost of deploying and managing the behaviour of such systems. IBM have probably captured the challenges most clearly and succinctly by launching the Autonomic Computing initiative and associated challenges [14].

Given that future applications and services will almost certainly be realised by coordinating the actions of relatively fine-grained component services, the question arises as to how this might be achieved. Clearly the component services need to be encouraged to cooperate together to provide higher-level services, and given the envisaged scale and complexity, much of this cooperation needs to be inherent, or "engineered into", the underlying system as self-organising principles. This paper presents a solution that is a step along that path.

The need for self-managing approaches for Web Services has been acknowledged by the Open Grid Services Architecture (OGSA) in their stated objectives for "self-management services" and "service level managers" but it is recognised that this work is at an early stage and is currently more aspirational than actual [8]. The approach currently being taken is clearly motivated by that advocated by IBM as the "autonomic management" control loop [15] - in essence this relies on a "generic control loop pattern" comprising monitoring, analysis, projection and action phases [8]. While this pattern will clearly form an important element in many autonomic systems, in this paper we advocate autonomic approaches that are more "inherent" in the design and behaviour of the system as a whole. This has the additional benefit of producing a more lightweight architecture.

In fact our approach is much more closely allied with the techniques and algorithms arising from a rather different community whose research is sometimes labelled as "self*" and often draws upon highly interdisciplinary concepts and models, such as biologically inspired autonomic solutions [16, 17]. Of particular interest to us is the work of Babaoglu who has drawn heavily on biological inspiration to realise a number of highly effective self-organised solutions to P2P data storage and routing, such as T-Man [3]. This and related work provides decentralised solutions and algorithms that are capable of maintaining reliable "overlay networks" over which service can be delivered, even when many component nodes may be individually unreliable [18].

This paper builds on solutions such as those advocated by T-Man, using similar self-organising principles to maintain an overlay network, but also allows individual nodes (service providers/consumers) to dynamically change their "type" in response to perceived demand in a fashion that provides useful autonomic features that support desirable system-level behaviour. While much of the existing research has focussed on data sharing over P2P overlay networks [19, 20], our focus is rather on the provision of a richer set of services via an overlay network that helps coordinate the cooperative behaviour of many individual service components. In this sense it is more closely related to the Chameleon system for self-organised and decentralised P2P web services [21].

Finally, we should stress that ultimate aspirations of both the Self* and Web Service research communities are in fact quite closely aligned. In practice we expect the real benefit will be most effectively realised when solutions such as those presented in this paper begin to be combined with those arising from initiatives such as the Open Grid Services Architecture in a truly interdisciplinary fashion.

Basic Principles

Like T-Man, Embryo uses exclusively gossiping to propagate information throughout the network and individual nodes can choose to swap neighbours based on whatever they learn about their counterparts through this process. Unlike in T-Man, they do not select neighbours based on a static identifier, but by trying to establish a set of symbiotic relationships with partners whose "specialty" complements their own at the time when the link is created. Because individual nodes can subsequently choose to stop hosting a given service and start hosting another (i.e. change type), their

relationships can become unsuitable which will eventually initiate a rewiring process. These dynamics, resulting from multiple nodes changing type and neighbourhood links concurrently, are very analogous to those found in morphogenetic systems.

For simplicity, even though these limitations would probably not apply as strictly in any real co-operative P2P system, Embryo cells are assumed to host only one service at a time (i.e. they cannot simultaneously perform several functions) and to have a fixed maximum degree (i.e. they cannot create additional links, meaning that establishing a new connection requires terminating another one). We also make the implicit assumption that link cost and/or capacity is homogeneous throughout the system, and so that rewiring of the overlay doesn't in any way affect performance (i.e. two nodes hosting the same service are equally capable of providing that service to any other peer)..

The local reasoning loop is similar to the one we used in previous work [22]. Whenever a node requires a service that it is not hosting itself, it first determines if it already knows a provider. If it does, it just sends a request to this particular neighbour. If it doesn't (or if the known provider fails to provide the service due to having changed type since the link was established) the originator of the request tries to identify an alternative provider. Unlike in our previous work however, this is not done by broadcasting a request, but by searching a locally kept stack of "adverts".

Adverts form the basis for the gossiping system of Embryo. Basically, every time that a node fails to identify a suitable provider, it prepares an advert specifying its own identification, which service is needed and which one it can provide in return (i.e. its present type). Adverts are periodically exchanged between neighbours, and propagated for a fixed amount of time (as in many P2P systems, this "time-to-live" mechanism is used to stop traffic from increasing indefinitely, by ensuring that outdated requests are discarded). Before forwarding adverts, every node keeps a local copy, building itself a partial but expanding and regularly updated picture of the offer and demand throughout the system (the adverts stack).

When searching for a provider, a node sequentially examines the adverts (most recently received first), looking for one that contains a requests for its own current type and offers the service that it needs in return. If such an advert is found, the node contacts its poster and a handshake procedure is initiated, which will only succeed if:

- Both nodes still have symmetrical needs (i.e. the poster hasn't changed type or found another provider since the advert was created).
- Both nodes have a spare (i.e. currently unallocated) or useless (i.e. connected to a peer that doesn't provide a necessary service) link.

If it does, the new co-operative link is created.

If a node consistently fails to identify a provider for a particular service, it may choose to take the radical action of discontinuing the service that it is currently hosting and replace it with the one for which it has not been able to find a "collaborator". This obviously creates a crisis for the node and for its neighbours, as it instantly makes all existing symbiotic links obsolete (since the node that has just changed type will no longer be capable of providing the service for which they were established, making the relationship useless for its partners). The underlying assumption is that this crisis can and will be resolved by a cascade of other modifications (of the overlay's topology and/or of individual nodes' specialty), and that the change of type will contribute to increase availability of the incriminated service.

The key difference between Embryo and T-Man is therefore that in the former, the self-organisation process involves both rewiring and modification of local properties. In other words: a node can migrate towards a location in the network where its current type is needed, change type in an attempt to turn itself into a kind of unit more appropriate to its current location, or even combine both procedures.

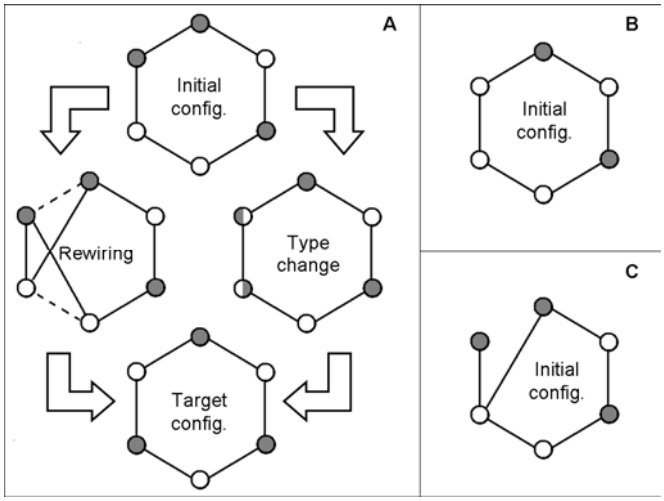

Fig. 1. Illustration of how "rewiring" and "differentiation" processes can lead to the same target configuration (A) and potentially problematic initial conditions (B and C).

Figure 1a illustrates, in a particularly trivial case, how the two processes can produce the same result. If the system was managed exclusively via T-Man "rewiring", it could only follow the left-hand path, whilst if it was relying on differentiation only, as in our own previous work [23, 24], it could only follow the right-hand path. The advantage of being able to combine both methods as in Embryo is made clear by the fact that, had the initial configuration been the one shown in fig. 1b, it would have been impossible to reach the target by rewiring only. Similarly, if it had been 1c, type change alone would have been insufficient.

Only with Embryo is it guaranteed that the exact target configuration can be reached from any of the initial configurations shown on fig. 1, and it is clear that the further the initial conditions are from the desired system state (topologically and/or in terms of node type distribution), the more potentially useful it is to be able to combine rewiring and differentiation.

Detailed algorithm

Our simulated implementation is based on a P2P architecture in which every node is expected to provide and request services to/from its counterparts. As a result, all units are currently governed by the same decision rules, even though it is relatively straightforward to introduce variants in

which parameter values (or the rules themselves) would be specific to an individual and reflect its unique constraints and/or requirements.

Decision making

In the present state of Embryo, every node is assumed to perform only one function at a time, i.e. it cannot simultaneously host/provide more than one set of services (assimilated to belonging to a specific "node type"). As a result, acquiring the ability to perform a new function requires changing type (which implies losing the ability to perform the previous one). However, this simplification was introduced for clarity only and is not a fundamental limitation of the proposed algorithm (qualitatively similar collective decision dynamics could be obtained even if every node was capable of performing several functions, i.e. belonging simultaneously to multiple types).

The functions to which a node needs access are assumed to be completely identified locally (i.e. every participant "knows" its own needs). Whenever it requires a function performed, the node behaves as follows:

- If it already knows (i.e. is connected through the overlay to) a provider, i.e. another node belonging to the right "type", it sends a service request.
- If it knows no provider (or if the provider fails to answer the request for whatever reason), it looks through its locally kept list of adverts to identify a suitable candidate and initiate contact (see "messaging").
- If it doesn't find a compatible advert or if the handshake fails (for whatever reason, see "messaging" for details) it has 2 options:
    - prepare a new "advert" to offer a partnership (see "messaging")
    - turn itself into the requested type and become its own provider.

The last action is the basis of the specialization process. The decision by the node to turn itself into the requested type is probabilistic and is based on its perception of the corresponding function's availability. The probability P of changing type obeys:

$$P = 1-1/(1+(x/x_c)^{\alpha}) \qquad (1)$$

where x is the number of failed requests for that particular service, decremented (incremented) by one at each successful (unsuccessful) attempt, and re-initialised to zero if the node turns itself into the corresponding type, and xc and α are parameters. The choice of function is arbitrary and another could have been used instead (preliminary results only suggest that it should be a sigmoid, which confirms intuition for those familiar with similarly self-organising systems found in nature). For the purpose of the proof-of-concept simulations, we used $\alpha = 2$ and $x_c = 4N$ (where N is the number of types).

We have also experimented with variants in which the probability that a node changes type also decreases with the number of such "metamorphoses" that it has undergone already. For that purpose, we used the following transformation from P into P*:

$$P^* = P\, e^{-\beta y N} \qquad (2)$$

where y is the number of previous type changes, N is the total number of types and β is a (positive) parameter. However, to facilitate interpretation of the results, this modification was de-activated in the version used for the simulations.

Messaging

Embryo relies on gossiping along co-operative links in the overlay network to propagate information about system state. Every node keeps a local list of the so called "adverts" that have reached it, indexed first by the function that they offer, second by their "age" (newest on top). An advert contains 4 distinct pieces of information:

- The type of the sender at the time when the advert was generated (i.e. the service/function on offer)
- The type/function requested in exchange (i.e. the service needed by the sender at the time when the advert was generated)
- The unique ID of the sender (e.g. an IP address or computer name)
- A timestamp

At irregular intervals (probabilistic decision), a node opens its "inbox", where new incoming messages are stored between inspections. It is a rule that, when opening an advert message, the recipient immediately checks whether the local list already contains an entry from the same provenance. If it does and the new advert's timestamp designates it as more recent (which isn't necessarily the case, as a newer advert could have arrived first if following a different gossiping route), it replaces the older one. As a result, there can never be more than one entry per node (sender) in the local list of adverts.

Whenever a node receives an advert that modifies its own local list (i.e. new provenance or new offer/request from an already identified source), it also creates a copy in its "outbox". The content of the outbox is forwarded to the node's neighbours (in the co-operative overlay network), also at irregular intervals (probabilistic decision). Constraints can be imposed on the number of messages that can be sent to every neighbour in order to accommodate link capacity. Also, a time limit can be added so that possibly "outdated" adverts do not unnecessarily clog the network.

When in need of a service for which it either knows no provider or has failed to contact it, a node will look through its locally maintained list of adverts. If it finds one with the right characteristics, i.e. if:

- The service on offer is the one that has just been identified as being currently unavailable (i.e. the one that triggered consultation of the adverts list)
- The service requested in exchange matches its own type (i.e. it can honour its part of the deal)

then the node makes an attempt at contacting the sender of that particular advert.

Assuming that this attempt is successful, a handshake procedure follows whereby both nodes examine the opportunity of forging a new cooperative relationship. This will succeed only if:

- The sender of the advert hasn't changed type since it was sent (and so is still capable of providing the service advertised).
- It still hasn't identified another provider for the service hosted by the initiator of the handshake.

Other factors would probably need to be taken into consideration in a real implementation (e.g. QoS level, service charge…) but these are not modelled in the proof-of-concept simulation. Note that, according to this procedure, a node will never maintain more than N-1 links at a time (i.e. the number of services that it cannot provide to itself), but see the discussion about "volunteering" in conclusions and future work.

If the handshake fails (i.e. communication with the sender was successfully established but one or both of the nodes rejected creating a co-operative link), then the adverts is identified as obsolete and cleared from the local list maintained by the initiator of the negotiation. In this case, the requesting node resumes its search through the list of adverts offering the desired service until either it reaches the last advert in the list (newest first) or finds a suitable provider (i.e. handshake succeeds). The failure to establish a workable partnership has two possible causes:

- A service imbalance (i.e. there isn't any node hosting the required service that needs access to the function corresponding to the type of the node trying to initiate a new co-operative link), in which case the best corrective action to take is to change type.
- The local information about availability is inaccurate (out-of-date or somehow corrupted) or incomplete, in which case the best corrective action is to (re-)advertise the desired co-operative relationship, so that potential candidates that are either unknown or wrongly identified as unsuitable (i.e. assumed to host a different service or not to be "interested" in the initiator's type/offer) become aware of the opportunity.

The decision to follow either course of action is made based on exp. (1), whereby the probability of choosing the "metamorphosis" option increases with the number of failed attempts (which can be interpreted as an indication that service imbalance, not lack of information, is indeed the cause of the repeated inability to forge a partnership).

Results

We used Monte Carlo simulation to gather evidence about the overall efficiency (speed, scalability, robustness…) of Embryo. We do not have space here to present all of our data, so we will focus on showing how system behaviour is affected by the value of the two main parameters, i.e. population size and number of services (N).

All results are for 100 independent realisations. The simulation stops when all peers are either "fully satisfied" (i.e. they have one first neighbour of every type different from their own) or belong to a disconnected sub-graph that cannot reach steady state (size <N). Cases in which this condition was not met before reaching an arbitrary time limit were only encountered in networks of less than 64 peers in the N = 17 scenario and were discarded.

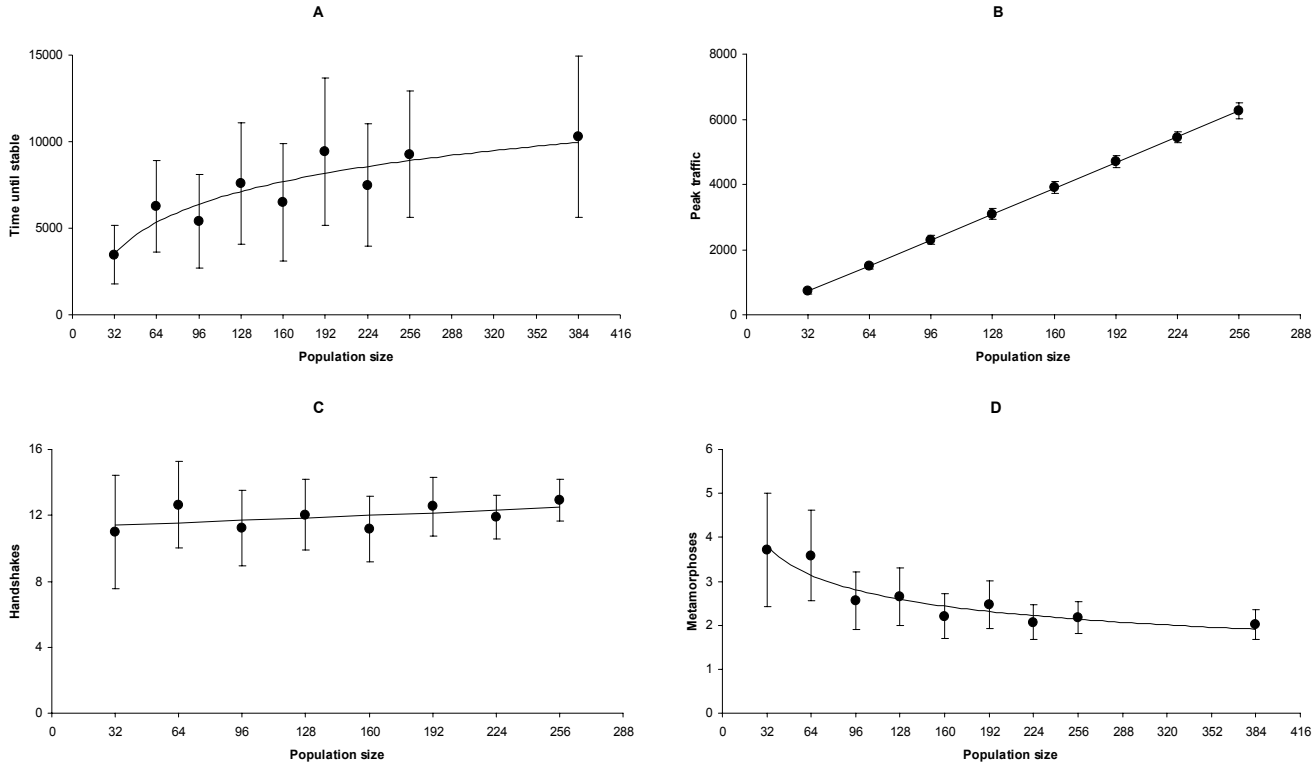

Fig. 2. Evolution of four key variables as a function of population size for N = 9 types or services. (A) Time to steady state. (B) Peak traffic. (C) Number of successful handshakes per node. (D) Number of metamorphoses per node. Error bars indicate standard deviation

Note that since our objective is to demonstrate the fundamental properties of the algorithm and present generic trends revealed by the simulation, units are arbitrary (e.g. "time" is simply the number of simulation steps, "traffic" is the number of adverts being propagated per time-step). For the same reason, we focus on average values, though we also provide the standard deviation as an indication of variability. A more detailed statistical analysis would only be justified if the data consisted in experimental measurements, or if we had simulated system operation at a considerably lower level (e.g. by emulating realistic communication protocols featuring the equivalent of latency, packet loss etc.).

Figure 2 shows the evolution of some key variables, namely the time to reach steady state (A), the peak traffic (B), the number of successful handshakes (C) and the number of metamorphoses (D) for N = 9 services and a variable number of peers. Clearly, the time to reach steady state is a logarithmic function of the population size, while the peak traffic (maximum number of adverts being propagated per time unit) grows linearly with the value of that same parameter. The number of successful handshakes per peer (corresponding to rewiring of the overlay) only increases very slowly with population size, while the number of metamorphoses (corresponding to type change) appears to obey an inverse power law.

These trends all emphasize Embryo's scalability with respect to population size. In particular, the drop in the average number of metamorphoses per peer (likely to be the most "costly" operation) seems to indicate that our algorithm would actually perform better in large systems than in small ones.

As for the influence of the number of services, we ran a number of tests with N = 17. The results are shown in figure 3. We discarded the results for 32 peers as this proved to be a pathological case, with less than 50% of simulation runs converging before reaching the time limit. We attribute this fact to the comparatively low population size / number of services ratio (which implies that the only steady state involves one fully connected graph of 17 peers). Otherwise, the information shown on fig. 3 is identical to the one shown on fig. 2.

Overall, scalability with respect to the number of services is confirmed, though the noise level is comparatively high for the time to reach stable state (fig. 3a). For intermediate to large population sizes, the time to converge doesn't appear to increase much compared with the N = 9 scenario, which is in accordance with the logarithmic trend. As for peak traffic it appears almost unaffected by the number of services. Interestingly, though higher in absolute terms, the number of successful handshakes per peer (fig. 3c) now decreases with population size, which again tends to indicate that performance actually increases as the system becomes larger.

.

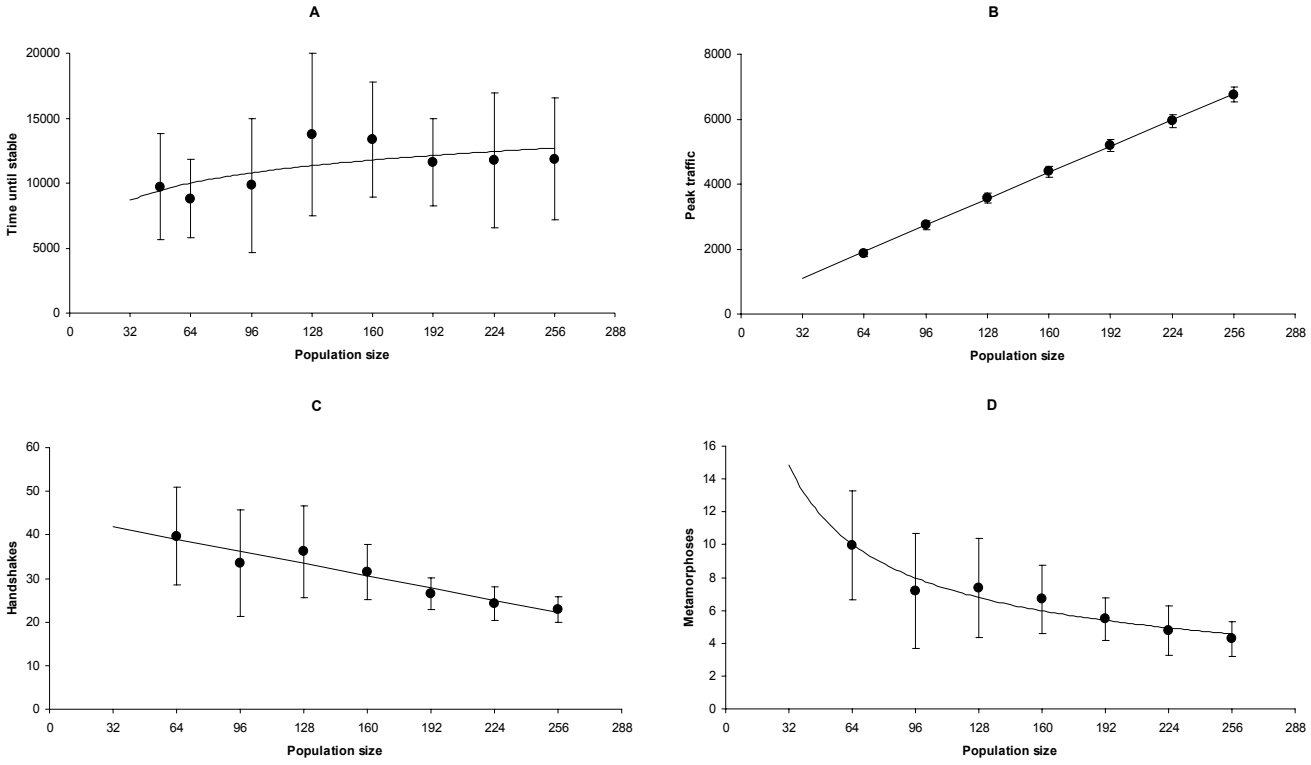

Fig. 3. Evolution of four key variables as a function of population size for N = 17 types or services. (A) Time to steady state. (B) Peak traffic. (C) Number of successful handshakes per node. (D) Number of metamorphoses per node. Error bars indicate standard deviation.

Conclusions and future work

All our results seem to designate Embryo as a suitable design philosophy to build an efficient and reliable P2P service provision infrastructure, especially in large and unpredictable resource-sharing communities. We argue that this makes our algorithm a strong candidate for autonomic deployment and maintenance of ICT systems at the interface between Service-Oriented Architecture (SOA) and Grid computing. This however would require taking into account some additional characteristics that, for the sake of clarity and completeness, were not included in the present study.

For example, the procedure for establishing a relationship described in this paper implies reciprocity/symmetry, in the sense that a co-operative link is only established if each partner decides that it is in its own best interest to choose the other as a provider for a required service. On the contrary, a node can change type (which renders all of its co-operative relationships immediately obsolete) or terminate a link without consulting (or even notifying) its counterpart(s). This simultaneously offers a guarantee against cheating (since any node can unilaterally decide to withdraw from a relationship that it judges unsatisfactory) and makes it obvious that "selfishness" is not an obstacle to the self-organisation process.

We have however experimented with a modified version of the decision and messaging infrastructure in which the initiator of the handshake procedure may accept to perform a function for another node even if it has no personal interest in doing so (we call this procedure "volunteering"). In this version, we assume that a node is able to provide a service to more peers than it has needs (i.e. it has more links than the minimum necessary to meet all of its own requirements). Preliminary findings suggest that such "volunteering" can be highly beneficial to the community as it appears to speed up the self-organisation process and leave fewer or no nodes "excluded" from the final (stable) overlay.

Finally, in biological development, the "preferred neighbourhood" varies from one cell type to the other, leading to the formation of functional organ and tissues, which are basically specialised structures made of a an aggregation of cells belonging to a small sub-set of all possible types. Interestingly, this is also the case in P2P service provision, as not all peers need access to all services, and so the "homogeneous full coverage" scenario described in this paper is obviously a simplification. The more complex collective dynamics likely to emerge in an extended version of Embryo taking into account these various other aspects will be the subject of future work.

References

[1] S. H. Strogatz, "Exploring complex networks" Nature 410, 268-276 (2001).

[2] A.-L. Barabasi, R. Albert. "Emergence of scaling in random networks" Science 286, 509-512 (1999).

[3] M. Jelasity, O. Babaoglu "T-Man: Gossip-based overlay topology management" In: Proceedings of the 3rd International Workshop on Engineering Self-Organising Applications, Utrecht (2005).

[4] S. Kumar, P.J. Bentley. (eds.) "On Growth, Form and Computers" Elsevier Academic Press, London (2003).

[5] J.-J. Suh, S. G. Quan, S.-H. Park, Y. Y. Kim, "Adaptive File Distribution in P2P Network Using Ant Colony Optimization for Smart Home Environment" International Conference on Hybrid Information Systems, Nov. 2006.

[6] Michael Stal, "Web services: beyond component-based computing" Communications of the ACM, vol. 45, no. 10, Oct. 2002.

[7] L. F. Cabrera, C. Kurt, D. Box. "An Introduction to the Web Services Architecture and Its Specifications" Version 2.0, Oct. 2004. http://msdn.microsoft.com/webservices/webservices/understanding/advancedwebservices/default.aspx?pull=/library/en-us/dnwebsrv/html/introwsa.asp

[8] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich. "The Open Grid Services Architecture, Version 1.0" Informational Document, Global Grid Forum (GGF), January 29, 2005. http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf

[9] I. Foster, C. Kesselman, J. Nick, S. Tuecke "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration" Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002. http://www.globus.org/alliance/publications/papers/ogsa.pdf

[10] M. Weisner (1991) "The computer for the 21st century" Scientific American, 265(3):94—104 (1991).

[11] IEEE Pervasive Computing, IEEE Computer Society, ISSN: 1536-1268.

[12] L. McNamara, C. Mascolo, L. Capra "Trust and Mobility Aware Service Provision for Pervasive Computing" Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI) held at 4th International Conference on Pervasive Computing, May 2006.

[13] L. Capra, S. Zachariadis, C. Mascolo "Q-CAD: QoS and Context Aware Discovery Framework for Mobile Systems" ICPS, July 2005.

[14] J. O. Kephart, D. M. Chess "The Vision of Autonomic Computing" IEEE Computer, Jan. 2003.

[15] N. Chase "An autonomic computing roadmap" Oct 2004 http://www-128.ibm.com/developerworks/library/ac-roadmap/

[16] A. Keller, J. P. Martin-Flatin (eds). Second IEEE International Workshop on Self-Managed Networks, Systems & Services (SelfMan 2006 Proceedings), June 2006, Dublin, Ireland; published by Springer Verlag as Volume 3996 of the Lecture Notes in Computer Science (LNCS) Series.

[17] O. Babaoglu, M. Jelasity, A.Montresor, C. Fetzer, S.Leonardi, A. van Moorsel, M van Steen “Self-Star Properties in Complex Information Systems", Lecture Notes in Computer Science, Hot Topics, vol. 3460, Springer-Verlag, 2005.

[18] G. P. Jesi, A. Montresor, O. Babaoglu “Proximity-aware Superpeer Overlay Topologies” Second IEEE International Workshop on Self-Managed Networks, Systems & Services (SelfMan 2006), June 2006, Dublin, Ireland.

[19] K. Aberer, A. Datta, M. Hauswirth, "P-Grid: Dynamics of Self-organizing Processes in Structured P2P systems", In: Peer-to-Peer Systems and Applications, LNCS 3485, Springer, Aug 2005.

[20] K. Aberer, L. Onana Alima, A. Ghodsi, S. Girdzijauskas, M. Hauswirth, S. Haridi "The essence of P2P: A reference architecture for overlay networks", The Fifth IEEE International Conference on Peer-to-Peer Computing, Konstanz, 31 Aug - 2 Sep 2005.

[21] C. Adam, R. Stadler “Implementation and Evaluation of a Middleware for Self-Organizing Decentralized Web Services” Second IEEE International Workshop on Self-Managed Networks, Systems & Services (SelfMan 2006), June 2006, Dublin, Ireland

[22] F. Saffre, H. R. Blok, “SelfService: a Theoretical Protocol for Autonomic Distribution of Services in P2P Communities” In: Proceedings of the 12th IEEE International Conference and Workshops on Engineering of Computer-Based Systems, Greenbelt (2005).

[23] F. Saffre, J. Halloy, J. L. Deneubourg. “The Ecology of the Grid” In: Proceedings of the 2nd IEEE International Conference on Autonomic Computing, Seattle (2005).

[24] F. Saffre, J. Halloy, M. Shackleton, J. L. Deneubourg. “Self-Organised Service Orchestration through Collective Differentiation” IEEE Trans. on Systems Man and Cybernetics, 36(6), 1237-1246. (2007)